JAVA SWING BASED – COURT MANAGEMENT SYSTEM - SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfillment of the
Requirements for the award of the Degree of
BACHELOR OF TECHNOLOGY
IN

INFORMATION TECHNOLOGY

By

APPALI KRISHNA CHAITANYA <1602-19-737-308>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020-2021

BONAFIDE CERTIFICATE

This to Certify that the project report titled **“COURT MANAGEMENT SYSTEM”** project work of Mr.APPALI KRISHNA CHAITANYA bearing Roll.no:1602-19-737-308 who carried out this project under my supervision in the IV semester for the academic year 2020-2021.

Signature	*Signature*
external examiner	*internal examiner*

Court Management System

A KRISHNA CHAITANYA
IT-B
1602-19-737-308
DBMS

ABSTRACT:-

The Court Management System is a Design framework for computerizing the legal administrations in India. Utilizing this framework, offended party can without much of a stretch offer grievance to the concerned expert and they can see status of their cases and so forth. This framework will decrease the time and hazard to the clients contrast and manual framework. The undertaking performs expected capacity with required accuracy. Considering the current scenario (pandemic situation) this app is very much useful for different sections of people and a lot of physical process has been reduced.

REQUIREMENT ANALYSIS

List of tables:

- Client
- Advocate
- Types
- Court
- Slot
- Booking
- Selects
- Has
- Has_Timings

List of attributes with their domain type:-

Client:

Id of the client: client_id - number
Name of the client: client_name - varchar2(20)
Password: client_pass - varchar2(20)

Advocate:

Id of the advocate: advocate_id - number
Name of the advocate: advocate_name - varchar2(20)
Rating of the advocate: advocate_rating - varchar2(20)

Types:

Type of advocate: advocate_type - varchar2(20)
Experience: experience – number
Advocate type id: advocate_type_id - varchar2(20)

Court:

Pincode: pincode - number
State name: state_name - varchar2(20)
City name: city_name - varchar2(20)

Slot:

Id of the slot: slot_id – varchar2(20)
Starting date: from_date - varchar2(10)
Approximate duration: tentative_years - number
Name of the judge: judge_name - varchar2(20)

THROUGH THE PROJECT:

This project helps everyone to easily select advocate without going out from home. Selection is also made easy by giving experience and rating to each advocate. This project also tells the user tentative period of much long their case might take. In the period of digitalization where everything can be done from home this project helps in reduction of a lot of work for the individuals.

ARCHITECTURE AND TECHNOLOGY USED:

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL Plus.

Java SWING:

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the
earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Java-SQL Connectivity using JDBC:

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.

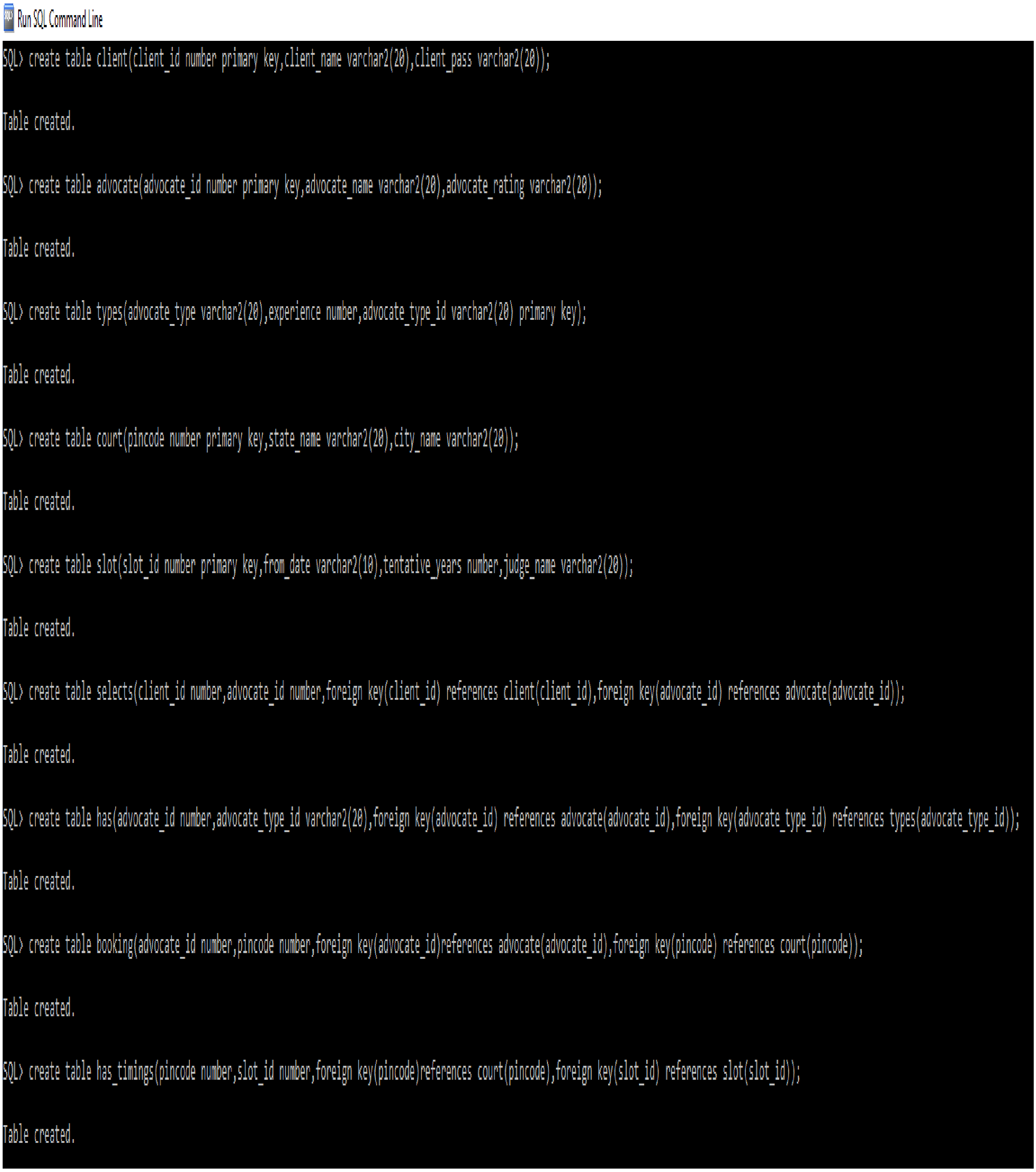
The connection to the database can be performed using Java programming (JDBC API) as:

```
try

{

        Class.forName(“oracle.jdbc.driver.OracleDriver”);

}

catch(Exception e)

{

        System.err.println(“Unable to find and load driver”);

        System.exit(1);

}

public void connectToDB()

{

        try

        {
                connection =
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","chaitanya","vas
        avi");

        statement = connection.createStatement();

        }

        catch(SQLException connectException)

        {
```

```
                    System.out.println(connectException.getMessage());

                    System.out.println(connectException.getSQLState());

                    System.out.println(connectException.getErrorCode());

                    System.exit(1);

          }

}
```

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

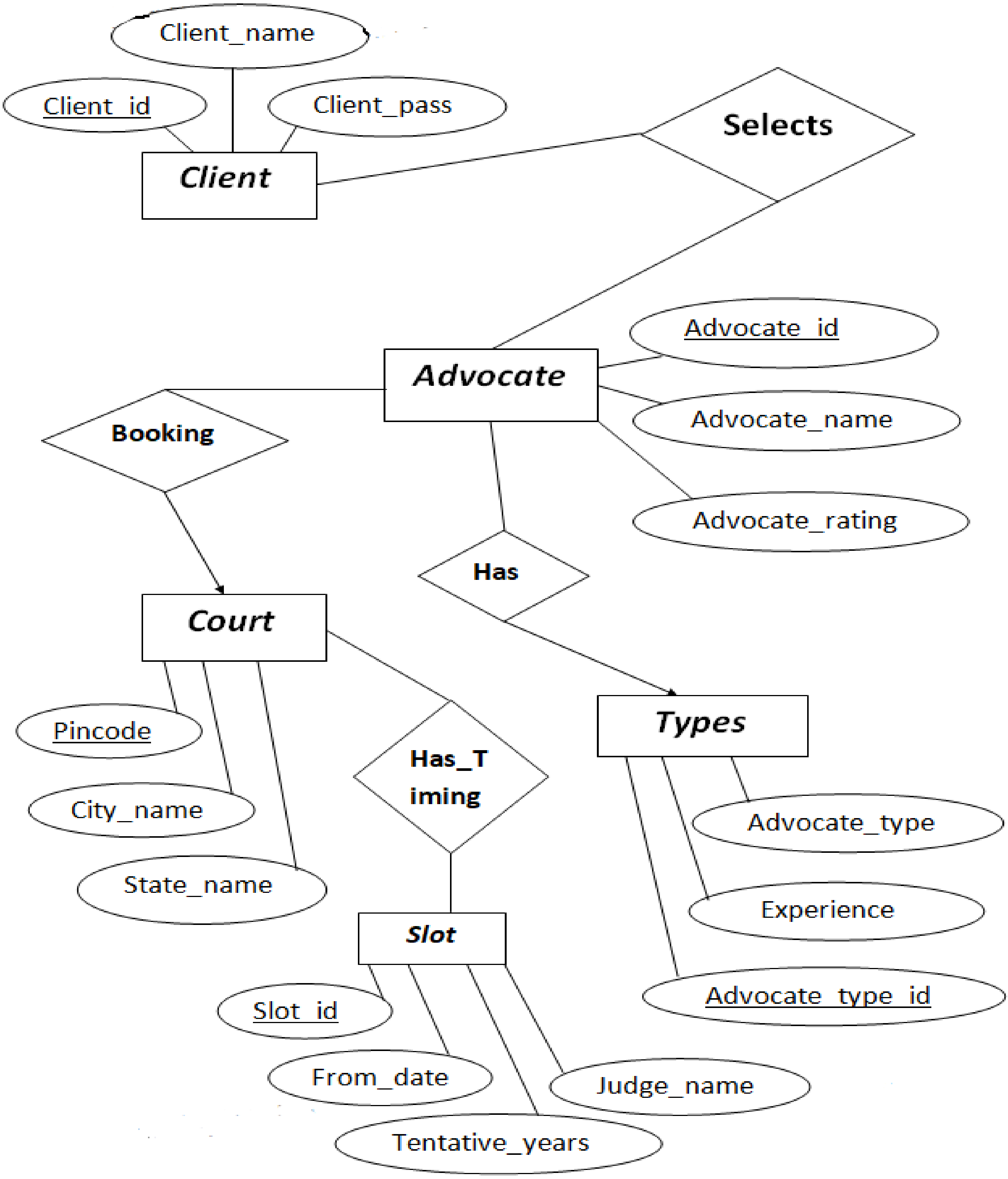
Table Created in SQL for above mentioned purpose is as:



