**JAVA SWING BASED**

**LAYERING OF BANK ACCOUNT**

**-SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

***BACHELOR OF TECHNOLOGY***

***IN***

**INFORMATION TECHNOLOGY**

By

**KARTHIK RAMAVATH<1602-20-737-018>**

**Under the Guidance of**

**B. Leelavathy**



**Department of Information Technology****Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2021-2022**

# BONAFIDE CERTIFICATE

This to Certify that the project report titled

**“LAYERING OF BANK ACCOUNT”** project work of Ms. Karthik Ramavath bearing Roll.no:1602-20-737-018 who carried out this project under my supervision in the IV semester for the academic year 2021-2022.

*Signature*   *Signature*   *external examiner internal examiner*

ROLL NO: 1602-20-737-018

NAME: Karthik Ramavath [2]

## **ABSTRACT:**

## This is the second stage where the origins of the funds are concealed by moving them around in a series of complex bank transfers or financial transactions. Out of the various techniques of layering, the most common is to make electronic transfers between different jurisdictions and through offshore accounts. For instance, the criminal may wire the proceeds to a corporation in an overseas island where the money trail doesn’t end because the same corporation can be owned by a trust on another island.

**REQUIREMENT ANALYSIS:**

**LIST OF TABLES:**

* Branch
* Account
* Customer

**LIST OF ATTRIBUTES WITH THEIR DOMAIN TYPES:**

**BRANCH:**

* bid number(10)
* bname varchar(20)
* blocation varchar(20)

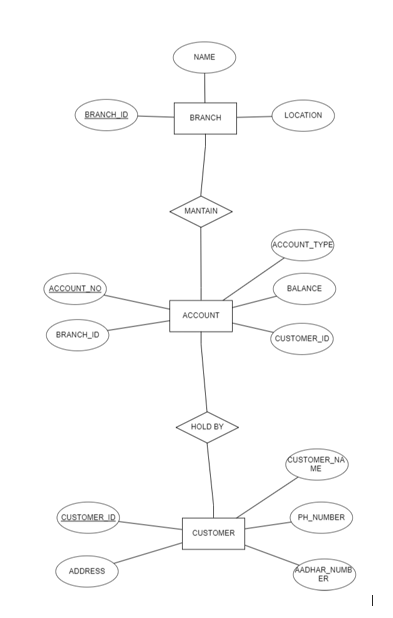
**ACCOUNT:**

* bid number
* acc\_no number
* acc\_type varchar2(20)
* balance number(5)

**CUSTOMER:**

* cid number
* cname varchar2(20)
* address varchar2(20)
* ph\_no number(10)
* aadhar\_num number(12)

**ER DIAGRAM:**



**RELATIONAL MODEL:**

**DDL OPERATIONS:**

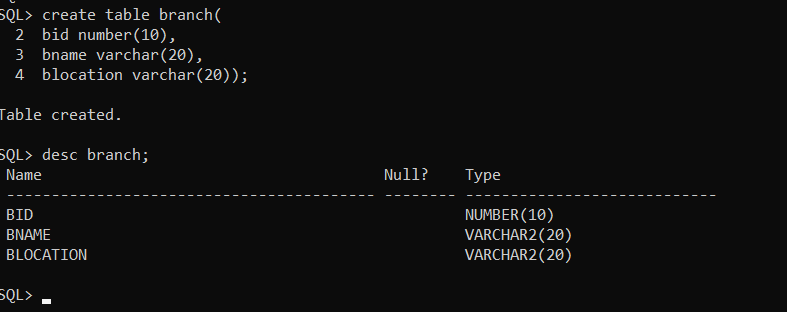
**BRANCH:**

create table branch(

bid number(10),

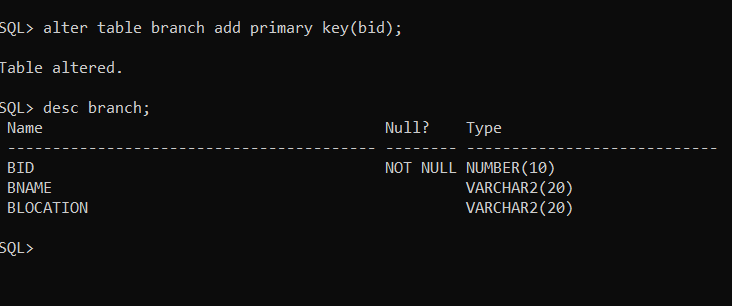
bname varchar(20),

blocation varchar(20));

