Financial Management System (with c# and SQL*)

by

Maddukuri Nivas (19BCE1010)

A project Report Submitted to

Dr. Appalaraju Muralidhar

School Of Computer Science and Engineering

In partial fulfilment of the requirements for the course of

CSE2004 Database Management Systems

In

B.Tech. COMPUTER SCIENCE AND ENGINEERING

VIT CHENNAI
Vandalur – Kelambakkam Road
Chennai – 600127
APRIL 2020

Title: Financial Management System

By

Maddukuri Nivas   19BCE1010

ABSTRACT

The main idea to do this Project is to give different and very useful offerings to the Customer by hiding the unusual content. The project will contain three different logins for Customer, employee and admin.

Customer Login: Customer will contain the options like
1) Withdraw
2) Deposit
3) Transaction
4) Balance Enquiry
5) Loan
6) Changing PIN

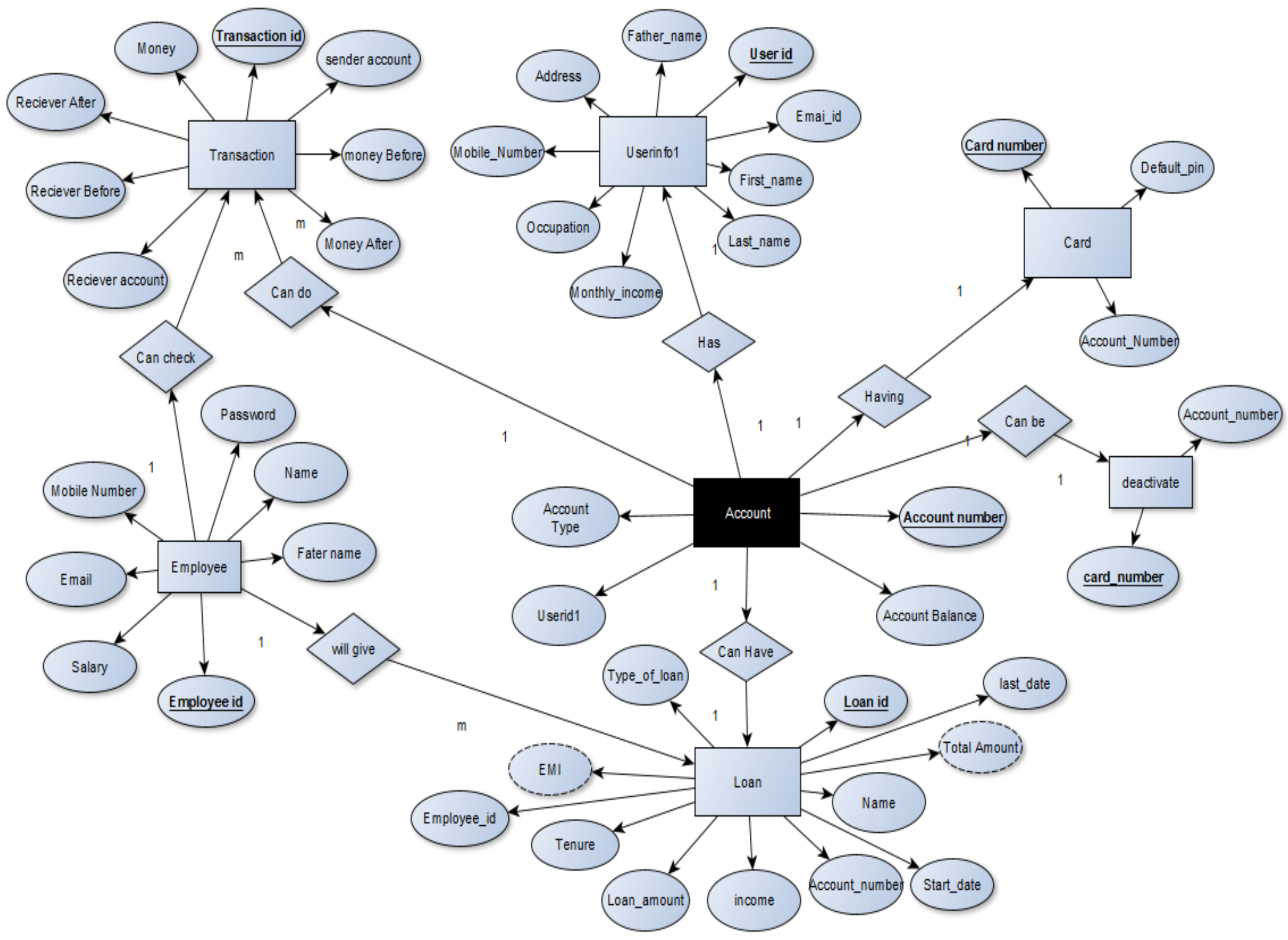
Employee Login: In Employee Login, Employee will contain options like
1) Creating Account for Customer,
2) Adding and withdrawing money in customer account
3) Loan section
4) Deactivate and activating Customer Account
5) Changing Password Option
6) Admin Can see the loan Takers who crossed the last date to pay the loan in his load page.