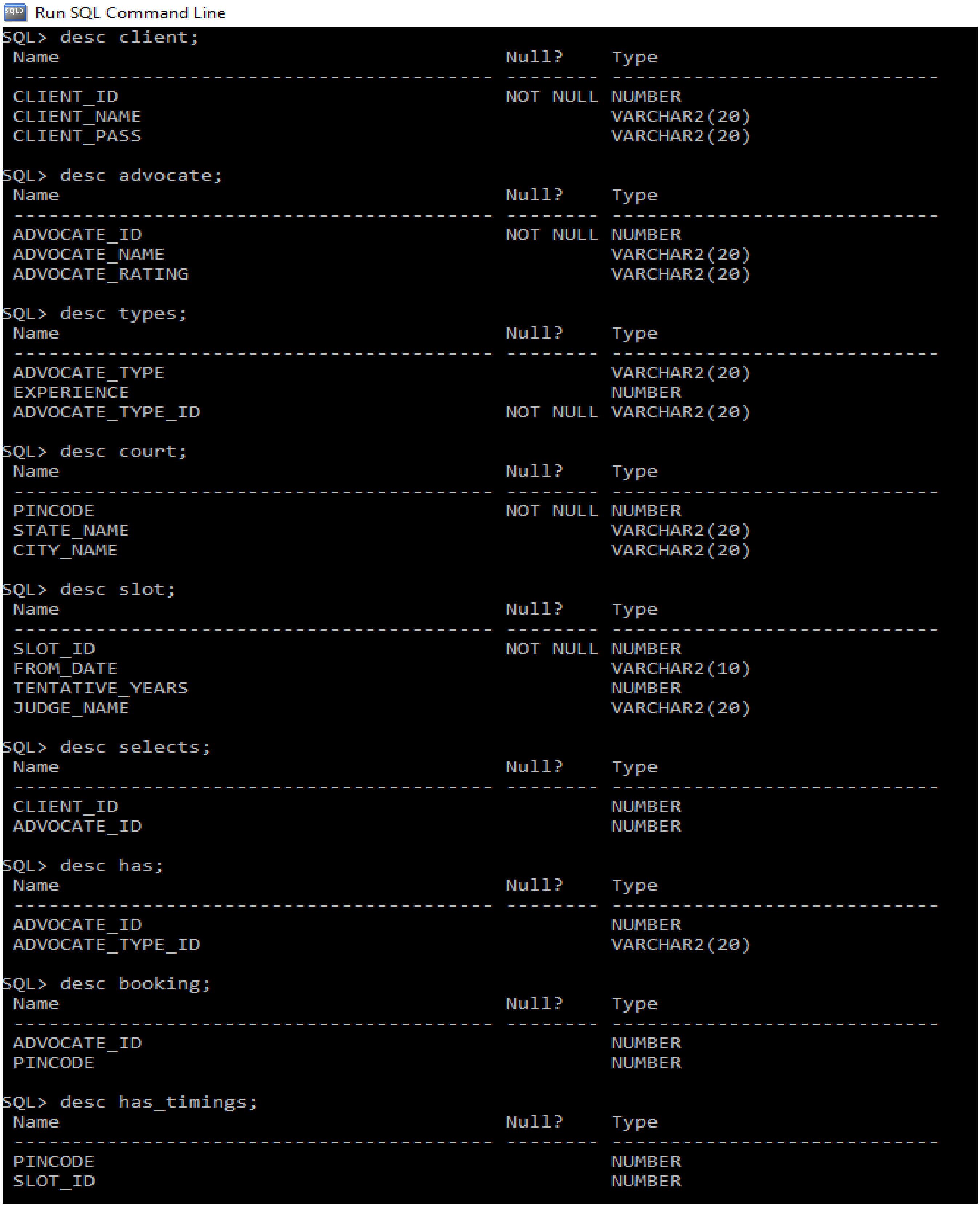
```
SQL> create table client(client_id number primary key,client_name varchar2(20),client_pass varchar2(20));

Table created.

SQL> create table advocate(advocate_id number primary key,advocate_name varchar2(20),advocate_rating varchar2(20));

Table created.

SQL> create table types(advocate_type varchar2(20),experience number,advocate_type_id varchar2(20) primary key);

Table created.

SQL> create table court(pincode number primary key,state_name varchar2(20),city_name varchar2(20));

Table created.

SQL> create table slot(slot_id number primary key,from_date varchar2(10),tentative_years number,judge_name varchar2(20));

Table created.

SQL> create table selects(client_id number,advocate_id number,foreign key(client_id) references client(client_id),foreign key(advocate_id) references advocate(advocate_id));

Table created.

SQL> create table has(advocate_id number,advocate_type_id varchar2(20),foreign key(advocate_id) references advocate(advocate_id),foreign key(advocate_type_id) references types(advocate_type_id));

Table created.

SQL> create table booking(advocate_id number,pincode number,foreign key(advocate_id)references advocate(advocate_id),foreign key(pincode) references court(pincode));

Table created.

SQL> create table has_timings(pincode number,slot_id number,foreign key(pincode)references court(pincode),foreign key(slot_id) references slot(slot_id));

Table created.
```

DESIGN:

ER DIAGRAM:



Database Design:



```
SQL> desc client;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_ID                                 NOT NULL NUMBER
 CLIENT_NAME                                        VARCHAR2(20)
 CLIENT_PASS                                        VARCHAR2(20)

SQL> desc advocate;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADVOCATE_ID                               NOT NULL NUMBER
 ADVOCATE_NAME                                      VARCHAR2(20)
 ADVOCATE_RATING                                    VARCHAR2(20)

SQL> desc types;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADVOCATE_TYPE                                      VARCHAR2(20)
 EXPERIENCE                                         NUMBER
 ADVOCATE_TYPE_ID                          NOT NULL VARCHAR2(20)

SQL> desc court;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PINCODE                                   NOT NULL NUMBER
 STATE_NAME                                         VARCHAR2(20)
 CITY_NAME                                          VARCHAR2(20)

SQL> desc slot;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SLOT_ID                                   NOT NULL NUMBER
 FROM_DATE                                          VARCHAR2(10)
 TENTATIVE_YEARS                                    NUMBER
 JUDGE_NAME                                         VARCHAR2(20)

SQL> desc selects;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENT_ID                                          NUMBER
 ADVOCATE_ID                                        NUMBER

SQL> desc has;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADVOCATE_ID                                        NUMBER
 ADVOCATE_TYPE_ID                                   VARCHAR2(20)

SQL> desc booking;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADVOCATE_ID                                        NUMBER
 PINCODE                                            NUMBER

SQL> desc has_timings;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PINCODE                                            NUMBER
 SLOT_ID                                            NUMBER
```

Implementation:

Program:

Main.java:

```
package CourtManagementSystem;

public class Main {
      public static void main(String []args)
      {
            new ProjectDemo();
      }

}
```

OracleCon.java:

```
package CourtManagementSystem;


import java.sql.*;

public class OracleCon {

public static void main(String args[]){

            try{
                  //step1 load the driver class
                  Class.forName("oracle.jdbc.driver.OracleDriver");   //
or Class.forName("oracle.jdbc.OracleDriver");

                  //step2 create  the connection object
                  Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","cha
itanya","vasavi");

                  //step3 create the statement object
                  Statement stmt=con.createStatement();

                  //step4 execute query
                  ResultSet rs=stmt.executeQuery("select * from sailors");
                  while(rs.next())
                              System.out.println(rs.getInt(1)+"
"+rs.getString(2)+"  "+rs.getString(3));

                  //step5 close the connection object
                  con.close();

                  }catch(Exception e){ System.out.println(e);}

      }
}
```

ProjectDemo.java:

```
package CourtManagementSystem;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
//import java.awt.Image;
//import java.awt.Toolkit;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

//import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JMenuItem;
//import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class ProjectDemo extends JFrame {

            private static final long serialVersionUID = 1L;
            private JMenu
mnclient,mnselects,mnadvo,mnhas,mnbook,mntypes,mncourt,mnhastim,mnslot;
            private JMenuBar mnBar;
            private JMenuItem miSubmit1,miModify1,miDelete1,miView1;
            private JMenuItem miSubmit2,miModify2,miDelete2,miView2;
            private JMenuItem miSubmit3,miModify3,miDelete3,miView3;
            private JMenuItem miSubmit4,miModify4,miDelete4,miView4;
            private JMenuItem miSubmit5,miModify5,miDelete5,miView5;
            private JMenuItem miSubmit6,miModify6,miDelete6,miView6;
            private JMenuItem miSubmit7,miModify7,miDelete7,miView7;
            private JMenuItem miSubmit8,miModify8,miDelete8,miView8;
            private JMenuItem miSubmit9,miModify9,miDelete9,miView9;
            private JTextField txtField;

            static JPanel p1;
            int x=0,y=100;

            void initialize()
            {
                  p1 = new JPanel();
                  mnclient = new JMenu("Client");
                  mnselects = new JMenu("Selects");
                  mnadvo = new JMenu("Advocate");
                  mnhas = new JMenu("Has");
                  mnbook = new JMenu("Booking");
                  mntypes = new JMenu("Types");
                  mncourt = new JMenu("Court");
                  mnhastim = new JMenu("Has_Timings");
                  mnslot = new JMenu("Slot");
                  mnBar = new JMenuBar();

                  miSubmit1 = new JMenuItem("Submit");
                  miModify1 = new JMenuItem("Modify");
                  miDelete1 = new JMenuItem("Delete");
                  miView1 = new JMenuItem("View");

                  miSubmit2 = new JMenuItem("Submit");
                  miModify2 = new JMenuItem("Modify");
                  miDelete2 = new JMenuItem("Delete");
                  miView2 = new JMenuItem("View");

                  miSubmit3 = new JMenuItem("Submit");
                  miModify3 = new JMenuItem("Modify");
                  miDelete3 = new JMenuItem("Delete");
                  miView3 = new JMenuItem("View");

                  miSubmit4 = new JMenuItem("Submit");
                  miModify4 = new JMenuItem("Modify");
                  miDelete4 = new JMenuItem("Delete");
                  miView4 = new JMenuItem("View");

                  miSubmit5 = new JMenuItem("Submit");
                  miModify5 = new JMenuItem("Modify");
                  miDelete5 = new JMenuItem("Delete");
                  miView5 = new JMenuItem("View");

                  miSubmit6 = new JMenuItem("Submit");
                  miModify6 = new JMenuItem("Modify");
                  miDelete6 = new JMenuItem("Delete");
                  miView6 = new JMenuItem("View");

                  miSubmit7 = new JMenuItem("Submit");
                  miModify7 = new JMenuItem("Modify");
                  miDelete7 = new JMenuItem("Delete");
                  miView7 = new JMenuItem("View");

                  miSubmit8 = new JMenuItem("Submit");
                  miModify8 = new JMenuItem("Modify");
                  miDelete8 = new JMenuItem("Delete");
                  miView8 = new JMenuItem("View");

                  miSubmit9 = new JMenuItem("Submit");
                  miModify9 = new JMenuItem("Modify");
                  miDelete9 = new JMenuItem("Delete");
                  miView9 = new JMenuItem("View");

                  txtField = new JTextField("        Court Management System            ");
                  txtField.setFont(new Font("Serif",Font.PLAIN,42));
                  txtField.setBackground(Color.cyan);
                  txtField.setEditable(false);
            }
```

```
	void addComponentsToFrame()
	{
		mnclient.add(miSubmit1);
		mnclient.add(miModify1);
		mnclient.add(miDelete1);
		mnclient.add(miView1);

		mnselects.add(miSubmit2);
		mnselects.add(miModify2);
		mnselects.add(miDelete2);
		mnselects.add(miView2);

		mnadvo.add(miSubmit3);
		mnadvo.add(miModify3);
		mnadvo.add(miDelete3);
		mnadvo.add(miView3);

		mnhas.add(miSubmit4);
		mnhas.add(miModify4);
		mnhas.add(miDelete4);
		mnhas.add(miView4);

		mnbook.add(miSubmit5);
		mnbook.add(miModify5);
		mnbook.add(miDelete5);
		mnbook.add(miView5);

		mntypes.add(miSubmit6);
		mntypes.add(miModify6);
		mntypes.add(miDelete6);
		mntypes.add(miView6);

		mncourt.add(miSubmit7);
		mncourt.add(miModify7);
		mncourt.add(miDelete7);
		mncourt.add(miView7);

		mnhastim.add(miSubmit8);
		mnhastim.add(miModify8);
		mnhastim.add(miDelete8);
		mnhastim.add(miView8);

		mnslot.add(miSubmit9);
		mnslot.add(miModify9);
		mnslot.add(miDelete9);
		mnslot.add(miView9);

		mnBar.add(mnclient);
		mnBar.add(mnselects);
		mnBar.add(mnadvo);
		mnBar.add(mnhas);
		mnBar.add(mnbook);
		mnBar.add(mntypes);
		mnBar.add(mncourt);
		mnBar.add(mnhastim);
		mnBar.add(mnslot);
		setJMenuBar(mnBar);

		p1.setLayout(new BorderLayout());
		p1.add(txtField,BorderLayout.CENTER);
		this.setLayout(new BorderLayout());
		add(p1,BorderLayout.CENTER);
	}

	void register()
	{
			ClientCMS c1 = new
ClientCMS(p1,ProjectDemo.this,miSubmit1,miModify1,miDelete1,miView1);
			c1.reg_client();
			SelectsCMS c2 = new
SelectsCMS(p1,ProjectDemo.this,miSubmit2,miModify2,miDelete2,miView2);
			c2.reg_selects();
			AdvocateCMS c3 = new
AdvocateCMS(p1,ProjectDemo.this,miSubmit3,miModify3,miDelete3,miView3);
			c3.reg_advocate();
			HasCMS c4 = new
HasCMS(p1,ProjectDemo.this,miSubmit4,miModify4,miDelete4,miView4);
			c4.reg_has();
			BookingCMS c5 = new
BookingCMS(p1,ProjectDemo.this,miSubmit5,miModify5,miDelete5,miView5);
			c5.reg_booking();
			TypesCMS c6 = new
TypesCMS(p1,ProjectDemo.this,miSubmit6,miModify6,miDelete6,miView6);
			c6.reg_types();
			CourtCMS c7 = new
CourtCMS(p1,ProjectDemo.this,miSubmit7,miModify7,miDelete7,miView7);
			c7.reg_court();
			HasTimingCMS c8 = new
HasTimingCMS(p1,ProjectDemo.this,miSubmit8,miModify8,miDelete8,miView8);
			c8.reg_hastiming();
			SlotCMS c9 = new
SlotCMS(p1,ProjectDemo.this,miSubmit9,miModify9,miDelete9,miView9);
			c9.reg_slot();

			addWindowListener(new WindowAdapter(){
			  public void windowClosing(WindowEvent evt)
			  {
				int a =
JOptionPane.showConfirmDialog(ProjectDemo.this, "Are you sure?","This will
close the UI",JOptionPane.OK_CANCEL_OPTION,JOptionPane.QUESTION_MESSAGE);
				if(a==0)
				{

ProjectDemo.this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
				}
```

```
                               else if(a==2)
                               {
                                     ProjectDemo.this.invalidate();
                                     ProjectDemo.this.validate();
                                     ProjectDemo.this.repaint();
                               }
                         }
                       });
            }

            public ProjectDemo()
            {
                  initialize();
                  addComponentsToFrame();
                  register();
                  pack();
                  setTitle("Court Management System");
                  setSize(600,500);
                  setVisible(true);
            }
}
```