****

**SETTING BID AS PRIMARY KEY:**

alter table branch add primary key(bid);



**ACCOUNT:**

create table account(

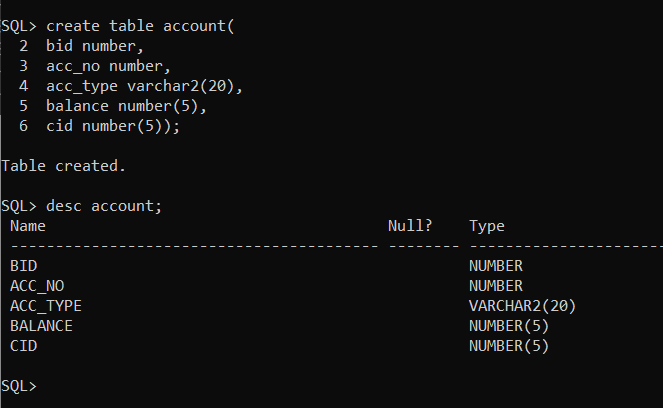
bid number,

acc\_no number,

acc\_type varchar2(20),

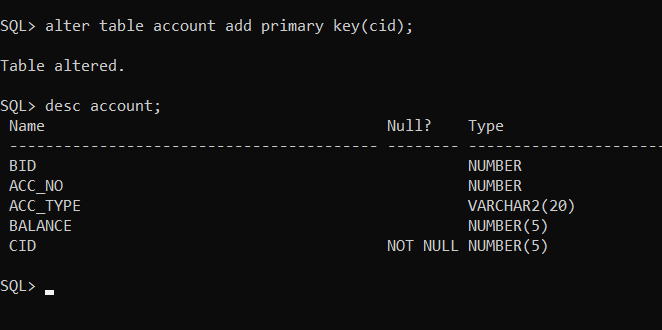
balance number(5),

cid number(5));



**SETTING CID AS PRIMARY KEY:**

alter table account add primary key(cid);



**CUSTOMER:**

create table customer(

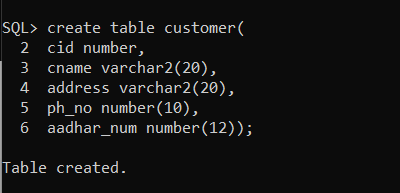
cid number,

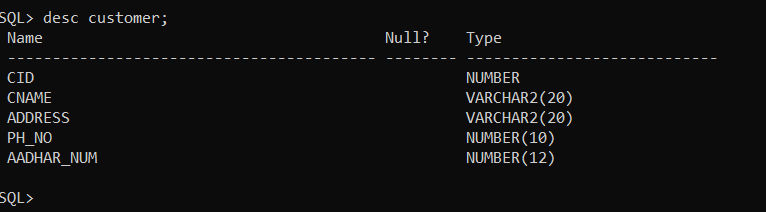
cname varchar2(20),

address varchar2(20),

ph\_no number(10),

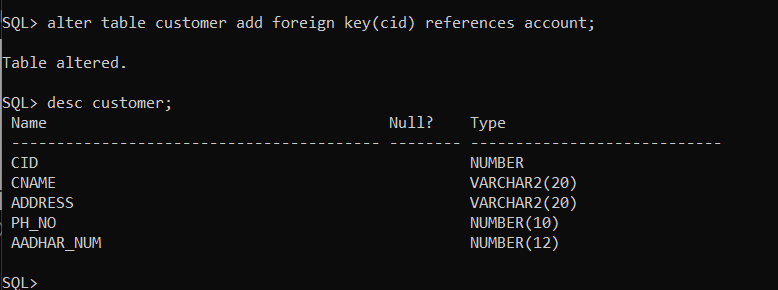
aadhar\_num number(12));