Admin Login: In Admin Login Admin will contain Options Like,
1) Adding Employee
2) Monitoring all the ATM’s
3) Customer can change his credentials by contacting ADMIN

Extra Section Added To this Project is:

Customer will immediately get message when there is any transaction, withdraw and Deposit in his account.

ER Diagram To this Project is:



In this Project First We included by creating Tables:

By the Query:

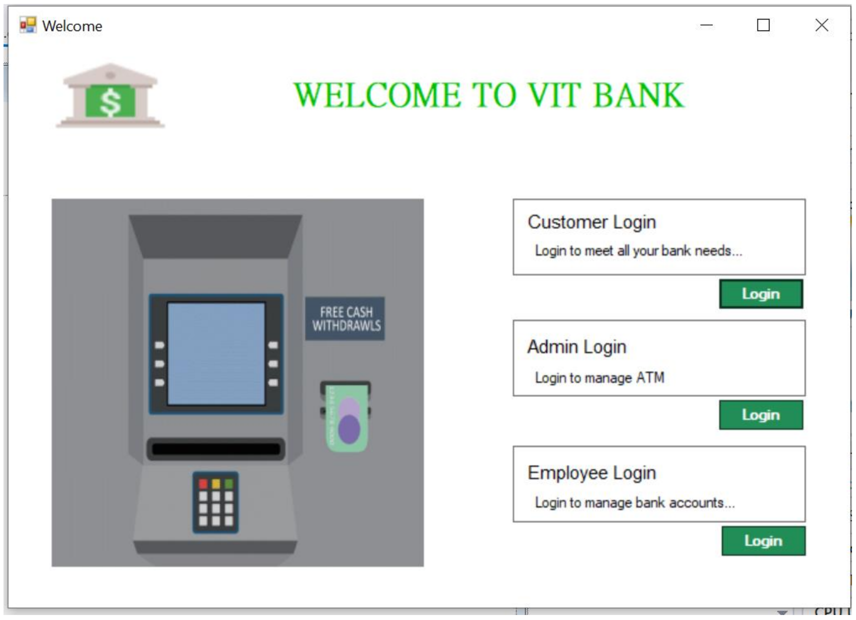
```
CREATE TABLE [dbo].[account] (
    [account_number] BIGINT       IDENTITY (3000, 1) NOT NULL,
    [account_type]   VARCHAR (50) NOT NULL,
    [balance]        FLOAT (53)   NULL,
    [userid1]        INT          NOT NULL,
    PRIMARY KEY CLUSTERED ([account_number] ASC),
    CONSTRAINT [userid] FOREIGN KEY ([userid1]) REFERENCES [dbo].[userinfo1] ([User_id])
);
```

All tables are created like this

Next we can created the forms by adding buttons and labels and related them.

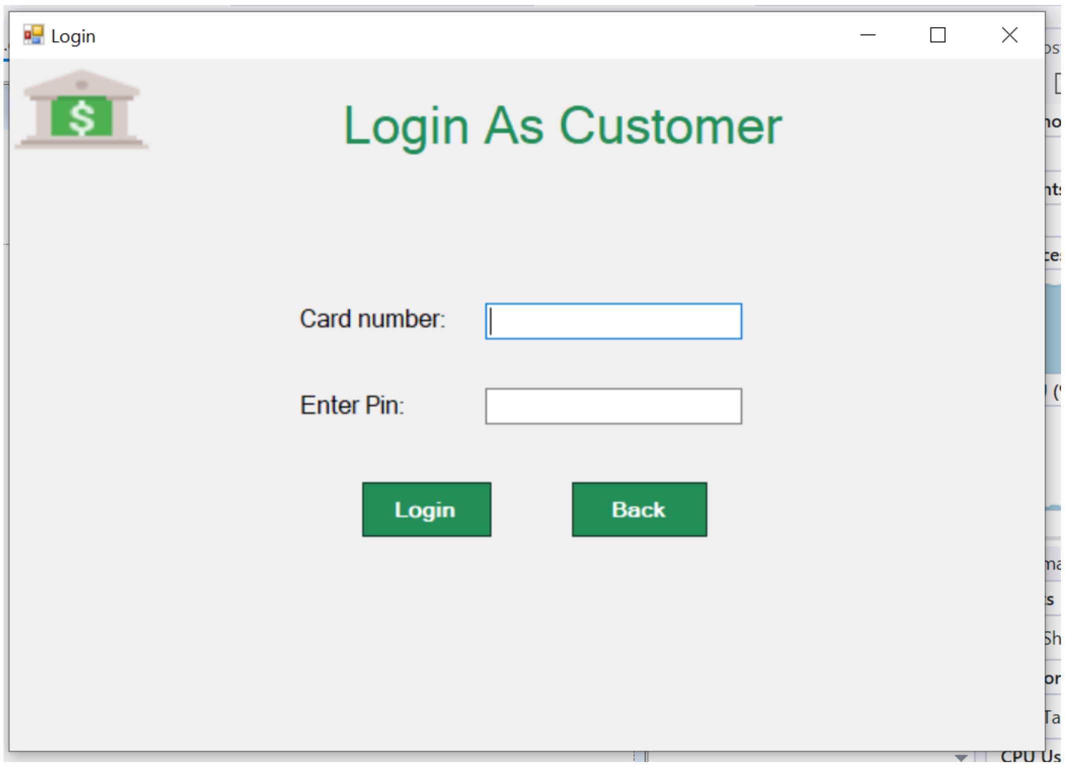
The Starting Page is:

- ➔ After clicking on the start button it will show the below form.
- ➔ Then you can select the option where you wanted to go.



Customer Login Page:

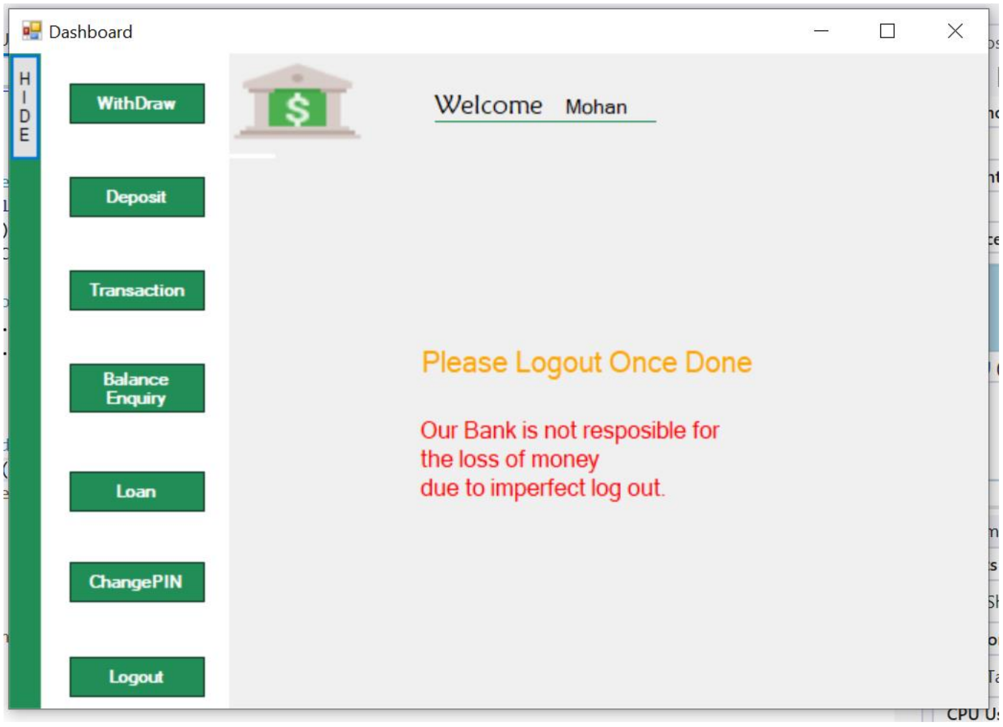
- ➔ After clicking on the customer login it will show the below form.
- ➔ As per the selection Option it will show the window form.



Code to accept a Customer (Query is highlighted):

```
public void getdata()
        {
            string name = textBox1.Text;
            string pass = textBox2.Text;
            recby = name;
            da = new SqlDataAdapter("Select card_number From Card where card_number like'" +
textBox1.Text + "'and default_pin='" + textBox2.Text + "'",cm);
            DataTable dt = new DataTable();
            da.Fill(dt);
            if(dt.Rows.Count == 1)
            {
                SqlDataAdapter da1 = new SqlDataAdapter("Select card_number From deactivate where
card_number ='" + textBox1.Text + "'", cm);
                DataTable dt1 = new DataTable();
                da1.Fill(dt1);
                if(dt1.Rows.Count == 1)
                {
                    MessageBox.Show("This Account is Deactivated");
                    textBox1.Text = "";
                    textBox2.Text = "";
                }
                else
                {
                    Dashboard obj = new Dashboard();
                    obj.Show();
                    this.Hide();
                }

            }
            else
            {
                MessageBox.Show("Invalid Credentials");
                clear();
            }
        }
```

If all the credentials are correct it will open the page:



Welcome Mohan

Please Logout Once Done

Our Bank is not resposible for
the loss of money
due to imperfect log out.