Client.java:

```
package CourtManagementSystem;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class ClientCMS {

        //client_id                                   NOT NULL number
        //client_name                                          varchar2(20)
        //client_pass                                          varchar2(20)

        private JPanel p1;
        private JFrame frame;

        private JMenuItem miSubmit1,miModify1,miDelete1,miView1;
        private JLabel lblclient_id,lblclient_name,lblclient_pass;
        private JTextField txtclient_id,txtclient_name,txtclient_pass;
        private JButton btn;
        private JTextArea txtmsg;
        private Connection connection;
        private Statement statement;
        public ClientCMS(JPanel p1,JFrame frame,JMenuItem miSubmit1,JMenuItem
miModify1,JMenuItem miDelete1,JMenuItem miView1) {
                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connectToDB();
                this.frame=frame;
                this.p1=p1;
                this.miSubmit1=miSubmit1;
                this.miDelete1=miDelete1;
                this.miModify1=miModify1;
                this.miView1=miView1;
                txtclient_id=new JTextField(20);
                txtclient_name=new JTextField(20);
                txtclient_pass=new JTextField(20);
                txtmsg=new JTextArea(8,50);
                lblclient_id=new JLabel("Client Id:");
                lblclient_name=new JLabel("Client Name:");
                lblclient_pass=new JLabel("Client Password:");
                //queryHandler();
        }
        public void connectToDB()
   {
                try
                {
                  connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","lasya","lasya");
                  statement = connection.createStatement();

                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
   }
        public void reg_client() {
                miSubmit1.addActionListener(new ActionListener() {
                         public void actionPerformed(ActionEvent ae) {
                                p1.removeAll();
                                frame.invalidate();
                                frame.validate();
                                frame.repaint();

                                JPanel p=new JPanel();
                                txtclient_id=new JTextField(20);
                                txtclient_name=new JTextField(20);
                                txtclient_pass=new JTextField(20);
                                txtmsg=new JTextArea(8,50);
                                btn=new JButton();
                                //a grid of lbl and txtfield
                                p.add(lblclient_id);
                                p.add(txtclient_id);
                                p.add(lblclient_name);
                                p.add(txtclient_name);
                                p.add(lblclient_pass);
                                p.add(txtclient_pass);
                                p.setLayout(new GridLayout(5,2));

                                p1.add(p);
                                p1.setBackground(Color.green);
                                p1.add(btn);
                                p1.add(txtmsg);//msg text area added to panel
                                btn.setText("SUBMIT");
                                p1.setLayout(new FlowLayout());
                                frame.add(p1,BorderLayout.CENTER);
                                frame.validate();

                                //register listener
                                btn.addActionListener(new ActionListener() {

                                    @Override
                                    public void actionPerformed(ActionEvent e) {
                                        // TODO Auto-generated method stub
                                        try {
                                            Statement statement =
connection.createStatement();
                                            String query= "INSERT INTO client
VALUES(" + txtclient_id.getText() + ", " + "'" + txtclient_name.getText() + "'," + "'" +
txtclient_pass.getText() + "'" + ")";
                                            int i = statement.executeUpdate(query);
                                                txtmsg.append("\nInserted " + i +
" rows successfully");
                                        } catch (SQLException e1) {
                                            // TODO Auto-generated catch block

                                            txtmsg.append(e1.getMessage());
                                        }
                                    }
                                });

                        }
                });

                miModify1.addActionListener(new ActionListener() {
```

```
                        public void actionPerformed(ActionEvent ae) {

                                p1.removeAll();
                                frame.invalidate();
                                frame.validate();
                                frame.repaint();
                                JPanel p=new JPanel();
                                txtclient_id=new JTextField(20);
                                txtclient_name=new JTextField(20);
                                txtclient_pass=new JTextField(20);
                                txtmsg=new JTextArea(8,50);
                                txtmsg.setEditable(false);
                                btn=new JButton();
                                List idlist =new List(10);
                                try
                                {
                                 ResultSet rs = statement.executeQuery("SELECT client_id
FROM client");
                                  while (rs.next())
                                  {
                                          idlist.add(rs.getString("client_id"));
                                  }
                                }
                                catch (SQLException e)
                                {
                                  txtmsg.append(e.getMessage());
                                }
                                p1.add(idlist);
                                p.add(lblclient_id);
                                p.add(txtclient_id);
                                p.add(lblclient_name);
                                p.add(txtclient_name);
                                p.add(lblclient_pass);
                                p.add(txtclient_pass);
                                p.setLayout(new GridLayout(5,2));
                                p1.add(p);
                                p1.setBackground(Color.yellow);
                                p1.add(btn);
                                p1.add(txtmsg);
                                btn.setText("MODIFY");
                                p1.setLayout(new FlowLayout());

                                frame.add(p1,BorderLayout.CENTER);
                                frame.validate();

                                idlist.addItemListener(new ItemListener()
                                {
                                        @Override
                                        public void itemStateChanged(ItemEvent arg0) {
                                                // TODO Auto-generated method stub
```

```
                                                  try
                                                  {
                                                    ResultSet rs = 
statement.executeQuery("SELECT * FROM client where CLIENT_ID 
="+idlist.getSelectedItem());
                                                          rs.next();

        txtclient_id.setText(rs.getString("CLIENT_ID"));

        txtclient_name.setText(rs.getString("CLIENT_NAME"));

        txtclient_pass.setText(rs.getString("CLIENT_PASS"));
                                                  }
                                                  catch (SQLException selectException)
                                                  {

        txtmsg.append(selectException.getMessage());
                                                  }
                                          }
                                  });


                                  btn.addActionListener(new ActionListener() {

                                          @Override
                                          public void actionPerformed(ActionEvent e) {
                                                  // TODO Auto-generated method stub
                                                  try
                                                  {
                                                          Statement statement = 
connection.createStatement();
                                                          txtmsg.append("UPDATE client "
                                                                          + "SET client_name='" + 
txtclient_name.getText() + "', "
                                                                          + "client_pass='" + 
txtclient_pass.getText() + "'"
                                                                     + " WHERE client_id = "
                                                                          + 
idlist.getSelectedItem()+"\n");
                                                          int i = 
statement.executeUpdate("UPDATE client "
                                                          + "SET client_name='" + 
txtclient_name.getText() + "', "
                                                          + "client_pass='" + 
txtclient_pass.getText() + "'"
                                                          + " WHERE client_id = "
                                                          + idlist.getSelectedItem());
                                                          System.out.println("successful");
```

```
                            txtmsg.append("\nUpdated " + i + " rows
successfully");
                        }
                        catch (SQLException insertException)
                        {

    txtmsg.append(insertException.getMessage());
                        }

                    }
                });
            }
        });


        miDelete1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {

                p1.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                JPanel p=new JPanel();
                txtclient_id=new JTextField(20);
                txtclient_name=new JTextField(20);
                txtclient_pass=new JTextField(20);
                txtmsg=new JTextArea(8,50);
                btn=new JButton();
                List idlist =new List(10);
                try
                {
                 ResultSet rs = statement.executeQuery("SELECT client_id
FROM client");
                  while (rs.next())
                  {
                      idlist.add(rs.getString("client_id"));
                  }
                }
                catch (SQLException e)
                {
                  txtmsg.append(e.getMessage());
                }
                p1.add(idlist);
                p.add(lblclient_id);
                p.add(txtclient_id);
                p.add(lblclient_name);
                p.add(txtclient_name);
                p.add(lblclient_pass);
                p.add(txtclient_pass);
                p.setLayout(new GridLayout(5,2));
```

```
                                p1.add(p);
                                p1.setBackground(Color.red);
                                p1.add(btn);
                                p1.add(txtmsg);
                                btn.setText("DELETE");
                                p1.setLayout(new FlowLayout());

                                frame.add(p1,BorderLayout.CENTER);
                                frame.validate();

                                idlist.addItemListener(new ItemListener()
                                {
                                        @Override
                                        public void itemStateChanged(ItemEvent arg0) {
                                                // TODO Auto-generated method stub

                                                try
                                                {
                                                  ResultSet rs =
statement.executeQuery("SELECT * FROM client where CLIENT_ID
="+idlist.getSelectedItem());
                                                        rs.next();

        txtclient_id.setText(rs.getString("CLIENT_ID"));

        txtclient_name.setText(rs.getString("CLIENT_NAME"));

        txtclient_pass.setText(rs.getString("CLIENT_PASS"));
                                                }
                                                catch (SQLException selectException)
                                                {

        txtmsg.append(selectException.getMessage());
                                                }
                                        }
                                });


                                btn.addActionListener(new ActionListener() {

                                        @Override
                                        public void actionPerformed(ActionEvent e) {
                                                // TODO Auto-generated method stub
                                                try
                                                {
                                                        Statement statement =
connection.createStatement();
                                                        int i = statement.executeUpdate("delete
from client where client_id=" + idlist.getSelectedItem());
```