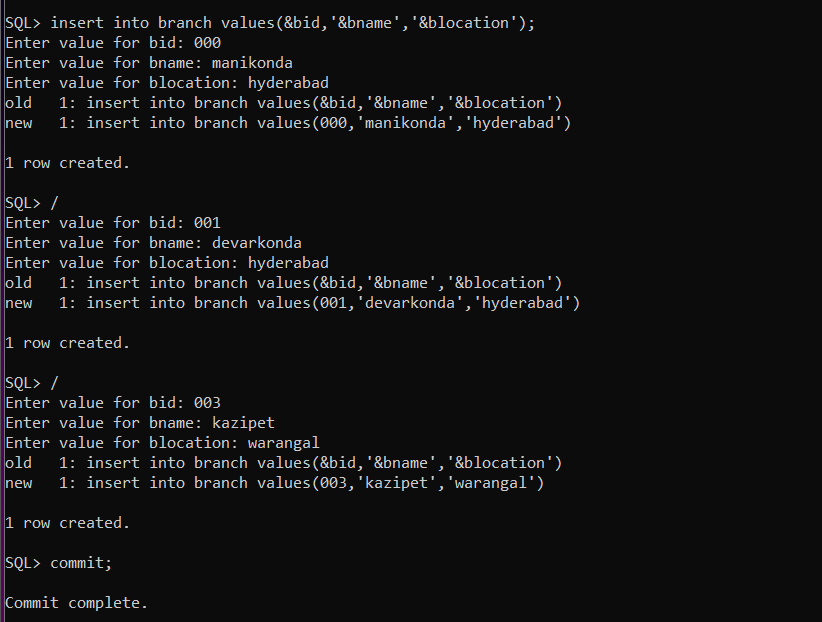
**SETTING CID AS FOREIGN KEY:**

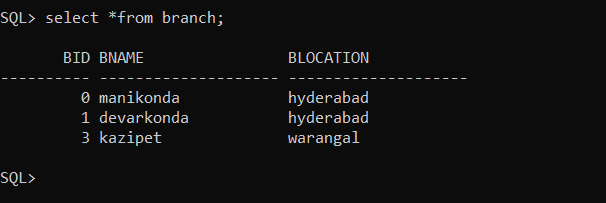
alter table customer add foreign key(cid) references account;