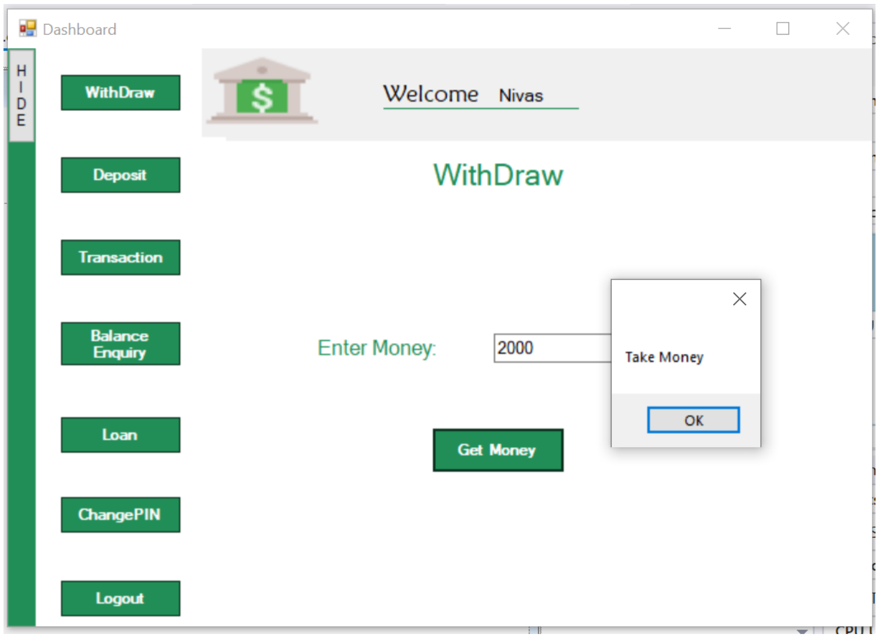
The Code to withdraw is (in this code it will also update the atm balance also):

```
if (textBox2.Text == "")
            {
                MessageBox.Show("Please Fill All Aspects");
            }
            else
            {
                cm.Open();
                SqlCommand com = new SqlCommand("select balance from account where 
account_number=(select account_number from Card where card_number='" + starting.recby.ToString() + 
"')", cm);
                SqlDataReader reader = com.ExecuteReader();
                reader.Read();
                string str;
                string bal;
                float niv = 0;
                float moh = 0;
                string accno;
                int c = 8000;
                if (reader.HasRows)
                {
                    str = reader["balance"].ToString();
                    reader.Close();
                    SqlCommand com1 = new SqlCommand("select atm_balance from atm_balance where 
atm_number='" + c + "'", cm);
                    SqlDataReader reader1 = com1.ExecuteReader();
                    reader1.Read();
                    if (reader1.HasRows)
                    {
                        bal = reader1["atm_balance"].ToString();
                        niv = float.Parse(str);
                        moh = float.Parse(bal);
                        float depo = float.Parse(textBox2.Text);
                        reader1.Close();
                        if (niv >= depo)
                        {
                            if (moh >= depo)
                            {
                                SqlCommand com2 = new SqlCommand("select account_number from Card 
where card_number='" + starting.recby.ToString() + "'", cm);
                                SqlDataReader reader2 = com2.ExecuteReader();
                                reader2.Read();
                                if (reader2.HasRows)
                                {
                                    accno = reader2["account_number"].ToString();
                                    reader2.Close();
                                    niv = niv - depo;
                                    moh = moh - depo;
                                    SqlCommand sqlcmd = new SqlCommand("withdraw", cm);
                                    sqlcmd.CommandType = CommandType.StoredProcedure;
                                    sqlcmd.Parameters.AddWithValue("@account_number", accno);
                                    sqlcmd.Parameters.AddWithValue("@balance", niv);
                                    sqlcmd.ExecuteNonQuery();
                                    SqlCommand sqlcm = new SqlCommand("atm_withdraw", cm);
                                    sqlcm.CommandType = CommandType.StoredProcedure;
                                    sqlcm.Parameters.AddWithValue("@atm_number", c);
                                    sqlcm.Parameters.AddWithValue("@atm_balance", moh);
                                    sqlcm.ExecuteNonQuery();
                                    string mob;
                                    SqlCommand com3 = new SqlCommand("select Mobile_Number from 
userinfo1 where User_id=(select userid1 from account where account_number=(select account_number 
from Card where card_number='" + starting.recby.ToString() + "'))", cm);
                                    SqlDataReader reader3 = com3.ExecuteReader();
                                    reader3.Read();
                                    if (reader3.HasRows)
                                    {
                                        mob = reader3["Mobile_Number"].ToString();
                                        using (var wb = new WebClient())
                                        {
                                            byte[] response =
wb.UploadValues("https://api.textlocal.in/send/", new NameValueCollection()
                                            {
                                            {"apikey" , "TQYPZV3sjH0-
1q5LmkniE0gQNtPOQOBA695SsGs2uR"},
                                            {"numbers" , "91" + mob },

                                            {"message" , "VIT BANK\nAccount Number: "+ accno +"\n" +
depo + " rupees withdrawn from your account\nBalance in your account is: " + niv +"\nContact number:
+916300189494"},

                                            });
                                            string result =
System.Text.Encoding.UTF8.GetString(response);
                                            MessageBox.Show("Take Money");

                                            reader3.Close();
                                            textBox2.Text = "";
                                        }

                                    }


                                }
                                else
                                {
                                    MessageBox.Show("Money you have entered exceeds the money in
atm");
                                }

                            }
                            else
                            {
                                MessageBox.Show("Money you have entered is more than the money in
your account");
                            }
                        }
                        cm.Close();
                    }

                }

}
```