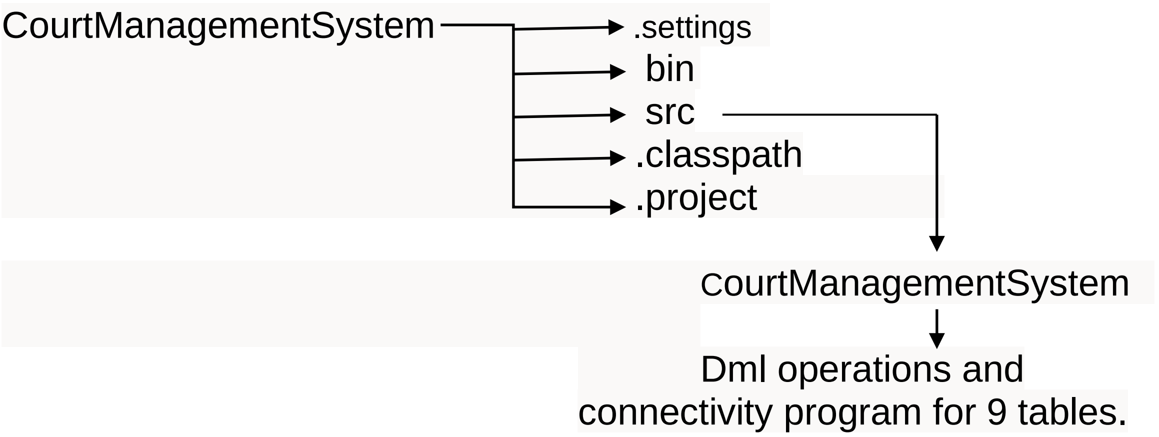
```
                                                        txtmsg.append("\nDeleted " + i + " rows
successfully");
                                                        txtclient_id.setText(null);
                                                        txtclient_name.setText(null);
                                                        txtclient_pass.setText(null);
                                                        idlist.removeAll();
                                                }
                                                catch (SQLException insertException)
                                                {

        txtmsg.append(insertException.getMessage());
                                                }

                                        }
                                });
                        }
                });

                miView1.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent e) {
                                // TODO Auto-generated method stub
                                p1.removeAll();
                                frame.invalidate();
                                frame.validate();
                                frame.repaint();
                                JTable  j;
                                DefaultTableModel  model  =  new  DefaultTableModel();
                                j  =  new  JTable(model);
                                model.addColumn("Client ID");
                                model.addColumn("Client  Name");
                                model.addColumn("Client Pass");
                                try {
                                        ResultSet rs=statement.executeQuery("select *from
client");

                                while(rs.next()) {
                                model.addRow(new
Object[]{rs.getString("client_id"),rs.getString("client_name"),rs.getString("client_pass")});
                                }
                                }
                                catch(SQLException e1) {
JOptionPane.showMessageDialog(frame,"Something Went Wrong");

                                }
                                JScrollPane  sp  =  new  JScrollPane(j);

                                p1.add(sp,BorderLayout.NORTH);
                                p1.setBackground(Color.orange);
```

```
                                        frame.invalidate();
                                        frame.validate();
                                        frame.repaint();

                              }
                    });


          }
}
```

GitHub links and folder structure

https://github.com/KrishnaChaitanya28/Court-Management-System-.git

Folder Structure:
CourtManagementSystem.zip
Court Management System.pdf(assignment 1)
Dbms_report(1602-19-737-308).pdf



CourtManagementSystem

.settings

bin

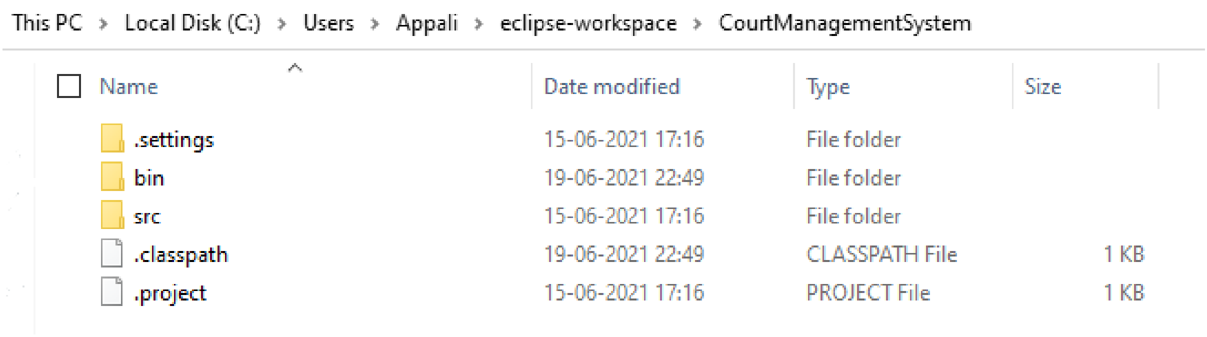
src

.classpath

.project

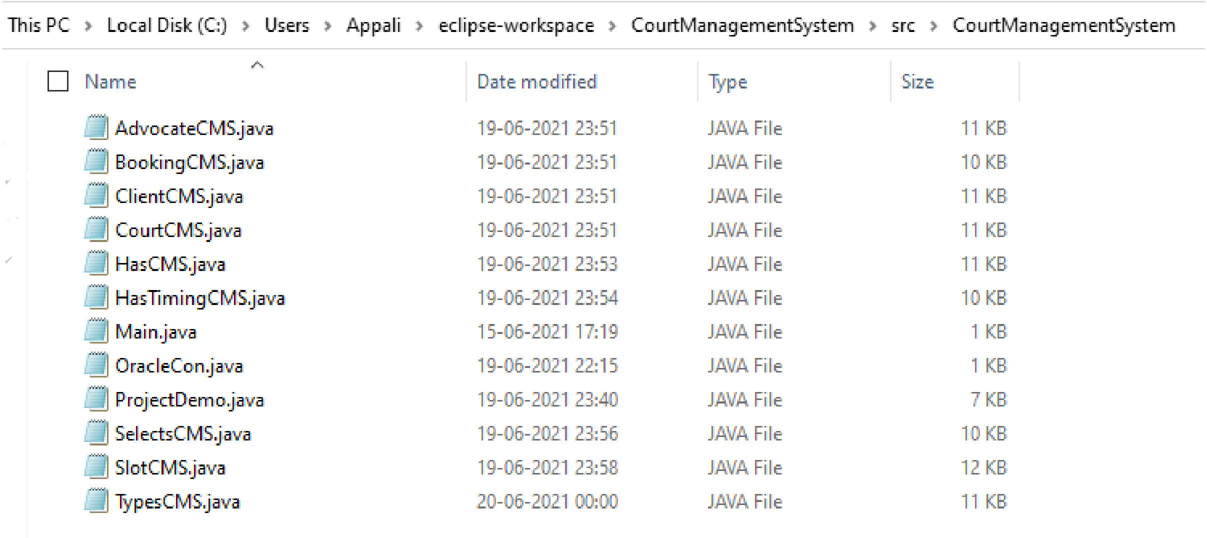
CourtManagementSystem

Dml operations and connectivity program for 9 tables.



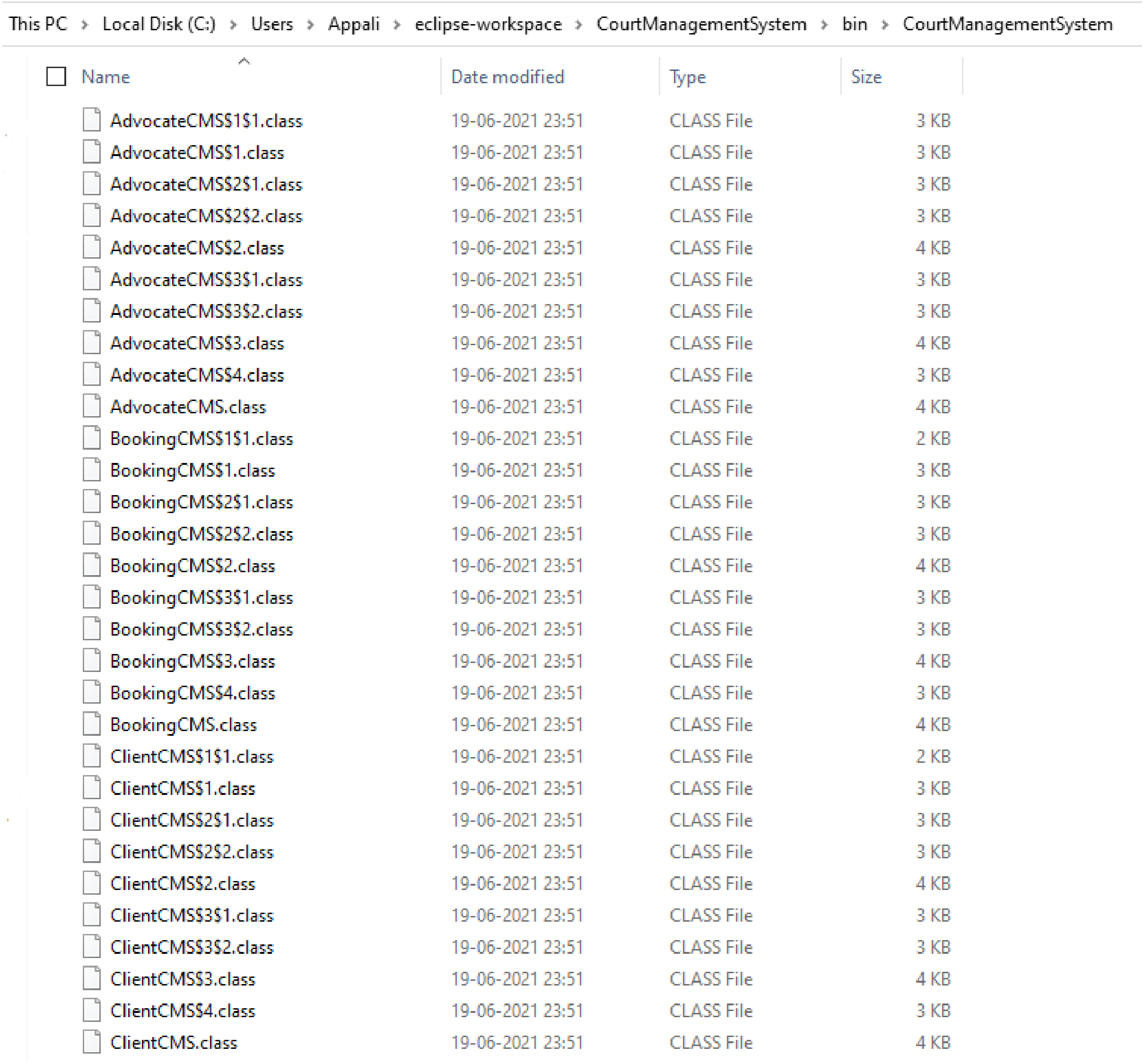
This PC › Local Disk (C:) › Users › Appali › eclipse-workspace › CourtManagementSystem

Name	Date modified	Type	Size
.settings	15-06-2021 17:16	File folder	
bin	19-06-2021 22:49	File folder	
src	15-06-2021 17:16	File folder	
.classpath	19-06-2021 22:49	CLASSPATH File	1 KB
.project	15-06-2021 17:16	PROJECT File	1 KB