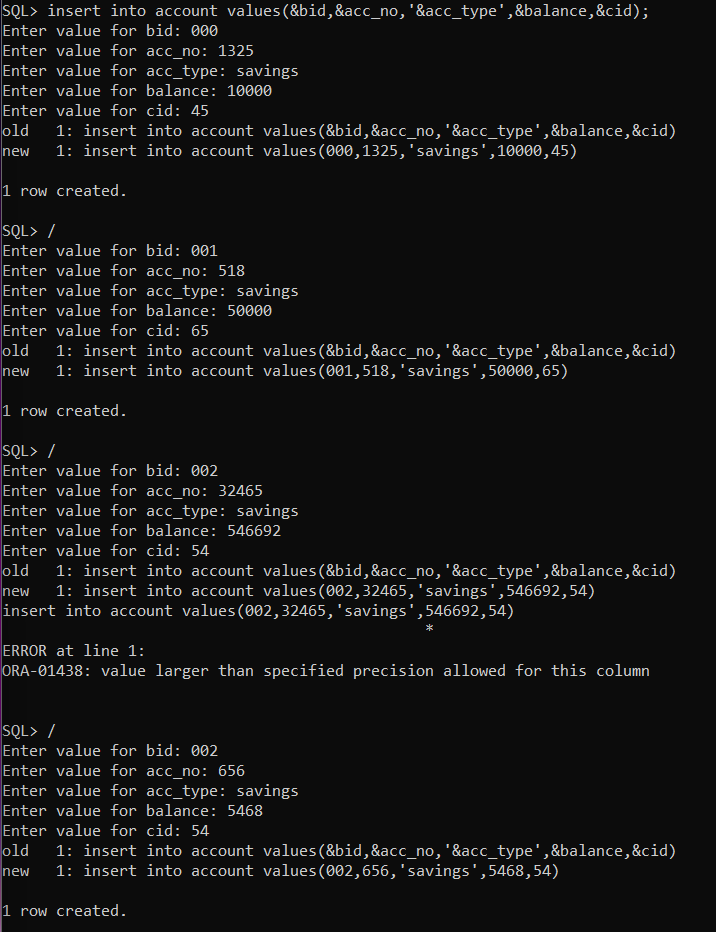
**DML OPERATIONS:**

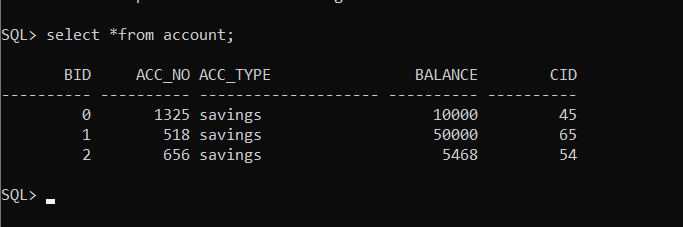
**Branch:**



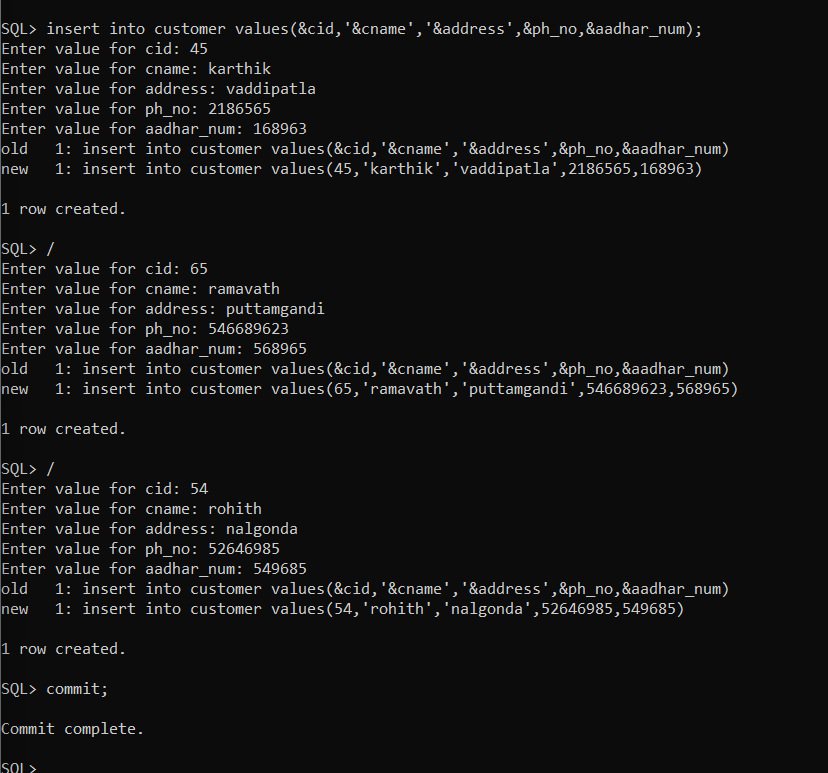


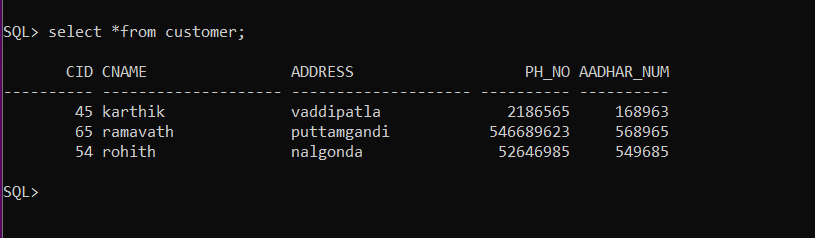
**Account:**





**Customer:**





**Implementation:**

Front end programs and its connectivity Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases. The connection to the database can be performed using Java programming (JDBC API) as:

import java.sql.\*;

public class TrialConnect{

public static void main(String[] args){

try{

Class.forName("oracle.jdbc.OracleDriver");