Update Query to get the money updated in account table as well as atm table is:

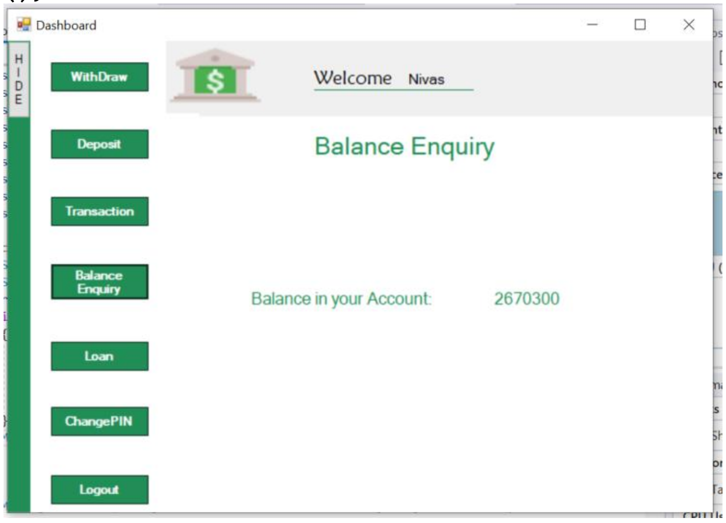
```
CREATE PROCEDURE [dbo].[withdraw]
       @account_number bigint,
       @balance float
AS
        Update [account] SET [balance]=@balance where account_number=@account_number
RETURN 0
CREATE PROCEDURE [dbo].[atm_withdraw]
       @atm_number int,
       @atm_balance float
AS
         Update [atm_balance] SET [atm_balance]=@atm_balance where atm_number=@atm_number
RETURN 0
```

After All Successful Updates it will Show:



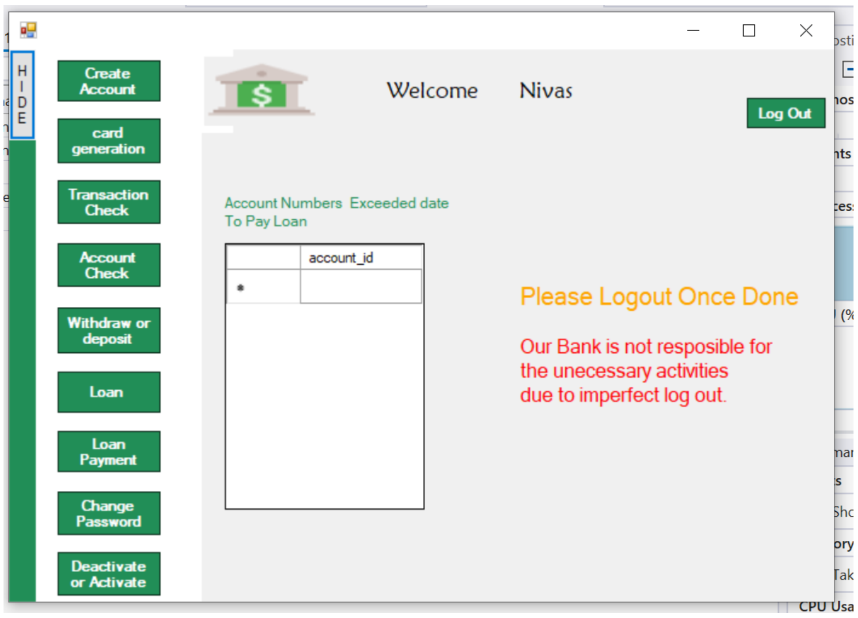
The Query to Show Balance is:

```
SqlConnection cm = new SqlConnection("Data 
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\Nivas\\Documents\\atm.mdf;Integrated 
Security=True;Connect Timeout=30");
            cm.Open();
            SqlCommand com = new SqlCommand("select * from account where account_number=(select 
account_number from Card where card_number ='"+starting.recby.ToString() + "')", cm);
            SqlDataReader reader = com.ExecuteReader();
            reader.Read();
            if (reader.HasRows)
            {
                label3.Text = reader["balance"].ToString();
                reader.Close();
            }
            cm.Close();
```



