**Bank Management System**

**Project Documentation of**

**“Bank Management System”**

Teacher:

Miss Salas Akbar

Students:

Fatima Butt (02-134201-037)

Muhammad Nawfal Burhan (02-134201-035)

Shepher Abraham Gill (02-134201-096)

# Acknowledgement:

We have taken serious efforts in this project. It would not have been possible without the support of our teachers “Miss Sameena”, and “Miss Salas”. We are grateful for the excellent quality of lectures they have provided us that made us achieve this project. We would also like to thank many YouTube channels who are sincerely passionate to educate the young minds of our generation without any cost. We would also like to express our special gratitude towards our parents for always supporting us from the beginning and until the end. Lastly, we are grateful to each other for becoming a great team and completing this project and it would not have been easy without each other’s help.

# Features:

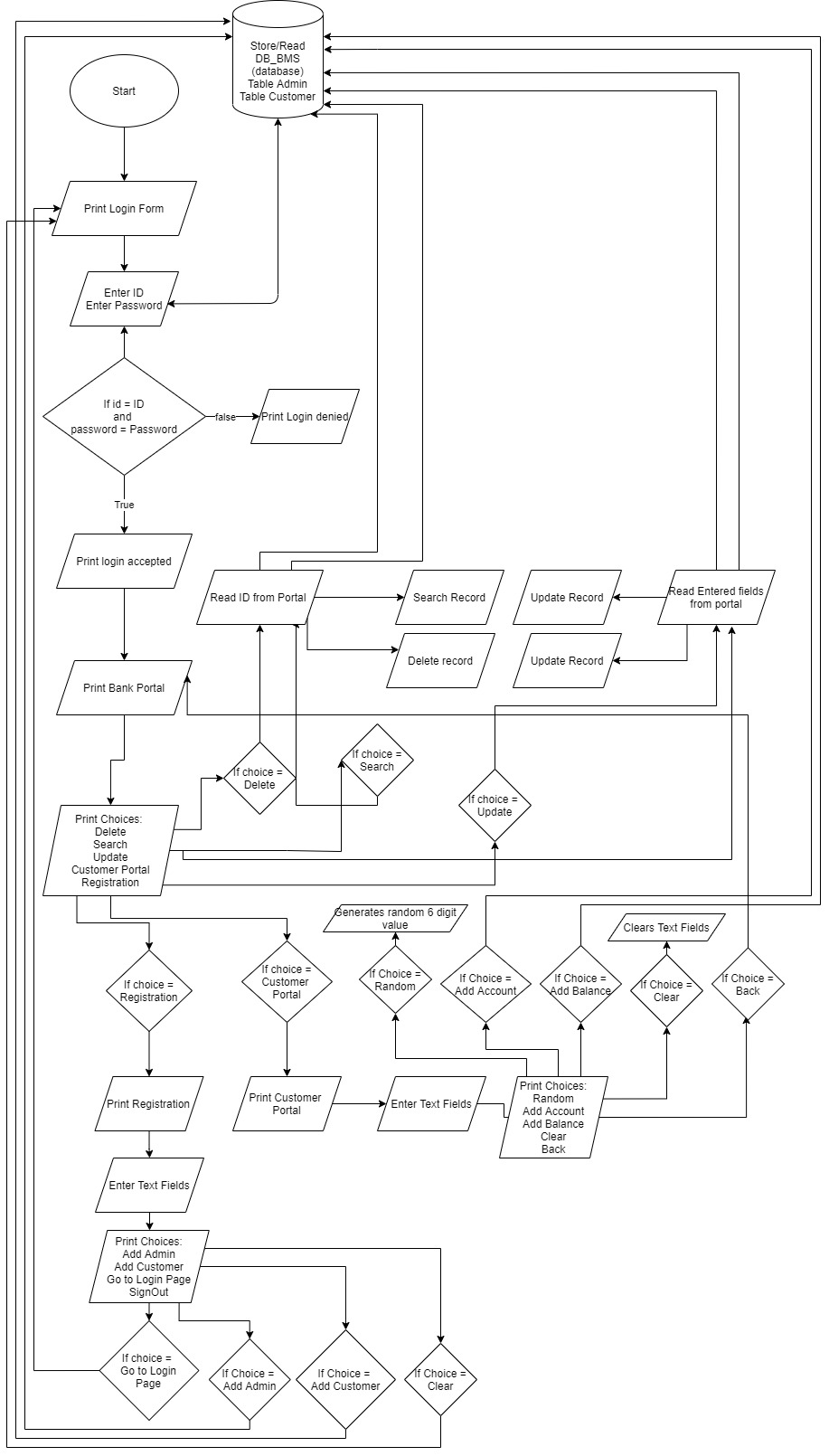
* This software can store information of users and admins.
* It categorizes all users in sections according to their information and chosen services
* Time complexity achieved at its best performance with an easy and user-friendly interface.