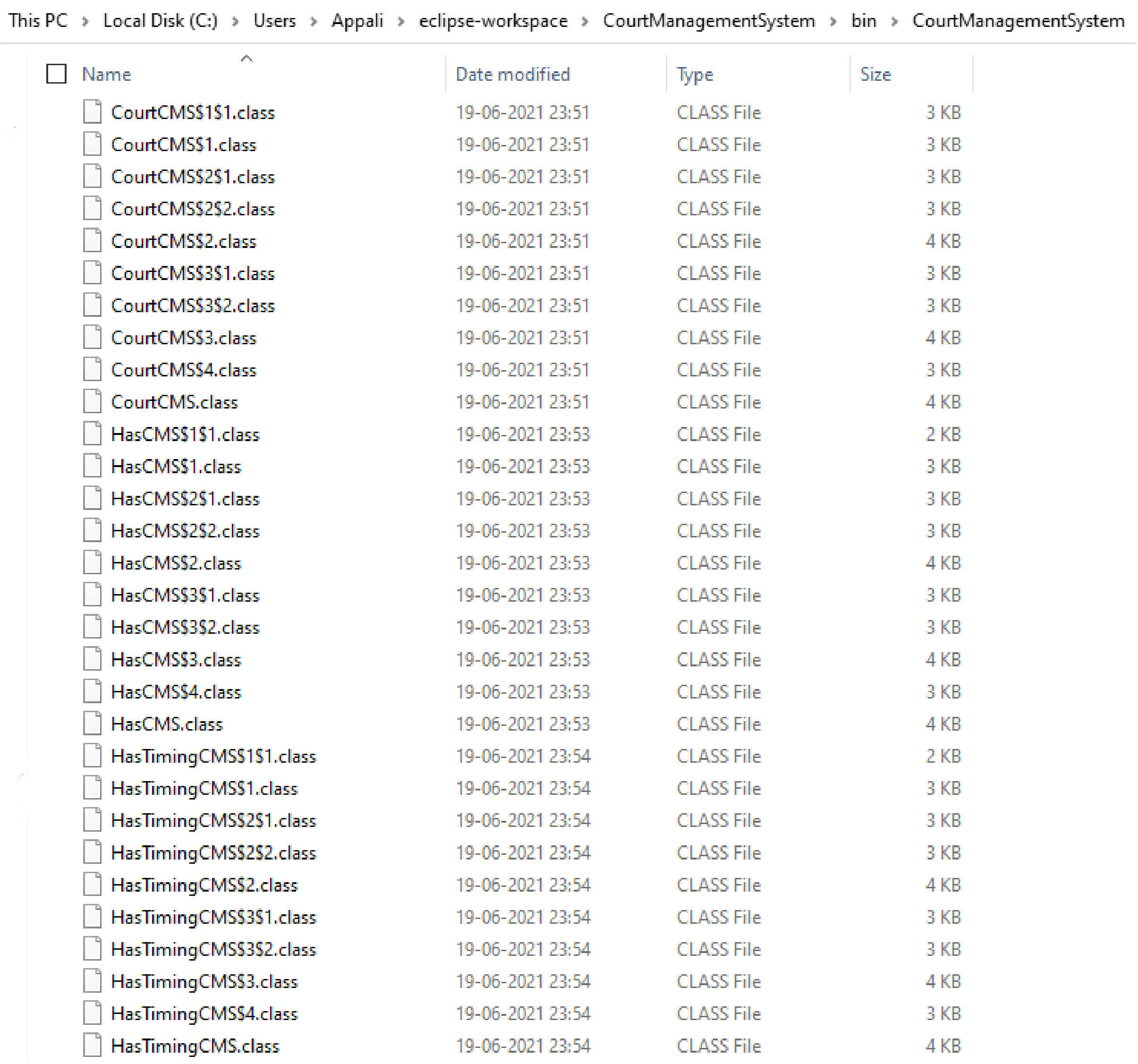
This PC › Local Disk (C:) › Users › Appali › eclipse-workspace › CourtManagementSystem › src › CourtManagementSystem

Name	Date modified	Type	Size
AdvocateCMS.java	19-06-2021 23:51	JAVA File	11 KB
BookingCMS.java	19-06-2021 23:51	JAVA File	10 KB
ClientCMS.java	19-06-2021 23:51	JAVA File	11 KB
CourtCMS.java	19-06-2021 23:51	JAVA File	11 KB
HasCMS.java	19-06-2021 23:53	JAVA File	11 KB
HasTimingCMS.java	19-06-2021 23:54	JAVA File	10 KB
Main.java	15-06-2021 17:19	JAVA File	1 KB
OracleCon.java	19-06-2021 22:15	JAVA File	1 KB
ProjectDemo.java	19-06-2021 23:40	JAVA File	7 KB
SelectsCMS.java	19-06-2021 23:56	JAVA File	10 KB
SlotCMS.java	19-06-2021 23:58	JAVA File	12 KB
TypesCMS.java	20-06-2021 00:00	JAVA File	11 KB



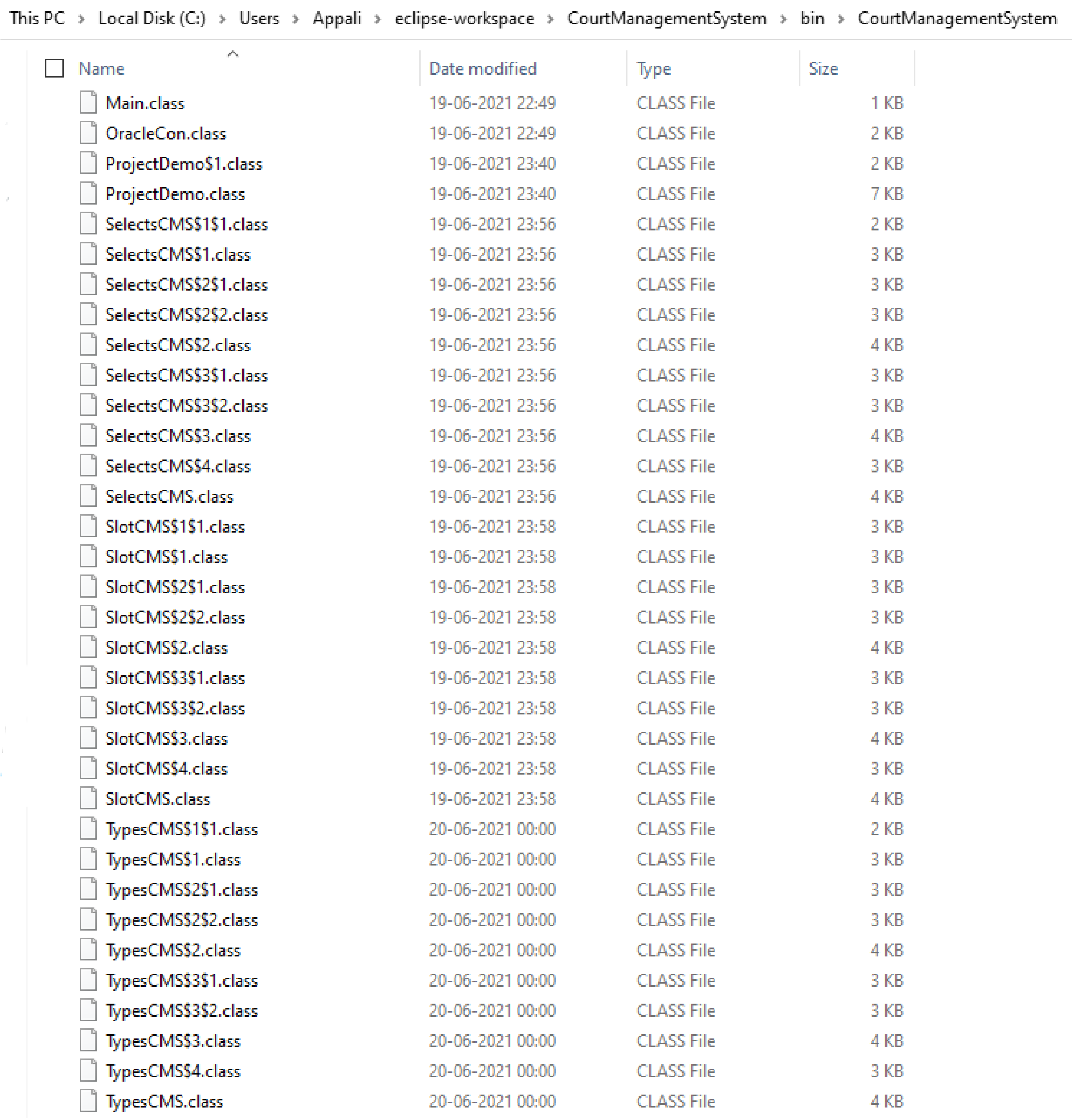
This PC › Local Disk (C:) › Users › Appali › eclipse-workspace › CourtManagementSystem › bin › CourtManagementSystem

□ Name	Date modified	Type	Size
AdvocateCMS$1$1.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS$1.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS$2$1.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS$2$2.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS$2.class	19-06-2021 23:51	CLASS File	4 KB
AdvocateCMS$3$1.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS$3$2.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS$3.class	19-06-2021 23:51	CLASS File	4 KB
AdvocateCMS$4.class	19-06-2021 23:51	CLASS File	3 KB
AdvocateCMS.class	19-06-2021 23:51	CLASS File	4 KB
BookingCMS$1$1.class	19-06-2021 23:51	CLASS File	2 KB
BookingCMS$1.class	19-06-2021 23:51	CLASS File	3 KB
BookingCMS$2$1.class	19-06-2021 23:51	CLASS File	3 KB
BookingCMS$2$2.class	19-06-2021 23:51	CLASS File	3 KB
BookingCMS$2.class	19-06-2021 23:51	CLASS File	4 KB
BookingCMS$3$1.class	19-06-2021 23:51	CLASS File	3 KB
BookingCMS$3$2.class	19-06-2021 23:51	CLASS File	3 KB
BookingCMS$3.class	19-06-2021 23:51	CLASS File	4 KB
BookingCMS$4.class	19-06-2021 23:51	CLASS File	3 KB
BookingCMS.class	19-06-2021 23:51	CLASS File	4 KB
ClientCMS$1$1.class	19-06-2021 23:51	CLASS File	2 KB
ClientCMS$1.class	19-06-2021 23:51	CLASS File	3 KB
ClientCMS$2$1.class	19-06-2021 23:51	CLASS File	3 KB
ClientCMS$2$2.class	19-06-2021 23:51	CLASS File	3 KB
ClientCMS$2.class	19-06-2021 23:51	CLASS File	4 KB
ClientCMS$3$1.class	19-06-2021 23:51	CLASS File	3 KB
ClientCMS$3$2.class	19-06-2021 23:51	CLASS File	3 KB
ClientCMS$3.class	19-06-2021 23:51	CLASS File	4 KB
ClientCMS$4.class	19-06-2021 23:51	CLASS File	3 KB
ClientCMS.class	19-06-2021 23:51	CLASS File	4 KB



This PC > Local Disk (C:) > Users > Appali > eclipse-workspace > CourtManagementSystem > bin > CourtManagementSystem

□ Name	Date modified	Type	Size
CourtCMS$1$1.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS$1.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS$2$1.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS$2$2.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS$2.class	19-06-2021 23:51	CLASS File	4 KB
CourtCMS$3$1.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS$3$2.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS$3.class	19-06-2021 23:51	CLASS File	4 KB
CourtCMS$4.class	19-06-2021 23:51	CLASS File	3 KB
CourtCMS.class	19-06-2021 23:51	CLASS File	4 KB
HasCMS$1$1.class	19-06-2021 23:53	CLASS File	2 KB
HasCMS$1.class	19-06-2021 23:53	CLASS File	3 KB
HasCMS$2$1.class	19-06-2021 23:53	CLASS File	3 KB
HasCMS$2$2.class	19-06-2021 23:53	CLASS File	3 KB
HasCMS$2.class	19-06-2021 23:53	CLASS File	4 KB
HasCMS$3$1.class	19-06-2021 23:53	CLASS File	3 KB
HasCMS$3$2.class	19-06-2021 23:53	CLASS File	3 KB
HasCMS$3.class	19-06-2021 23:53	CLASS File	4 KB
HasCMS$4.class	19-06-2021 23:53	CLASS File	3 KB
HasCMS.class	19-06-2021 23:53	CLASS File	4 KB
HasTimingCMS$1$1.class	19-06-2021 23:54	CLASS File	2 KB
HasTimingCMS$1.class	19-06-2021 23:54	CLASS File	3 KB
HasTimingCMS$2$1.class	19-06-2021 23:54	CLASS File	3 KB
HasTimingCMS$2$2.class	19-06-2021 23:54	CLASS File	3 KB
HasTimingCMS$2.class	19-06-2021 23:54	CLASS File	4 KB
HasTimingCMS$3$1.class	19-06-2021 23:54	CLASS File	3 KB
HasTimingCMS$3$2.class	19-06-2021 23:54	CLASS File	3 KB
HasTimingCMS$3.class	19-06-2021 23:54	CLASS File	4 KB
HasTimingCMS$4.class	19-06-2021 23:54	CLASS File	3 KB
HasTimingCMS.class	19-06-2021 23:54	CLASS File	4 KB