By the Same process like above Employee will also join:

- ➔ By entering the valid credentials Employee will also be logged in.
- ➔ In the Starting Page there will be a table visible which is nothing but if the employee gives loan to any customer. If the last date was crossed to pay loan then his/her Account number will be displayed.



Welcome Nivas

Account Numbers Exceeded date To Pay Loan

	account_id
*	

Please Logout Once Done

Our Bank is not resposible for the unecessary activities due to imperfect log out.

Code to create an Account for Customer:

Query to insert into userinfo1 table:

```
REATE PROCEDURE [dbo].[Add_customer]
	@First_name varchar(50),
	@Last_name varchar(50),
	@Mobile_number varchar(50),
	@Email varchar(50),
	@address varchar(50),
	@Monthly_income varchar(50),
	@Occupation varchar(50),
	@Father_name varchar(50)
AS
	INSERT INTO
[userinfo1](First_Name,Last_Name,Mobile_Number,[Address],Email_id,Occupation,Father_name,Monthly_inc
ome)
	VALUES(@First_name,@Last_name,@Mobile_number,@address,@Email,@Occupation,@Father_name,@Monthl
y_income)
```

Query to insert into Add_Account table:

```
CREATE PROCEDURE [dbo].[Add_Account]
	@account_type varchar(50),
	@Balance float,
	@userid1 int
AS
	INSERT INTO account(account_type,balance,userid1)
	VALUES(@account_type,@Balance,@userid1)
RETURN 0
```

Code to add user:

```
if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || textBox4.Text=="" ||
textBox5.Text=="" || textBox6.Text=="" || textBox7.Text=="" || richTextBox1.Text=="")

            {
                MessageBox.Show("Please Fill all aspects");
            }
            else
            {
                SqlCommand sqlcmd = new SqlCommand("Add_customer", cm);
                sqlcmd.CommandType = CommandType.StoredProcedure;
                sqlcmd.Parameters.AddWithValue("@First_name", textBox1.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@Last_name", textBox2.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@Mobile_number", textBox3.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@Email", textBox4.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@address", richTextBox1.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@Monthly_income", textBox5.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@Occupation", textBox6.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@Father_name", textBox7.Text.Trim());
                cm.Open();
                sqlcmd.ExecuteNonQuery();
                MessageBox.Show("Customer Added");

                SqlCommand com = new SqlCommand("select * from userinfo1 where User_id=(select 
max(User_id) from userinfo1)", cm);
                SqlDataReader reader = com.ExecuteReader();
                reader.Read();
                if (reader.HasRows)
                {
                    label13.Show();
                    label13.Text = reader["User_id"].ToString();
                    reader.Close();
                }
                cm.Close();
            }
```

Code to add Account:

```
if (textBox8.Text == "" || comboBox2.Text == "")
            {
                MessageBox.Show("Please Fill all aspects");

            }
            else
            {
                SqlCommand sqlcmd = new SqlCommand("Add_account", cm);
                sqlcmd.CommandType = CommandType.StoredProcedure;
                sqlcmd.Parameters.AddWithValue("@account_type", comboBox2.Text);
                sqlcmd.Parameters.AddWithValue("@Balance", textBox8.Text.Trim());
                sqlcmd.Parameters.AddWithValue("@userid1", label13.Text);
                cm.Open();
                sqlcmd.ExecuteNonQuery();


                label14.Visible = true;
                label15.Visible = true;

                SqlCommand com = new SqlCommand("select * from account where account_number=(select 
max(account_number) from account)", cm);
                SqlDataReader reader = com.ExecuteReader();
                reader.Read();
                if (reader.HasRows)
                {
                    label15.Text = reader["account_number"].ToString();
                    reader.Close();

                        using (var wb = new WebClient())
                        {
                            byte[] response = wb.UploadValues("https://api.textlocal.in/send/", new 
NameValueCollection()
                            {
                            {"apikey" , "TQYPZV3sjH0-1q5LmkniE0gQNtPOQOBA695SsGs2uR"},
                            {"numbers" , "91" + textBox3.Text.ToString() },
                            {"message" , "Welcome To VIT BANK \n Thank you for creating a account.\n
Vit bank gives all the best features\n your User ID is :" + label13.Text.ToString() +"\nYour Account
Number is: " + label15.Text.ToString() + "\n Your card will be activated soon. \n Contact Number:
6300189494.\nTHANK YOU"},

                            });
                            MessageBox.Show("Account Added");
                            string result = System.Text.Encoding.UTF8.GetString(response);
                        }
                    clear1();
                }

                cm.Close();
            }

}
```