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","zoha","zoha");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from CUSTOMER");

while(rs.next())

System.out.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3));

con.close();

}

catch(Exception e){

System.out.println(e);

}

}

}

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

**PROGRAM:**

**BRANCH:**

import javax.swing.\*;

class BranchUI

{

JTextField t1,t2,t3;

JLabel l1,l2,l3;

JPanel p;

public BranchUI()

{

createComponents();

addComponents();

}

void createComponents()

{

t1 = new JTextField();

t1.setBounds(250,20,200,30);

t2 = new JTextField();

t2.setBounds(250,80,200,30);

t3 = new JTextField();

t3.setBounds(250,140,200,30);

l1 = new JLabel("Branch Name : ");

l1.setBounds(100,20,100,30);

l2 = new JLabel("Branch Id : ");

l2.setBounds(100,80,100,30);

l3 = new JLabel("Location : ");

l3.setBounds(100,140,100,30);

p = new JPanel(null);

p.setBounds(0,0,600,400);

}

void addComponents()

{

p.add(l1);

p.add(t1);

p.add(l2);

p.add(t2);

p.add(l3);

p.add(t3);

}

}

**ACCOUNT:**

import javax.swing.\*;

class AccountUI

{

JTextField t1,t2,t3,t4,t5;

JLabel l1,l2,l3,l4,l5;

JPanel p;

public AccountUI()

{

createComponents();

addComponents();

}

void createComponents()

{

t1 = new JTextField();

t1.setBounds(250,20,200,30);

t2 = new JTextField();

t2.setBounds(250,80,200,30);

t3 = new JTextField();

t3.setBounds(250,140,200,30);

t4 = new JTextField();

t4.setBounds(250,200,200,30);

t5 = new JTextField();

t5.setBounds(250,260,200,30);

l1 = new JLabel("Branch Id : ");

l1.setBounds(100,20,100,30);

l2 = new JLabel("Account Number : ");

l2.setBounds(100,80,100,30);

l3 = new JLabel("Account Type : ");

l3.setBounds(100,140,100,30);

l4 = new JLabel("Balance : ");

l4.setBounds(100,200,100,30);

l5 = new JLabel("Customer ID : ");

l5.setBounds(100,260,100,30);

p = new JPanel(null);

p.setBounds(0,0,600,400);

}

void addComponents()

{

p.add(l1);

p.add(t1);

p.add(l2);

p.add(t2);

p.add(l3);

p.add(t3);

p.add(l4);

p.add(t4);

p.add(l5);

p.add(t5);

}

}

**CUSTOMER:**

import javax.swing.\*;

class CustomerUI

{

JTextField t1,t2,t3,t4,t5,t6;

JLabel l1,l2,l3,l4,l5,l6;

JPanel p;

public CustomerUI()

{

createComponents();

addComponents();

}

void createComponents()

{

t1 = new JTextField();

t1.setBounds(250,20,200,30);

t2 = new JTextField();

t2.setBounds(250,80,200,30);

t3 = new JTextField();

t3.setBounds(250,140,200,30);

t4 = new JTextField();

t4.setBounds(250,200,200,30);

t5 = new JTextField();

t5.setBounds(250,260,200,30);

l1 = new JLabel("Customer Name : ");

l1.setBounds(100,20,100,30);

l2 = new JLabel("Customer Id : ");

l2.setBounds(100,80,100,30);

l3 = new JLabel("Phone Number : ");

l3.setBounds(100,140,100,30);

l4 = new JLabel("Aadhar Number : ");

l4.setBounds(100,200,100,30);

l5 = new JLabel("Address : ");

l5.setBounds(100,260,100,30);

p = new JPanel(null);

p.setBounds(0,0,600,400);