# Project Analysis:

This software, which can be a great help for the Bank managing team is built specifically for chosen admins which allows them to store data into their system using a Graphical user interface. It allows them to store records into a database and enables them to edit them according to their needs. The records are stored automatically with a click of a button and is efficient as it displays the records within a second. Admins can operate with this software to check a clients current/savings balance and displays the balance according to his/her respective interest rate.

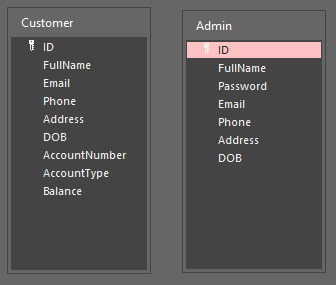
With this program, managing would be very easy as there would be no need to use the paper system and the software requires nothing more than even a low-spec system to work properly. The system software would be one click away to ease.

# Flowchart:



# Class Diagram:

# Database Diagram:



# **CODE:**

package bankmanagementsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

public class ConnectionToDB {

Connection ConToDB=null; // connect with DB

public Connection EstabConnec()

{

try

{

Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");

ConToDB=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\Zaka\\Desktop\\Assignment\\Sem2\\OOP Lab\\Bank-Management-System\_035\_037\_096\\DB\_BMS.accdb");

// JOptionPane.showMessageDialog(null, "Connected");

}

catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return ConToDB;

}

}

package bankmanagementsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

public class AdminLogin {

ConnectionToDB con = new ConnectionToDB();

Connection con\_obj = con.EstabConnec();

Statement stmt = null;

PreparedStatement pstmt = null;

ResultSet res = null;

String name, password, email, address, phone, dateOfBirth ;

public boolean addAdmin(String name, String password, String email,String address, String Phone, String dateOfBirth)

{

boolean b = false;

String sql ="insert into Admin(FullName, Password, Email, Phone, Address, DOB)values('"+name+"','"+password+"','"+email+"','"+address+"','"+Phone+"','"+dateOfBirth+"')";

try

{

stmt = con\_obj.createStatement();

int res =stmt.executeUpdate(sql);

if(res>0)

{

// JOptionPane.showMessageDialog(null, "Insterted");

b = true;

}

else

{

// JOptionPane.showMessageDialog(null, "Error");

b = false;

}

}

catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return b;

}

public boolean addCustomer(String name, String email,String address, String Phone, String dateOfBirth)

{

boolean b = false;

String sql ="insert into Customer(FullName, Email, Phone, Address, DOB)values('"+name+"','"+email+"','"+address+"','"+Phone+"','"+dateOfBirth+"')";

try

{

stmt = con\_obj.createStatement();

int res =stmt.executeUpdate(sql);

if(res>0)

{

// JOptionPane.showMessageDialog(null, "Insterted");

b = true;

}

else

{

// JOptionPane.showMessageDialog(null, "Error");

b = false;

}

}

catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return b;

}

public boolean LoginUser(String id, String password)

{

String loginString = "Select \* from Admin where ID ='"+id+"' and Password ='"+password+"' ";

boolean b;

try

{

pstmt = con\_obj.prepareStatement(loginString);

res = pstmt.executeQuery();

if (res.next())

{

b = true;

}

else

{

b = false;

}

}

catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

b = false;

}

return b;

}

}

package bankmanagementsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

public class Bank\_Portal {

ConnectionToDB ConToDB=new ConnectionToDB();

Connection con = ConToDB.EstabConnec();

Statement stat=null; // insert, update, delete

PreparedStatement pstat=null; // select

ResultSet res = null;

int id,accNum,balance;

String fullName,email,phone,address,DOB,accType;

public boolean DeleteUser(int UID)

{

boolean b=false;

String sql="Delete from Customer where ID='"+UID+"'";

try

{

stat=con.createStatement(); // convert a statement into query

int res=stat.executeUpdate(sql); // if that query is executed, means if data is entered then int will greater than 0 else it will be 0

if(res>0)

{

// JOptionPane.showMessageDialog(null, "Data Inserted");

b = true;

}

else

{

b=false;

//JOptionPane.showMessageDialog(null, "Error");

}

}catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return b;

}

public boolean FetchUser(int UID)

{

boolean b=false;

String FetchString= "Select \* from Customer where ID='"+UID+"'";

try

{

pstat=con.prepareStatement(FetchString);

res=pstat.executeQuery();

this.id=UID;

while(res.next())

{

this.fullName=res.getString("FullName");

this.email=res.getString("Email");

this.phone=res.getString("Phone");

this.address=res.getString("Address");

this.DOB=res.getString("DOB");

this.accNum=res.getInt("AccountNumber");

this.DOB=res.getString("DOB");

this.accType=res.getString("AccountType");

this.balance=res.getInt("Balance");

b= true;