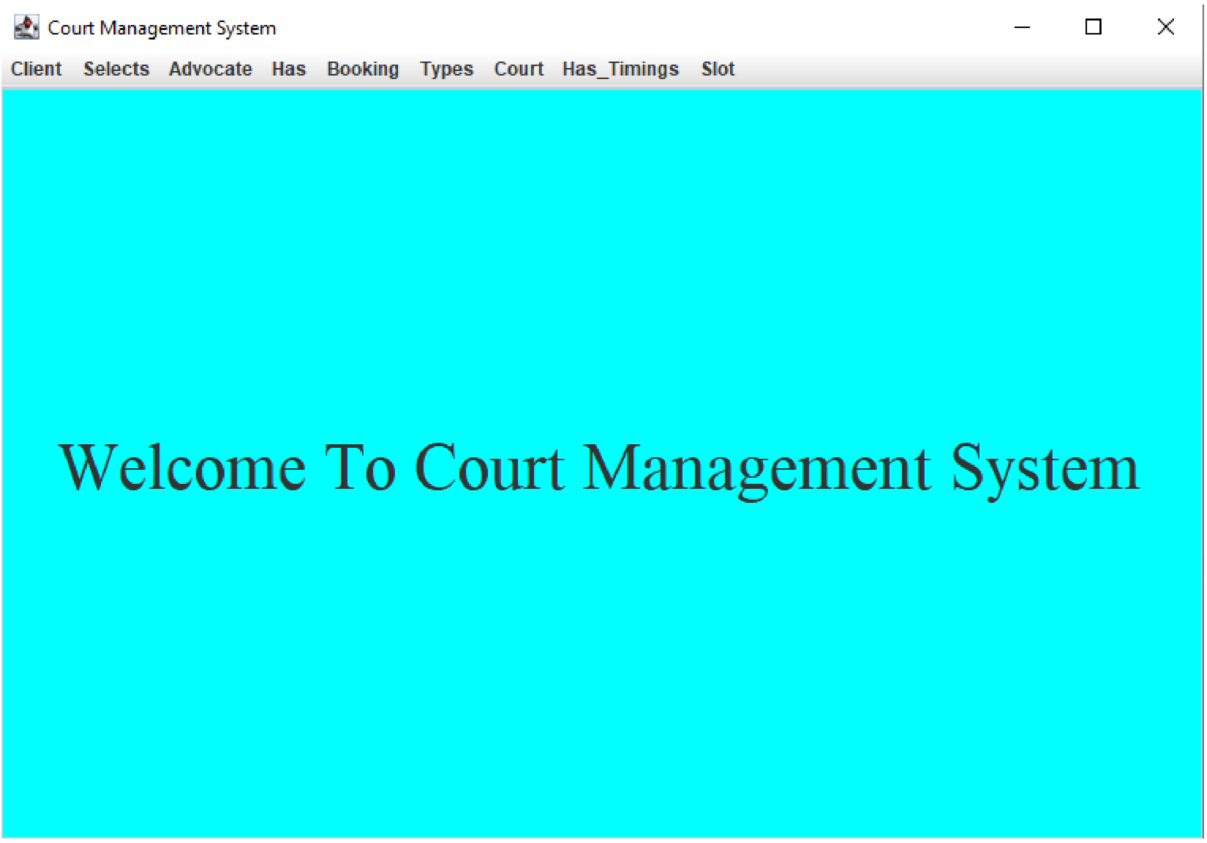
This PC › Local Disk (C:) › Users › Appali › eclipse-workspace › CourtManagementSystem › bin › CourtManagementSystem

□ Name	Date modified	Type	Size
Main.class	19-06-2021 22:49	CLASS File	1 KB
OracleCon.class	19-06-2021 22:49	CLASS File	2 KB
ProjectDemo$1.class	19-06-2021 23:40	CLASS File	2 KB
ProjectDemo.class	19-06-2021 23:40	CLASS File	7 KB
SelectsCMS$1$1.class	19-06-2021 23:56	CLASS File	2 KB
SelectsCMS$1.class	19-06-2021 23:56	CLASS File	3 KB
SelectsCMS$2$1.class	19-06-2021 23:56	CLASS File	3 KB
SelectsCMS$2$2.class	19-06-2021 23:56	CLASS File	3 KB
SelectsCMS$2.class	19-06-2021 23:56	CLASS File	4 KB
SelectsCMS$3$1.class	19-06-2021 23:56	CLASS File	3 KB
SelectsCMS$3$2.class	19-06-2021 23:56	CLASS File	3 KB
SelectsCMS$3.class	19-06-2021 23:56	CLASS File	4 KB
SelectsCMS$4.class	19-06-2021 23:56	CLASS File	3 KB
SelectsCMS.class	19-06-2021 23:56	CLASS File	4 KB
SlotCMS$1$1.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS$1.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS$2$1.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS$2$2.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS$2.class	19-06-2021 23:58	CLASS File	4 KB
SlotCMS$3$1.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS$3$2.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS$3.class	19-06-2021 23:58	CLASS File	4 KB
SlotCMS$4.class	19-06-2021 23:58	CLASS File	3 KB
SlotCMS.class	19-06-2021 23:58	CLASS File	4 KB
TypesCMS$1$1.class	20-06-2021 00:00	CLASS File	2 KB
TypesCMS$1.class	20-06-2021 00:00	CLASS File	3 KB
TypesCMS$2$1.class	20-06-2021 00:00	CLASS File	3 KB
TypesCMS$2$2.class	20-06-2021 00:00	CLASS File	3 KB
TypesCMS$2.class	20-06-2021 00:00	CLASS File	4 KB
TypesCMS$3$1.class	20-06-2021 00:00	CLASS File	3 KB
TypesCMS$3$2.class	20-06-2021 00:00	CLASS File	3 KB
TypesCMS$3.class	20-06-2021 00:00	CLASS File	4 KB
TypesCMS$4.class	20-06-2021 00:00	CLASS File	3 KB
TypesCMS.class	20-06-2021 00:00	CLASS File	4 KB

Testing:

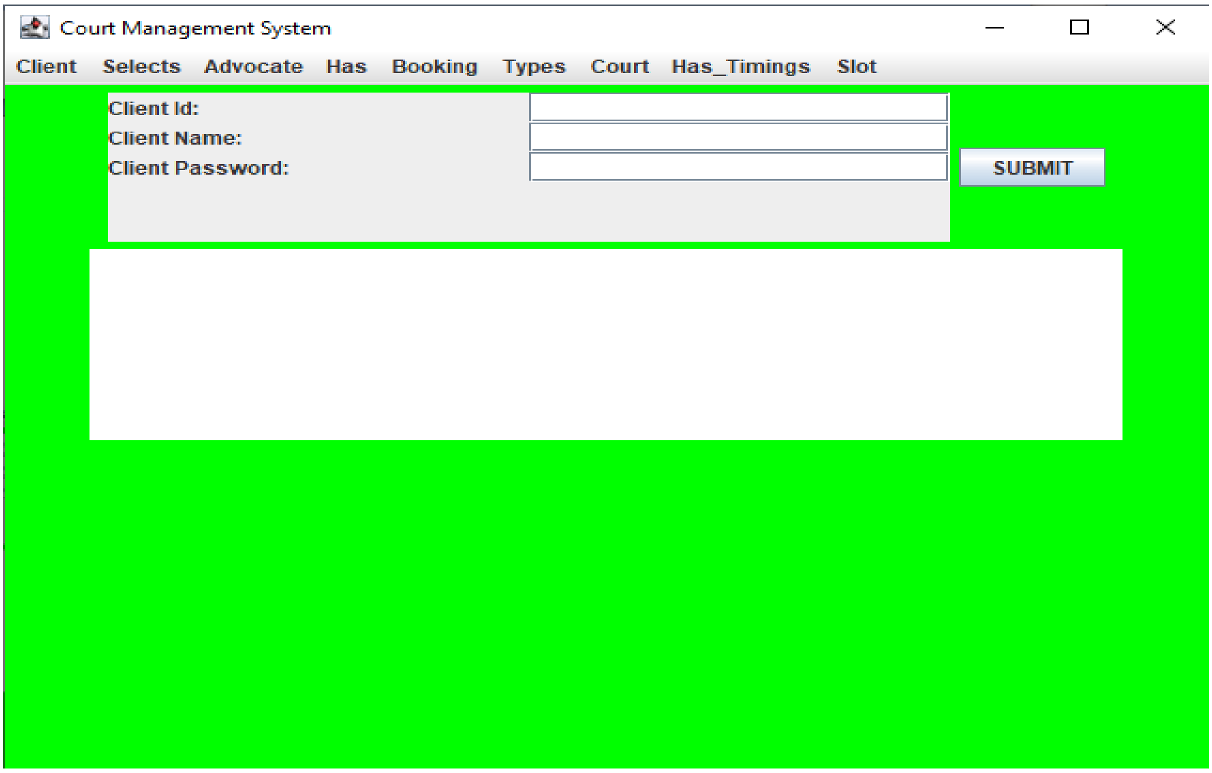
Java GUI Testing:

User Interface

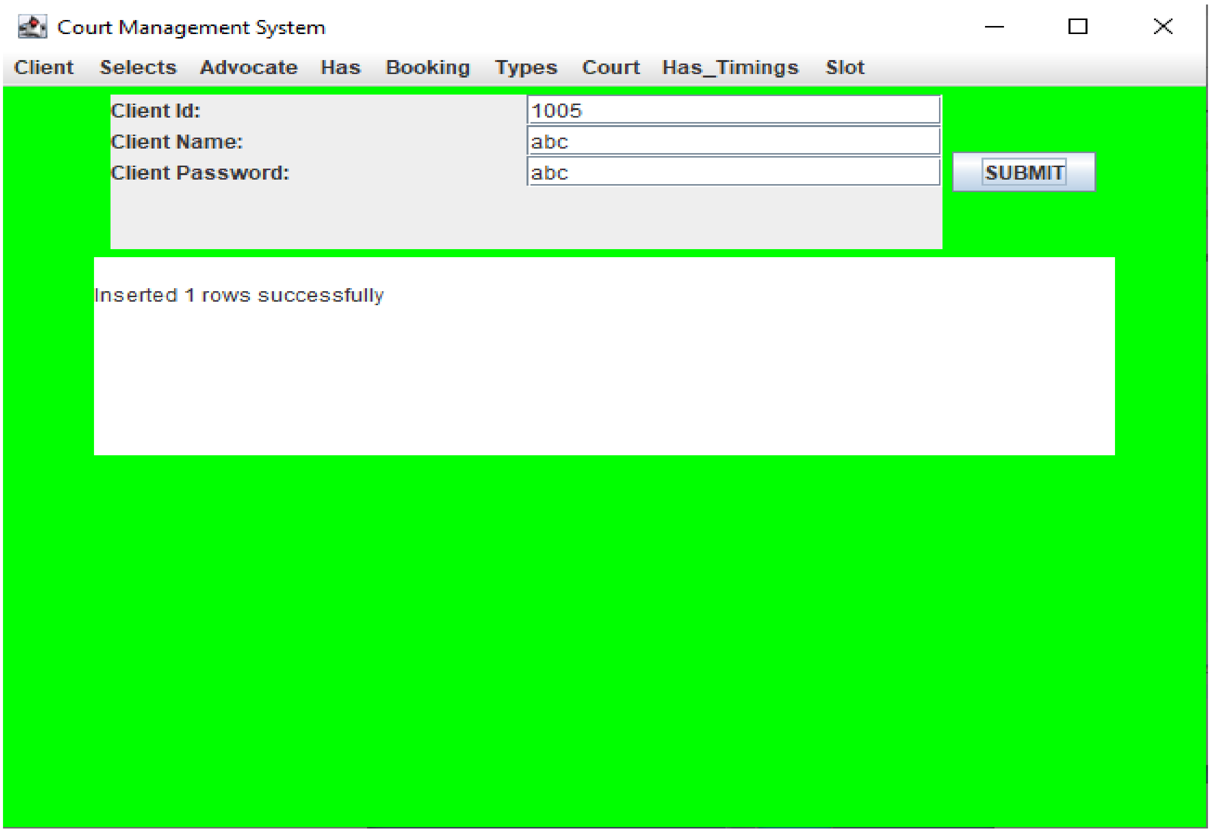


Welcome To Court Management System

Insert Operation(Submit)