}

void addComponents()

{

p.add(l1);

p.add(t1);

p.add(l2);

p.add(t2);

p.add(l3);

p.add(t3);

p.add(l4);

p.add(t4);

p.add(l5);

p.add(t5);

}

}

**MAIN:**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class MainUI extends JFrame implements ActionListener

{

BranchUI ob1;

AccountUI ob2;

CustomerUI ob3;

JButton submit,modify,delete,m1,m2,m3;

JPanel p1,p2,p3,pb;

JMenuBar mb;

public MainUI()

{

setSize(600,550);

setLayout(null);

setVisible(true);

setTitle("Layering of Bank Account");

ob1 = new BranchUI();

ob2 = new AccountUI();

ob3 = new CustomerUI();

createPanels();

createMenu();

createButtons();

addComponents();

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

void createPanels()

{

p1 = ob1.p;

p2 = ob2.p;

p3 = ob3.p;

pb = new JPanel(new FlowLayout(FlowLayout.CENTER,50,0));

pb.setBounds(0,400,600,150);

}

void createMenu()

{

mb = new JMenuBar();

m1 = new JButton("Branch");

m1.setFocusable(false);

m2 = new JButton("Account");

m2.setFocusable(false);

m3 = new JButton("Customer");

m3.setFocusable(false);

m1.addActionListener(this);

m2.addActionListener(this);

m3.addActionListener(this);

mb.add(m1);

mb.add(m2);

mb.add(m3);

}

public void actionPerformed(ActionEvent e)

{

remove(p1);

remove(p2);

remove(p3);

if(e.getSource()==m1)

add(p1);

else if(e.getSource()==m2)

add(p2);

else

add(p3);

}

void createButtons()

{

submit = new JButton("Submit");

submit.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

JOptionPane.showMessageDialog(new JFrame(),"Successfully Inserted!","NOTICE",JOptionPane.INFORMATION\_MESSAGE);

}

});

modify = new JButton("Modify");

modify.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

JOptionPane.showMessageDialog(new JFrame(),"Successfully Modified!","NOTICE",JOptionPane.INFORMATION\_MESSAGE);

}

});

delete = new JButton("Delete");

delete.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

JOptionPane.showMessageDialog(new JFrame(),"Successfully Deleted!","NOTICE",JOptionPane.INFORMATION\_MESSAGE);

}

});

pb.add(submit);

pb.add(modify);

pb.add(delete);

}

void addComponents()

{

add(p1);

add(pb);

setJMenuBar(mb);

}

public static void main(String a[])