}

}catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

b= false;

}

return b;

}

public boolean UpdateUser(int UID, String FName, String email, String phone, String address,String dob)//, int accNum, String accType, int balance)

{

boolean b=false;

String updatUser = "update Customer set (FullName,Email,Phone,Address,DOB) = ('"+FName+"','"+email+"','"+phone+"','"+address+"','"+dob+"') where ID ='"+UID+"' " ;

try

{

stat=con.createStatement();

int result = stat.executeUpdate(updatUser);

if(result>0)

{

b= true;

}

}

catch(SQLException ex)

{

JOptionPane.showMessageDialog(null, ex);

b= false;

}

return b;

}

}

package bankmanagementsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

public class CustomerPortal {

ConnectionToDB ConToDB=new ConnectionToDB();

Connection con = ConToDB.EstabConnec();

Statement stat=null; // insert, update, delete

PreparedStatement pstat=null; // select

ResultSet res = null;

int ID,AccNumbwe;

String AccType;

public boolean AddAccount(int ID ,int AccNum, String AccType)

{

boolean b=false;

String sql="update Customer set (AccountNumber,AccountType)=('"+AccNum+"','"+AccType+"') where ID='"+ID+"'";

try

{

stat=con.createStatement(); // convert a statement into query

int res=stat.executeUpdate(sql); // if that query is executed, means if data is entered then int will greater than 0 else it will be 0

if(res>0)

{

// JOptionPane.showMessageDialog(null, "Data Inserted");

b = true;

}

else

{

b=false;

//JOptionPane.showMessageDialog(null, "Error");

}

}catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return b;

}

}

package bankmanagementsystem;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class Accounts {

ConnectionToDB ConToDB=new ConnectionToDB();

Connection con = ConToDB.EstabConnec();

Statement stat=null; // insert, update, delete

PreparedStatement pstat=null; // select

ResultSet res = null;

int ID, AddBal ;

float PrevBal;

String AccType;

public boolean AddBalance(int ID, float bal)

{

this.ID=ID;

boolean b=false;

float ba =BalCheck(bal);

String sql="update Customer set Balance = '"+ba+"' where ID='"+this.ID+"'";

try

{

stat=con.createStatement(); // convert a statement into query

int res=stat.executeUpdate(sql); // if that query is executed, means if data is entered then int will greater than 0 else it will be 0

if(res>0)

{

b = true;

}

else

{

b=false;

}

}catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return b;

}

public float BalCheck(float bal)

{

float newBal = 0;

String FetchQuery= "Select \* from Customer where ID='"+this.ID+"'";

try

{

pstat=con.prepareStatement(FetchQuery);

res=pstat.executeQuery();

while(res.next())

{

AccType=res.getString("AccountType");

PrevBal=res.getFloat("Balance");

}

Current c = new Current();

Savings s = new Savings();

if(AccType.equals("Current"))

{

newBal=c.balCal(bal);

}

else

{

newBal= s.balCal(bal);

}

}catch(Exception ex)

{

JOptionPane.showMessageDialog(null, ex);

}

return newBal+PrevBal;

}

public float balCal(float bal)

{

PrevBal +=bal;

if(PrevBal<=5000000)

{

return PrevBal;

}

else

{

JOptionPane.showMessageDialog(null, "Limit Reached");

}

return 0;

}

}

package bankmanagementsystem;

public class Current extends Accounts{

@Override

public float balCal(float bal)

{

float calbal=super.balCal(bal);

return calbal;

}

}

package bankmanagementsystem;

public class Savings extends Accounts{

float IntRate= 0.02f;

@Override

public float balCal(float bal)

{

float cal = super.balCal(bal);

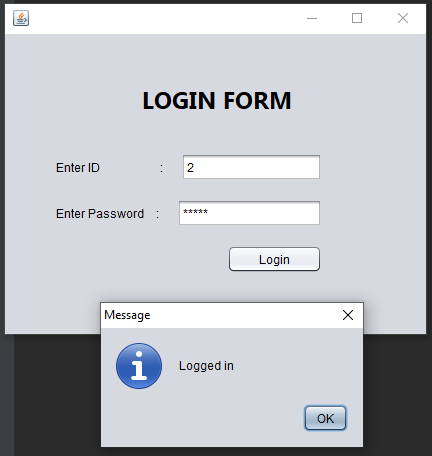
return (cal\*IntRate)+cal;