Loan Section:

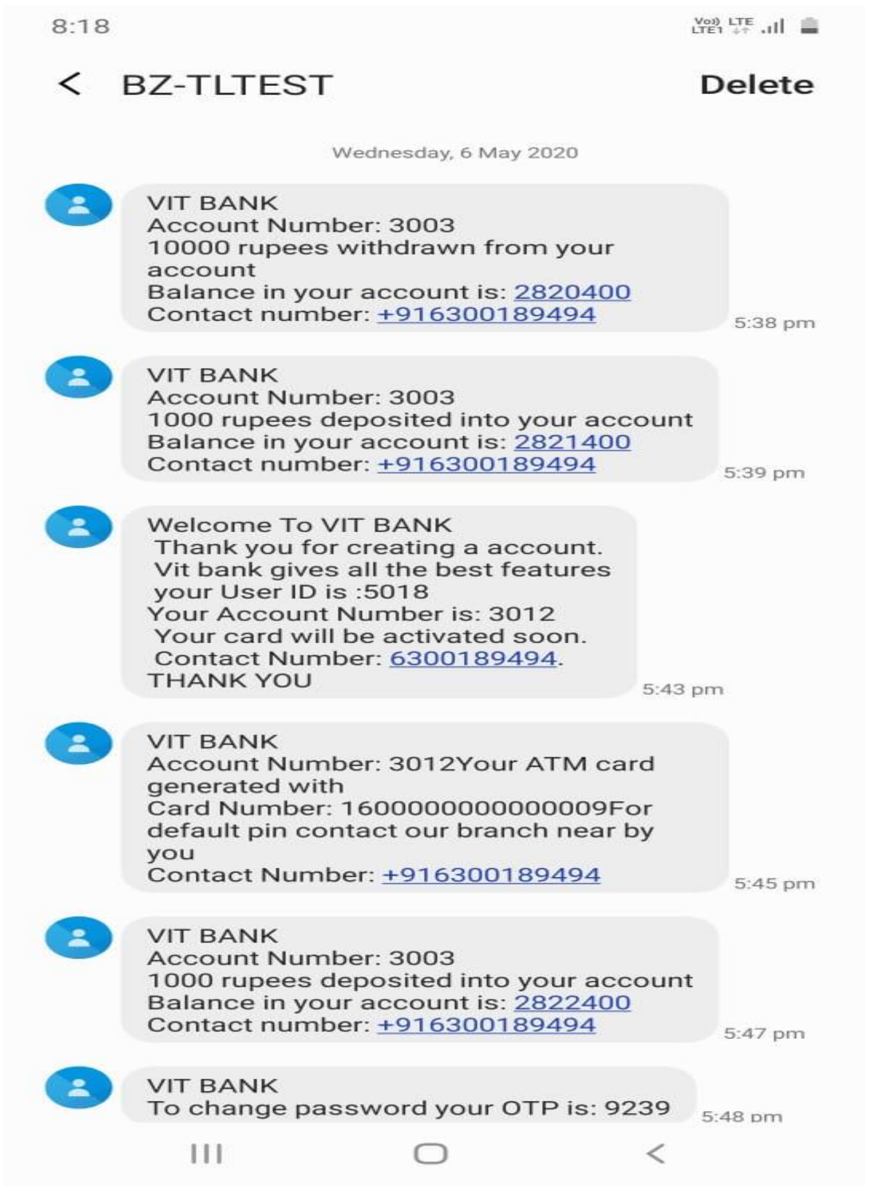
Loan given by calculating the cibil scores. The cibil scores are in the following Pattern

```
float result = 0;
                        int sum;
                        int ten;
                        float sum1 = 0;
                        float income = float.Parse(textBox2.Text);
                        for (int i = 0; i < dataGridView1.Rows.Count; i++)
                        {
                            sum = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);
                            ten = Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value);
                            sum1 = sum1 + ten;/*adding the tenure of all emi's using loop*/
                            result = result + (sum * ten);/*adding the emi*tenure in a loop*/
                        }
                        double per;
                        double percentage;
                        per = result / income;   /*income is the annual income*/
                        percentage = per * 100;
                        int cibil = 0;
                        int cibil1 = 0;
                        int totcibil = 0;
                        if (percentage <= 10 && percentage >= 0)
                        {
                            cibil = 500;
                        }
                        else if (percentage <= 20 && percentage > 10)
                        {
                            cibil = 450;
                        }
                        else if (percentage <= 30 && percentage > 20)
                        {
                            cibil = 400;
                        }
                        else if (percentage <= 40 && percentage > 30)
                        {
                            cibil = 350;
                        }
                        else if (percentage <= 50 && percentage > 40)
                        {
                            cibil = 300;
                        }
                        else if (percentage <= 60 && percentage > 50)
                        {
                            cibil = 250;
                        }
                        else if (percentage <= 70 && percentage > 60)
                        {
                            cibil = 200;
                        }
                        else
                        {
                            MessageBox.Show("Not Eligible For loan Because Percentage Greater Than 
70");
                            cibil = 1000;
                        }
                        if (sum1 <= 12 && sum1 >= 0)
                        {
                            cibil1 = 400;
                        }
                        else if (sum1 <= 24 && sum1 > 12)
                        {
                            cibil1 = 350;
                        }
                        else if (sum1 <= 36 && sum1 > 24)
                        {
                            cibil1 = 300;
                        }
                        else if (sum1 <= 48 && sum1 > 36)
                        {
                            cibil1 = 250;
                        }
                        else if (sum1 <= 60 && sum1 > 48)
                        {
                            cibil1 = 200;
                        }
                        else if (sum1 <= 72 && sum1 > 60)
                        {
                            cibil1 = 150;
                        }
                        else if (sum1 <= 84 && sum1 > 72)
                        {
                            cibil1 = 100;
                        }
                        else
                        {
                            MessageBox.Show("Not Eligible For loan Because Tenure Greater Than 84 
Months");
                            cibil1 = 1000;
                        }
                        totcibil = cibil + cibil1;
                        if (totcibil <= 900 && totcibil >= 300)
                        {
                          messageBox.Show(“Loan Will Be given”);
                        }
                        Else:
                        {
                            messageBox.Show(“Not eligibile for Loan”);
                        }
```