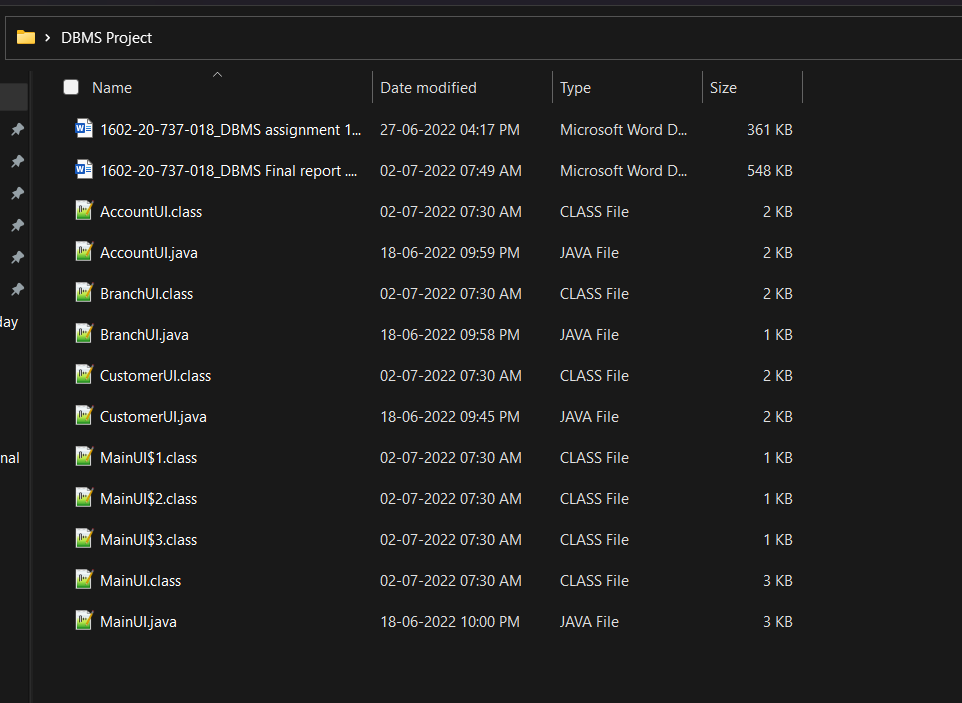
{

new MainUI();