}

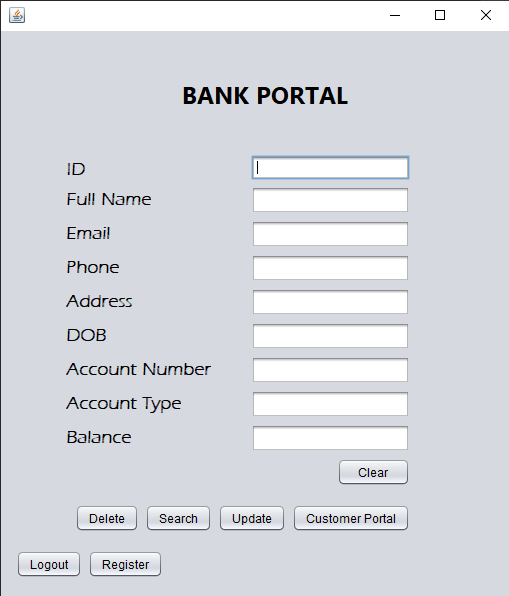
}

# **Outputs:**

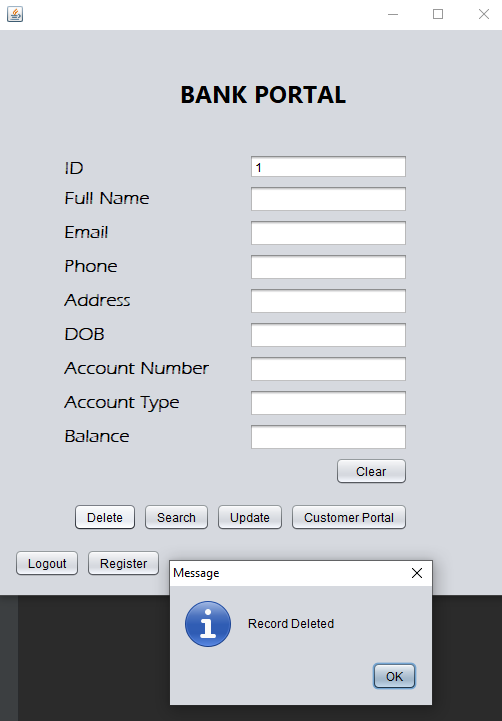
## Login Portal:

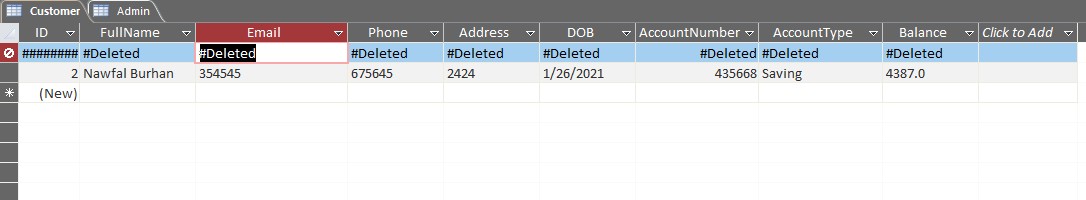


## Bank Portal:

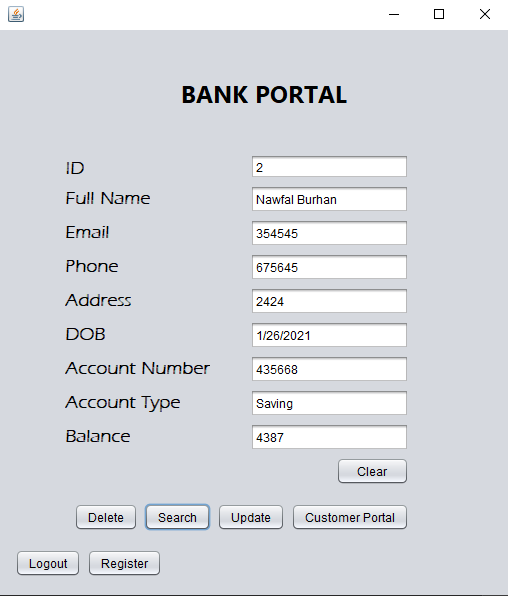


## Delete:

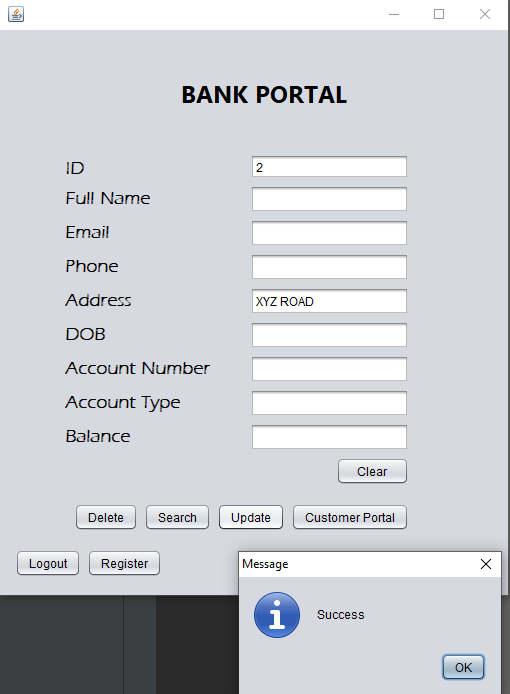


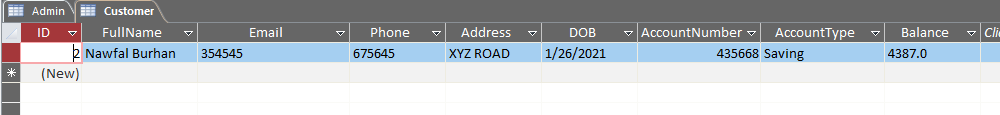


## Search:

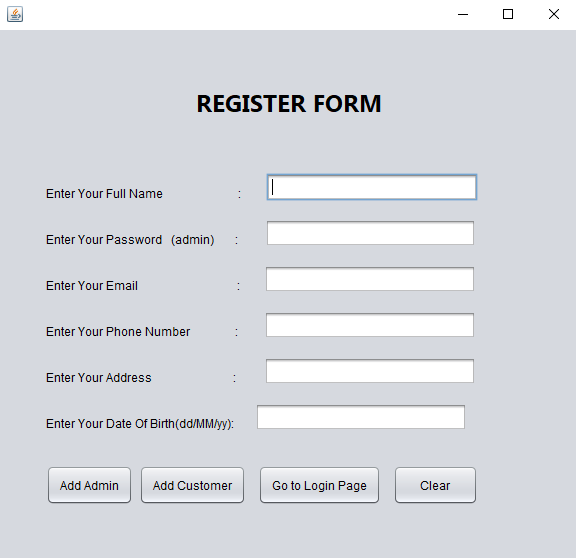


## Update:

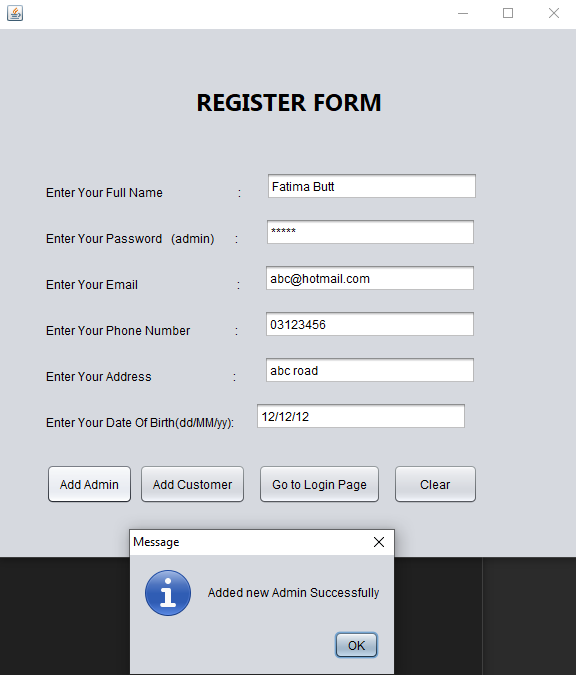


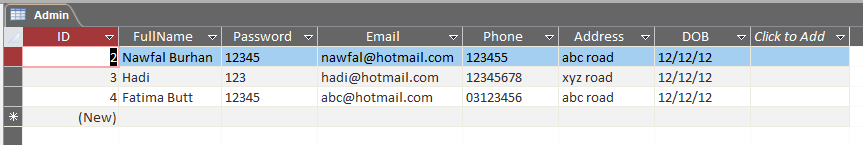


## Register Portal:

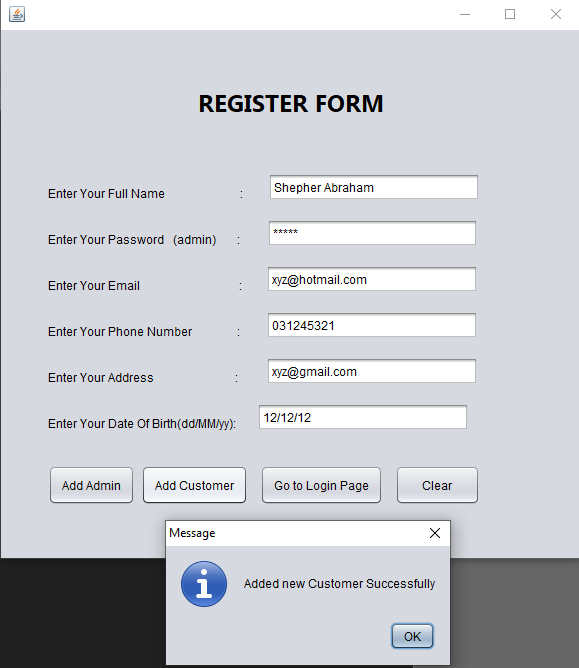


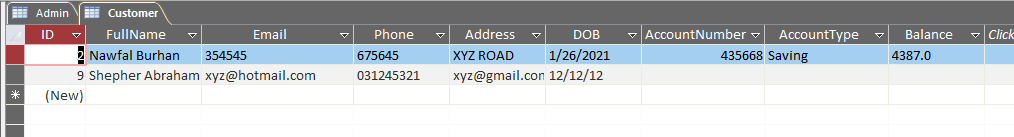
## Add Admin:





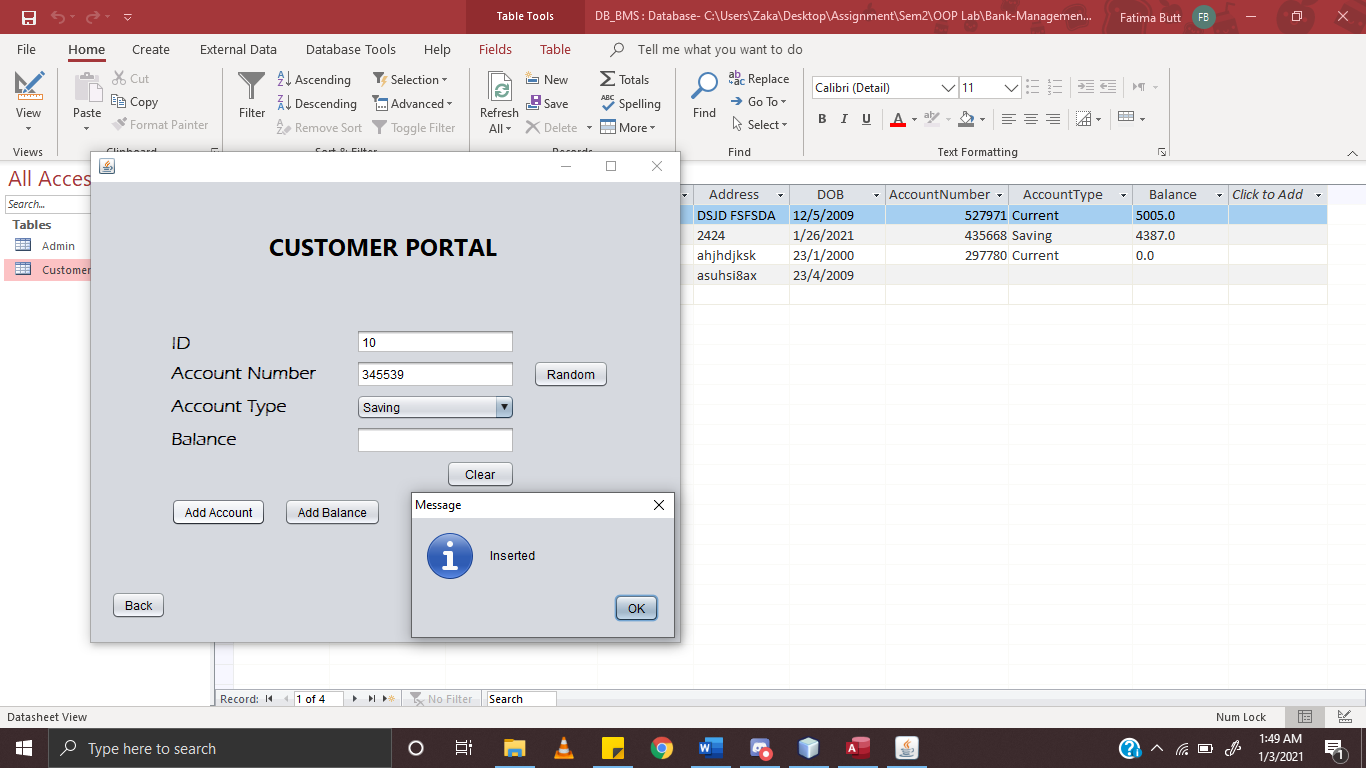
## Add Customer:

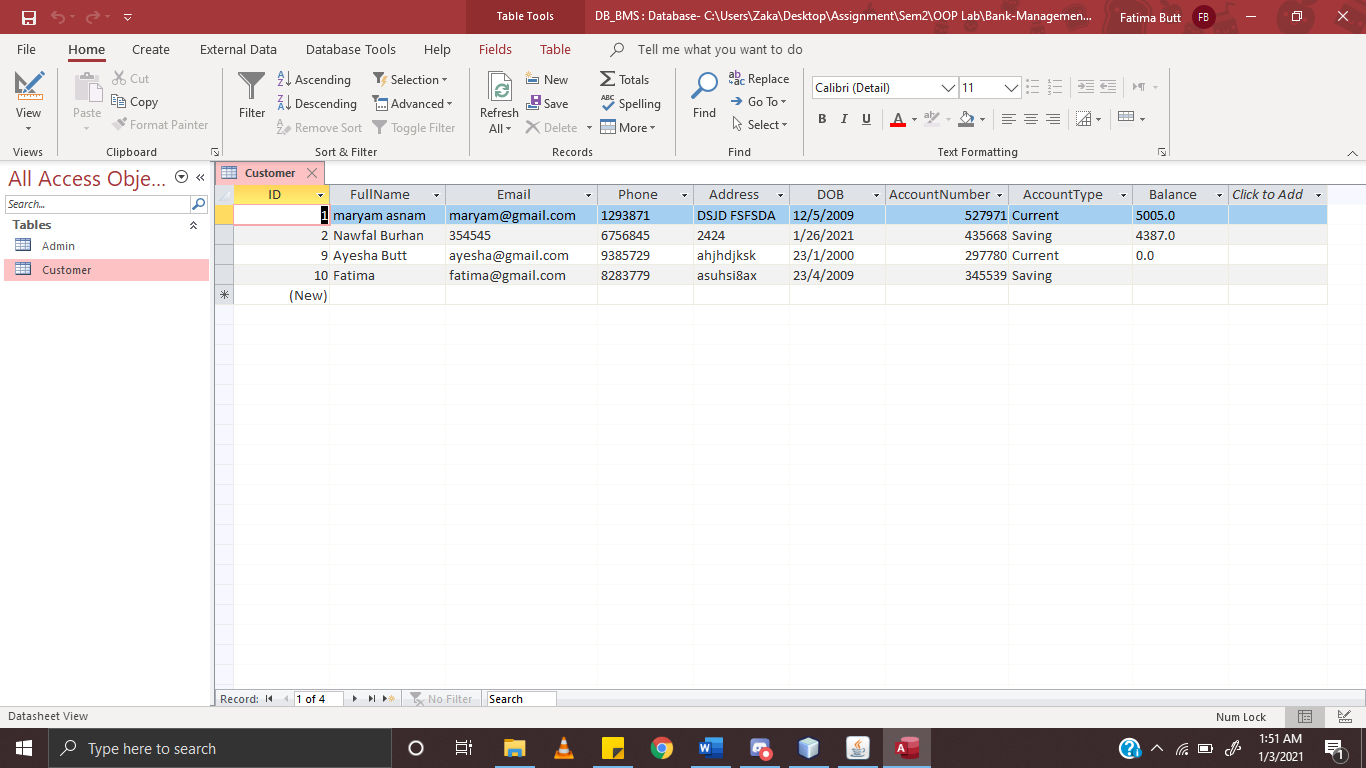




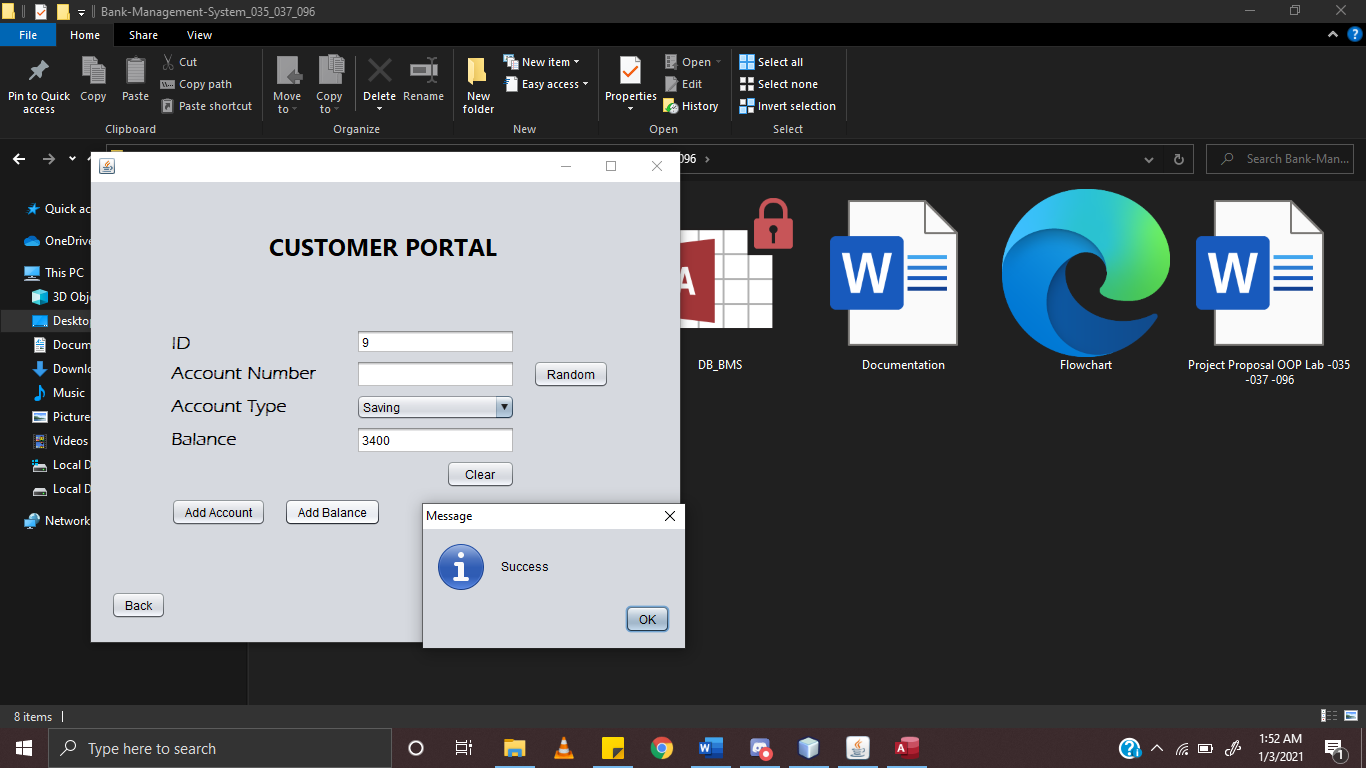
## Customer Portal:

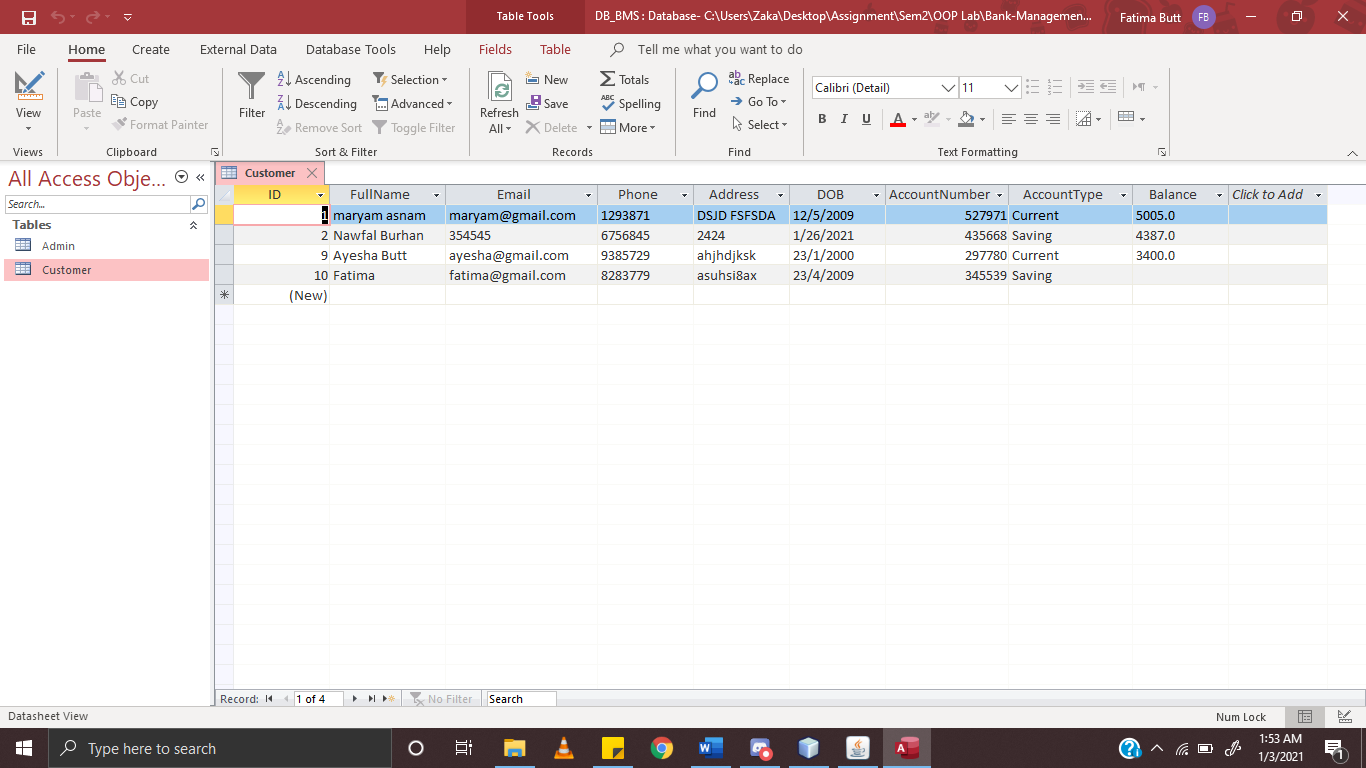
## Add Account:





## Add Balance:





# **UML Diagram:**