The select Query and update query are same as the above code to all the Options.

Due to more code I am sharing only some code and some Query.

The mobile messages are in the below Format:


8:18

BZ-TLTEST

Delete

Wednesday, 6 May 2020

VIT BANK
Account Number: 3003
10000 rupees withdrawn from your account
Balance in your account is: 2820400
Contact number: +916300189494
5:38 pm

VIT BANK
Account Number: 3003
1000 rupees deposited into your account
Balance in your account is: 2821400
Contact number: +916300189494
5:39 pm

Welcome To VIT BANK
Thank you for creating a account.
Vit bank gives all the best features
your User ID is :5018
Your Account Number is: 3012
Your card will be activated soon.
Contact Number: 6300189494.
THANK YOU
5:43 pm

VIT BANK
Account Number: 3012Your ATM card generated with
Card Number: 1600000000000009For default pin contact our branch near by you
Contact Number: +916300189494
5:45 pm

VIT BANK
Account Number: 3003
1000 rupees deposited into your account
Balance in your account is: 2822400
Contact number: +916300189494
5:47 pm

VIT BANK
To change password your OTP is: 9239
5:48 pm

Extra Features can be added if we use highly Modern Databases or no Sql databases in future are:

➔ If any transaction is failed in between due to power loss or high network traffic etc… then customer can easily get the money back or money to the account which he wants to transfer. This is based on the sum of money in both the accounts should be the same before and after the transaction. So this is the reason we cannot determine to which account this money is going after transaction failed.

➔ No Sql database is good for complex queries and have a dynamic schema. And it is a distributed database system with non-relational.

➔ Using highly modern databases we can include mini statement in this project.

References:

➔ **To do this project or to gain best knowledge here are some extraordinary links:**

https://cs.wmich.edu/gupta/teaching/cs4430/cs4430SummII19web/lectureNotesCS4430/Top%2018%20Database%20Projects%20Ideas%20for%20Students%20Lovelycoding_org.pdf

https://www.kashipara.com/project/topics/latest_sql-database-project-ideas_15

➔ **And you can also use some university or academy websites:**

W3Schools & Tutorials Point

➔ **And can also refer some textbooks:**

Database system concept by Henry Korth or CJ Date

And

Database Management System by Navathe

"If there are any mistakes in my explanation or documentation please forgive me. I hope I will improve better than this time to next time."

------------------------------------ The End ------------------------------------