}

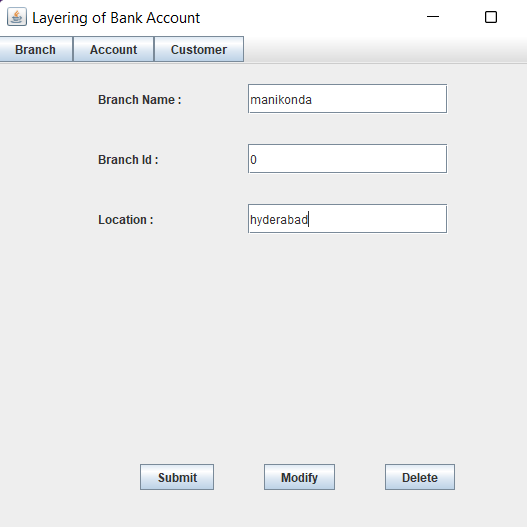
}

**FOLDER STRUCTURE:**

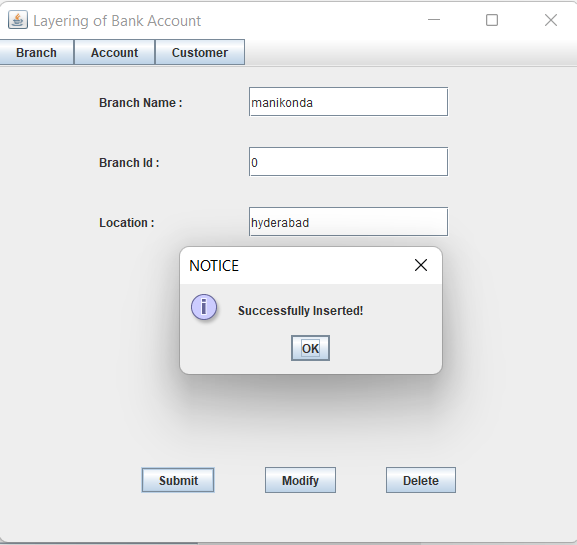
****

**Testing:**

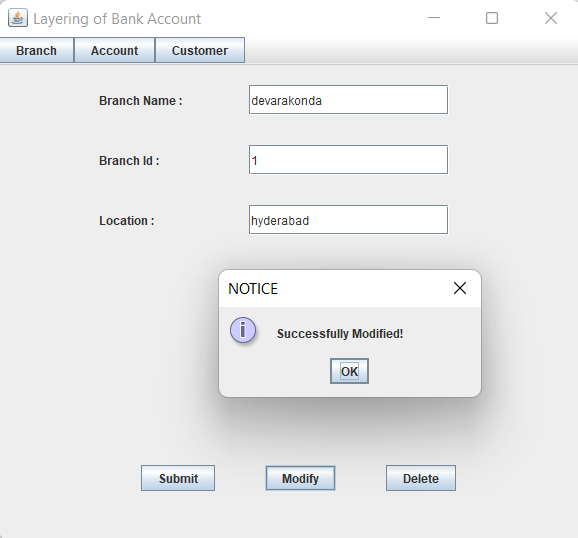
**BRANCH:**

****

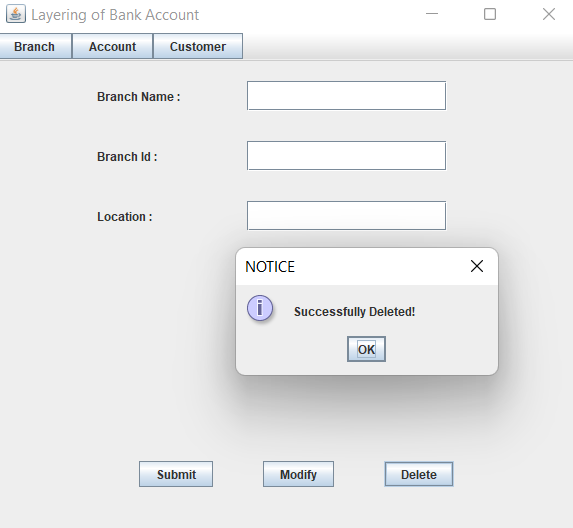
**SUBMIT:**



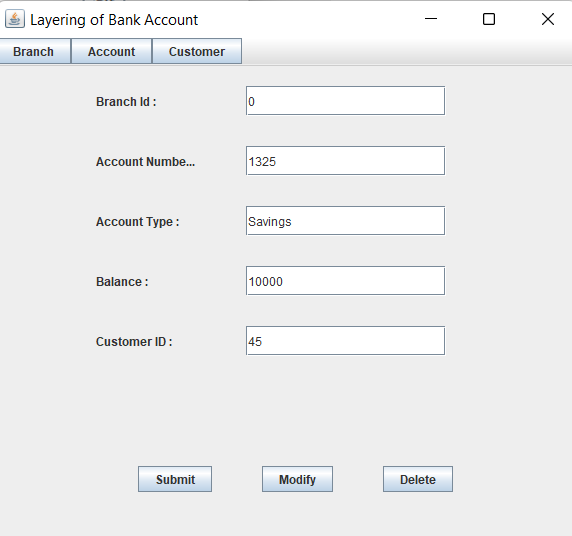
**MODIFY:**



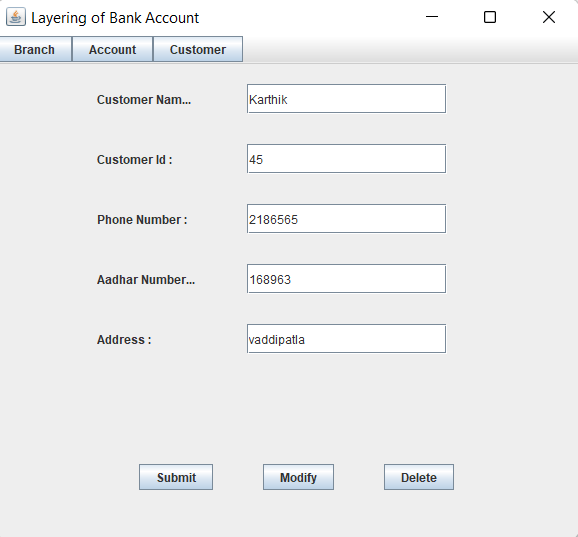
**DELETE:**



**ACCOUNT:**



**CUSTOMER:**

****

**Results:** I had successfully completed PROJECT on “LAYERING OF BANK ACOUNT”

**LINKS:**

**https://github.com/Karthik-Ramavath/Layering-of-Bank-Account**

**Discussion and future Work:**

Layering is essentially the use of placement and extraction over again, using varying amounts each time, to make transactions as hard as possible. Using this application we can Integrate all bank accounts related to a user through Aadhar identity.

**CONCLUSION:**

Thus, a Java SWING based ***LAYERING OF BANK ACCOUNT*** is created which is connected to the Oracle 11g database. Therefore, all the entries and details are directly updated on their respective tables created in the database