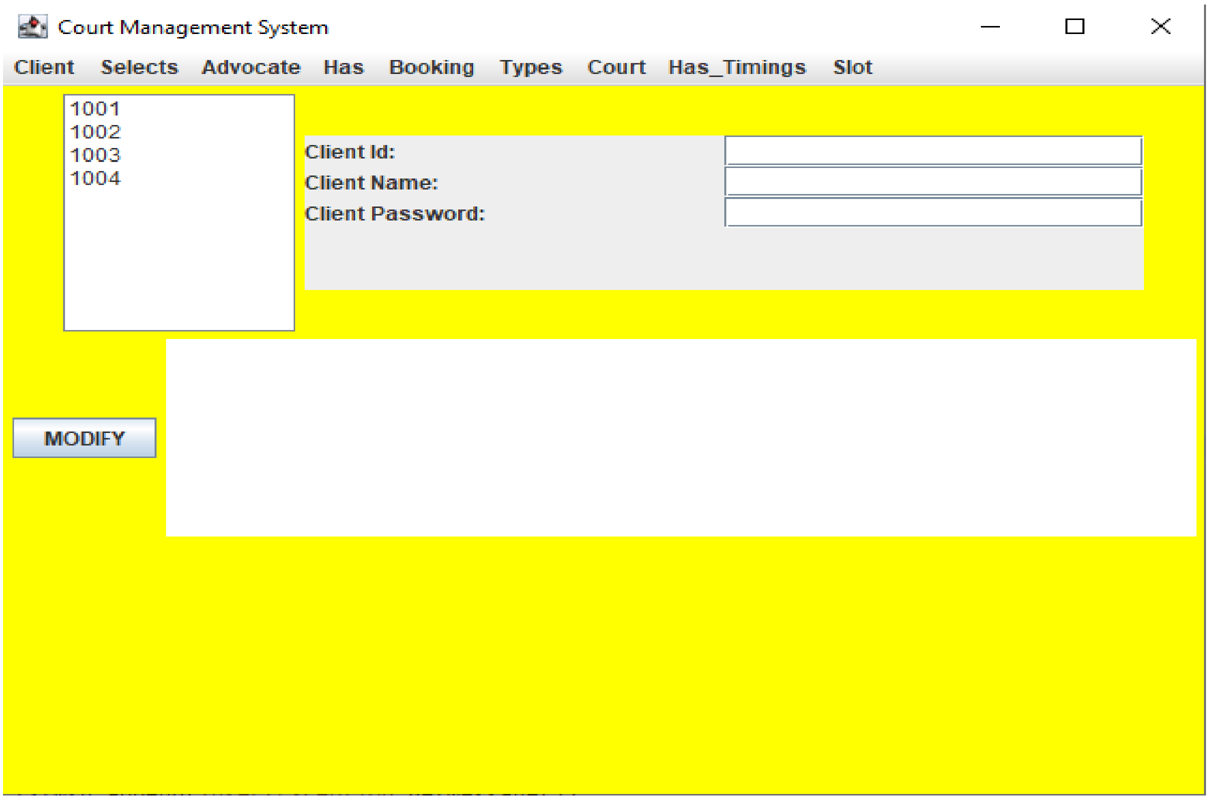
Client Id:
Client Name:
Client Password:



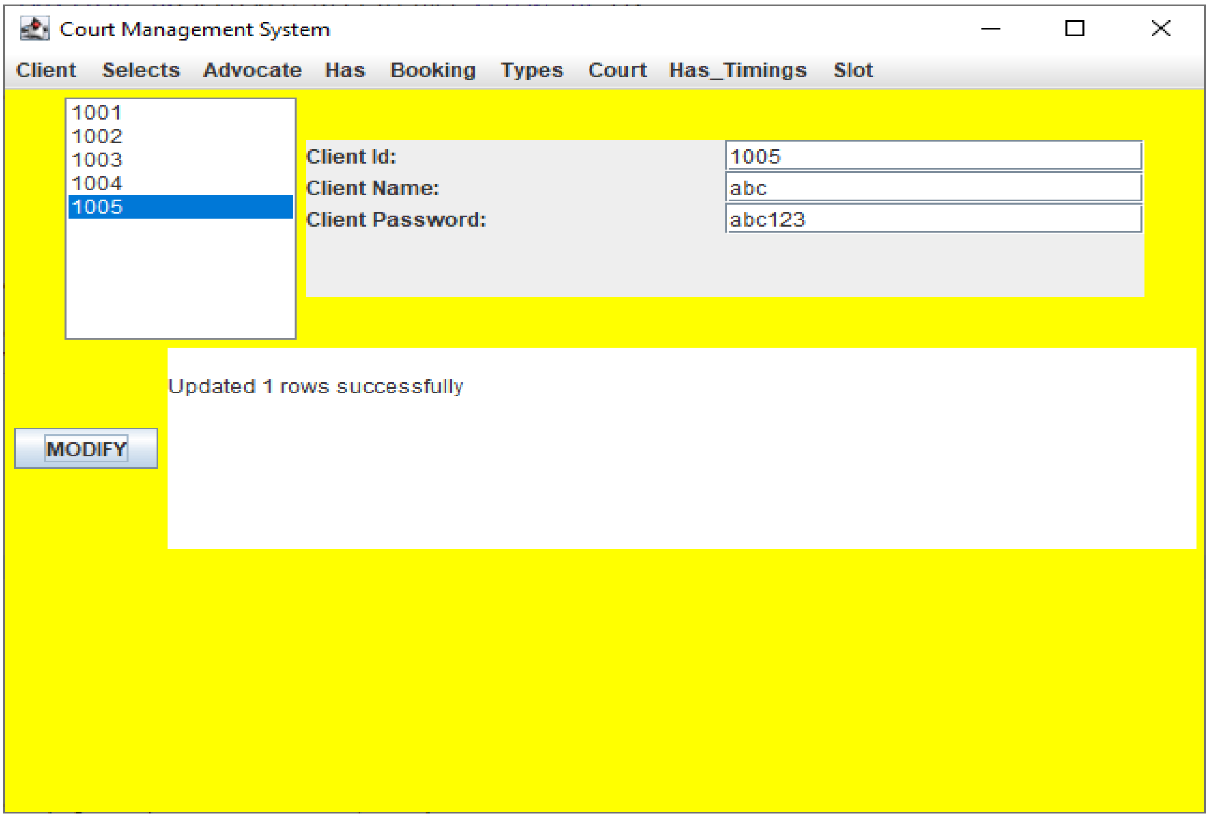
Client Id: 1005
Client Name: abc
Client Password: abc

```
Inserted 1 rows successfully
```

Update Operation(Modify)



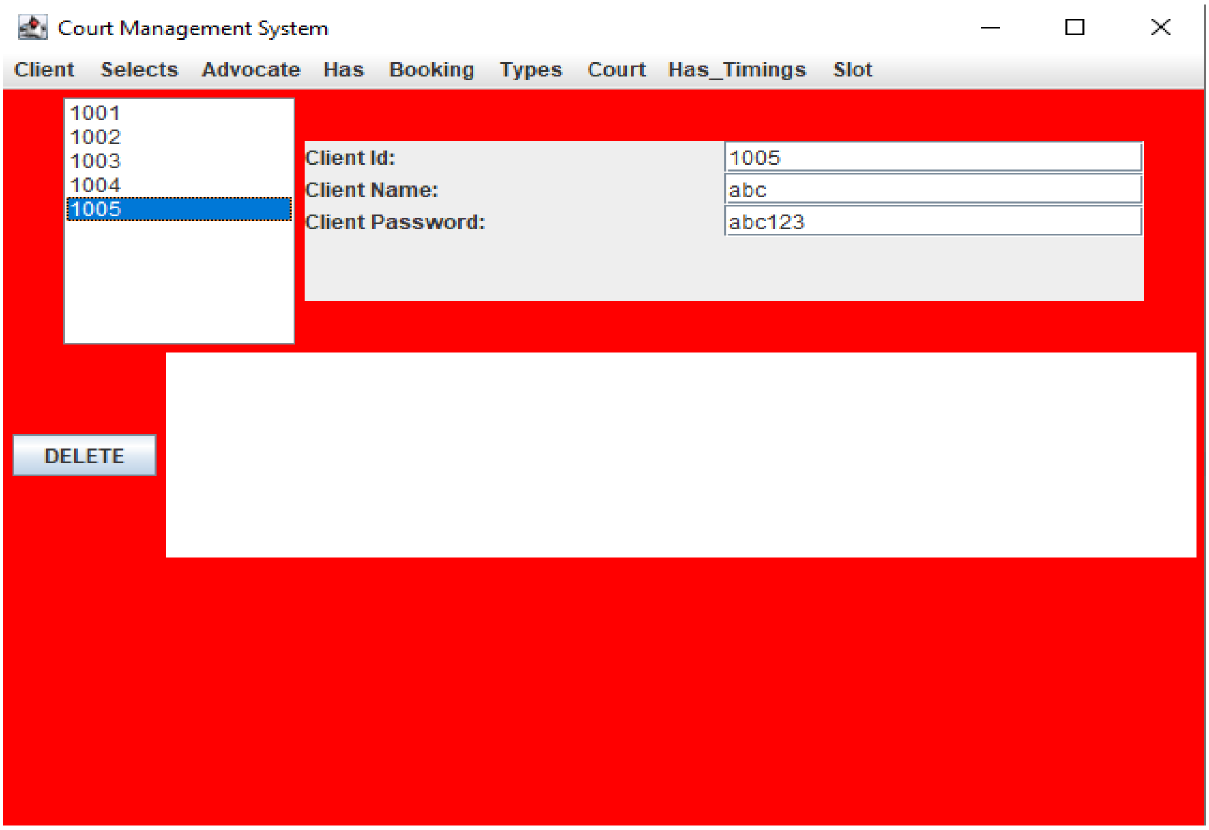
Client Id:
Client Name:
Client Password:



Client Id: 1005
Client Name: abc
Client Password: abc123

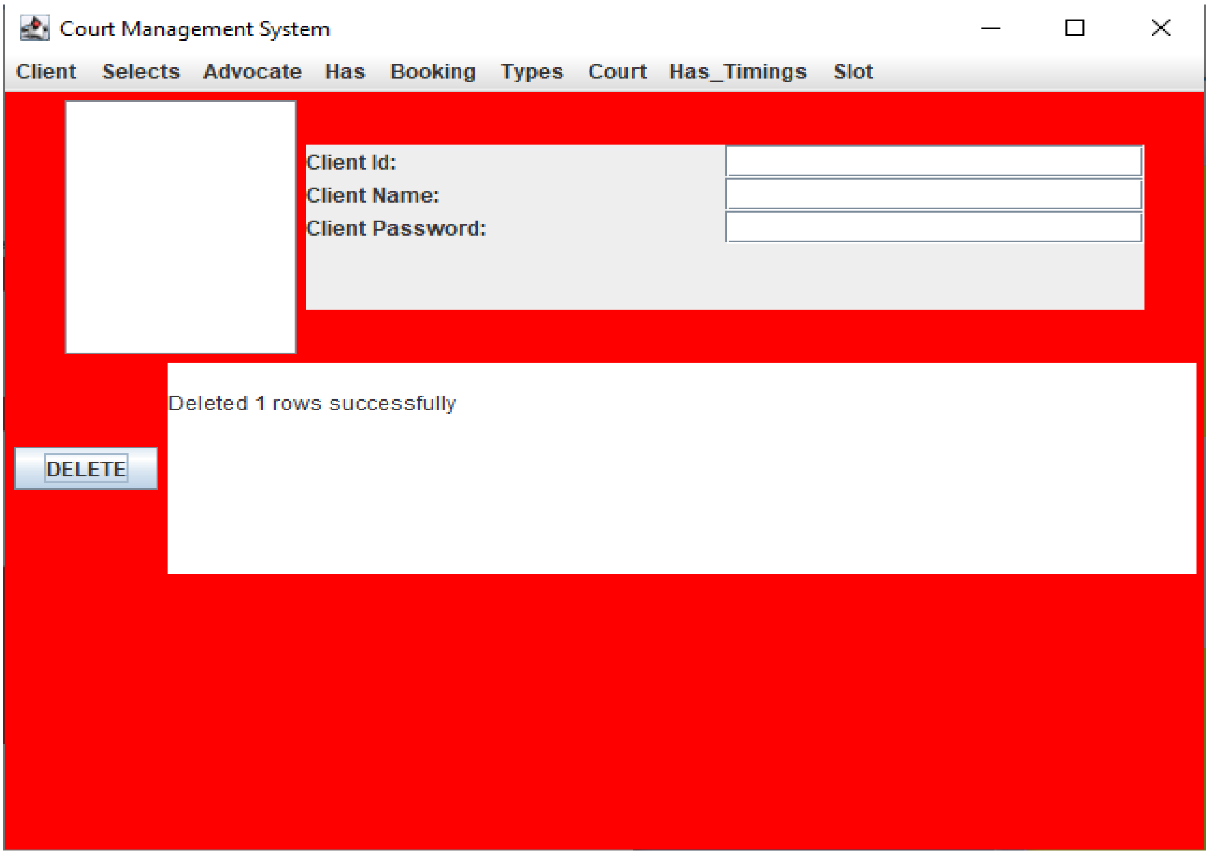
Updated 1 rows successfully

Delete Operation


Court Management System
Client Selects Advocate Has Booking Types Court Has_Timings Slot
DELETE

1001
1002
1003
1004
1005

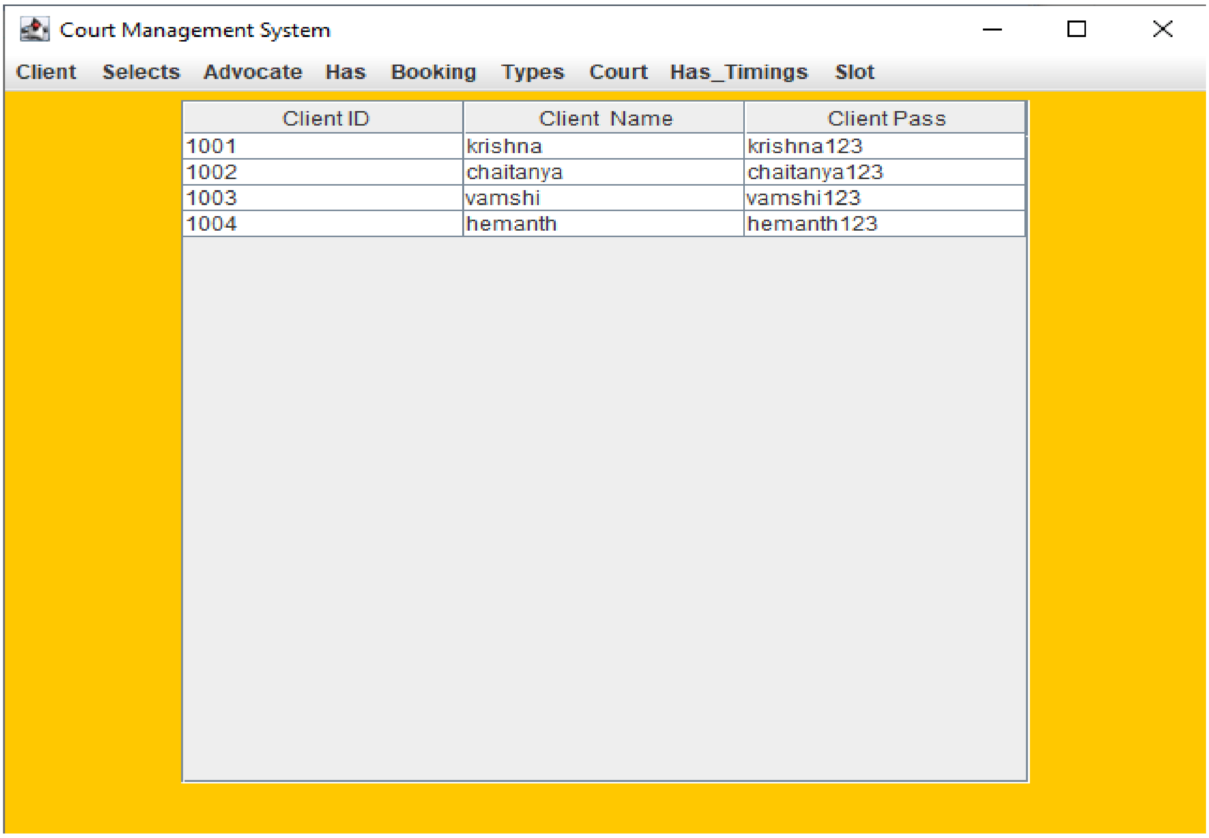
Client Id: 1005
Client Name: abc
Client Password: abc123


Court Management System
Client Selects Advocate Has Booking Types Court Has_Timings Slot
DELETE

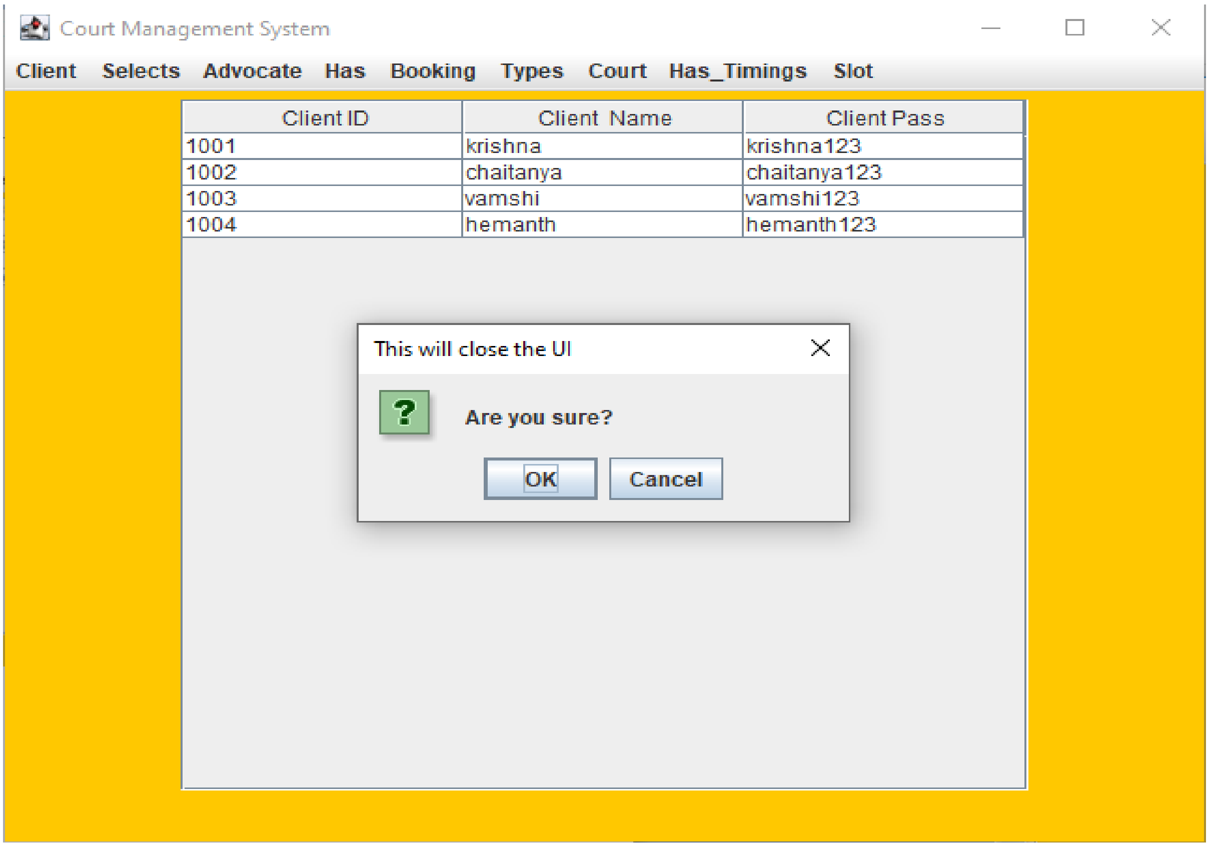
Client Id:
Client Name:
Client Password:

```
Deleted 1 rows successfully
```

Display Operation(View)


Court Management System
Client Selects Advocate Has Booking Types Court Has_Timings Slot

Client ID	Client Name	Client Pass
1001	krishna	krishna123
1002	chaitanya	chaitanya123
1003	vamshi	vamshi123
1004	hemanth	hemanth123

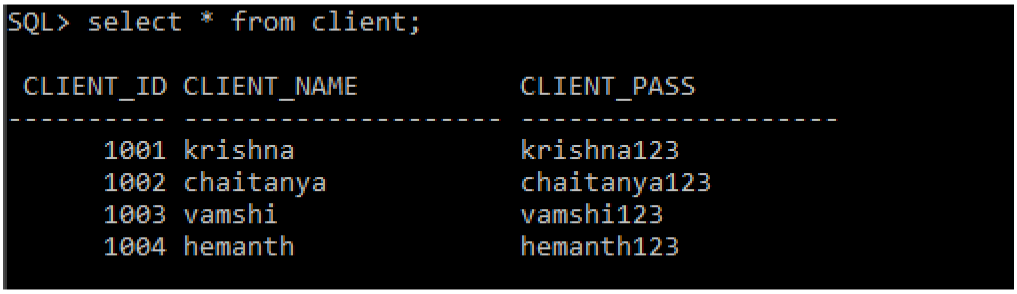

Court Management System
Client Selects Advocate Has Booking Types Court Has_Timings Slot

Client ID	Client Name	Client Pass
1001	krishna	krishna123
1002	chaitanya	chaitanya123
1003	vamshi	vamshi123
1004	hemanth	hemanth123

This will close the UI

Are you sure?

The data entered in the above form is updated in the table of the Oracle database 11g as:



```
SQL> select * from client;

 CLIENT_ID CLIENT_NAME          CLIENT_PASS
---------- -------------------- --------------------
      1001 krishna              krishna123
      1002 chaitanya            chaitanya123
      1003 vamshi               vamshi123
      1004 hemanth              hemanth123
```

Results:

I successfully completed this Assignment “Court Management System”.

Discussion and Future work

While doing this project I got to learn many new things (both front end and back end). This project helps many people around the country to complete most (80%) of work from home and also has feedback and experience column which helps users/customers/clients to select a suitable/experienced advocate.

In future I want to create a android/Ios app which will reduce most of the physical work and also there could be room for including advanced software and other technologies that could make the project more purposeful and better for future use.

References:

- https://docs.oracle.com/javase/7/docs/index.html
- https://www.javatpoint.com/dbms-tutorial
- http://www.sqlines.com/articles/java/sql_server_jdbc_connection

Thank You