**JAVA SWING BASED- SOCCER LEAGUE DATABSE SYSTEM-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

**R.SRUJAN TEJA <1602-20-737-046>**

**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

**“SOCCER LEAGUE DATABASE MANAGEMENT SYSTEM”** project work of R.SRUJAN TEJA bearing Roll.no:1602-20-737-046 who carried out this project under my supervision in the IV semester for the academic year 2021-2022.

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

**ABSTRACT:**

In this current scenario, there are many league matches happening in world. Soccer League database system utilizes league database that allow analyst to incorporate functions that are used by league application to fetch and view information. It gives the accurate information about the league matches. As we know soccer league gains more popularity, it attracts more endorsement for each team. The objective of this application is to develop a system where every information about matches are stored and maintained with clear information. In this project we build a Logistic model by which it has been shown that for a particular team, player roles, the goals scored by the team in a match and no of winnings per team and their position in the table in the league. The teams would be placed in the table according to the most number of wins in the match. It is a kind of soccer league manager type of application. From the user can manage the leagues and its fixtures. Here one can add different teams and different game times. Also, the user can simply add a number of players he wants. Later he can update each individuals profile.

**Introduction: REQUIREMENT ANALYSIS**

List of tables: 1.Team owner

2.Team

3.Player

4.rolls

5.Match

6.Venue

7.performance

List of attributes with their domain types:

Team owner

* ownerid number(5)
* ownername varchar2(15)

Team

* Teamid number(5)
* Teamname varchar2(15)
* ownerid number(3)
* noplayers number(2)

player

* playerid number(5)
* pname varchar2(15)
* pexperience number(2)
* teamid number(5)

Rolls

* playerid number(5)
* rollplayed varchar2(10)

Match

* matchid number(5)
* teamid number(5)

Venue

* venueid number(5)
* vname varchar2(10)
* vaddress varchar2(10)

Performance

* pid number(5)
* matchid number(5)
* goalsavg number(5)

**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:1522:ORCL","mydbms","mydbms"); 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:**

1.create table owneripl (ownerid number(5) primary key,oname varchar(15));

Table created.

2.create table team(teamid number(5) primary key,tname varchar(10),ownerid number(5) references owneripl(ownerid),noplayers number(2));

Table created.

3.create table player(playerid number(5) primary key,pname varchar(15),pnationality varchar(10),pexperience number(2),teamid number(5) references team(teamid));

Table created.

4.create table rolls(playerid number(5) references player(playerid),rollplayed varchar(10));

Table created.

5.create table match(matchid number(5) primary key,teamid number(5) references team(teamid));

Table created.

6.create table venue(venueid number(5) primary key,vname varchar(10), vaddress varchar(10));

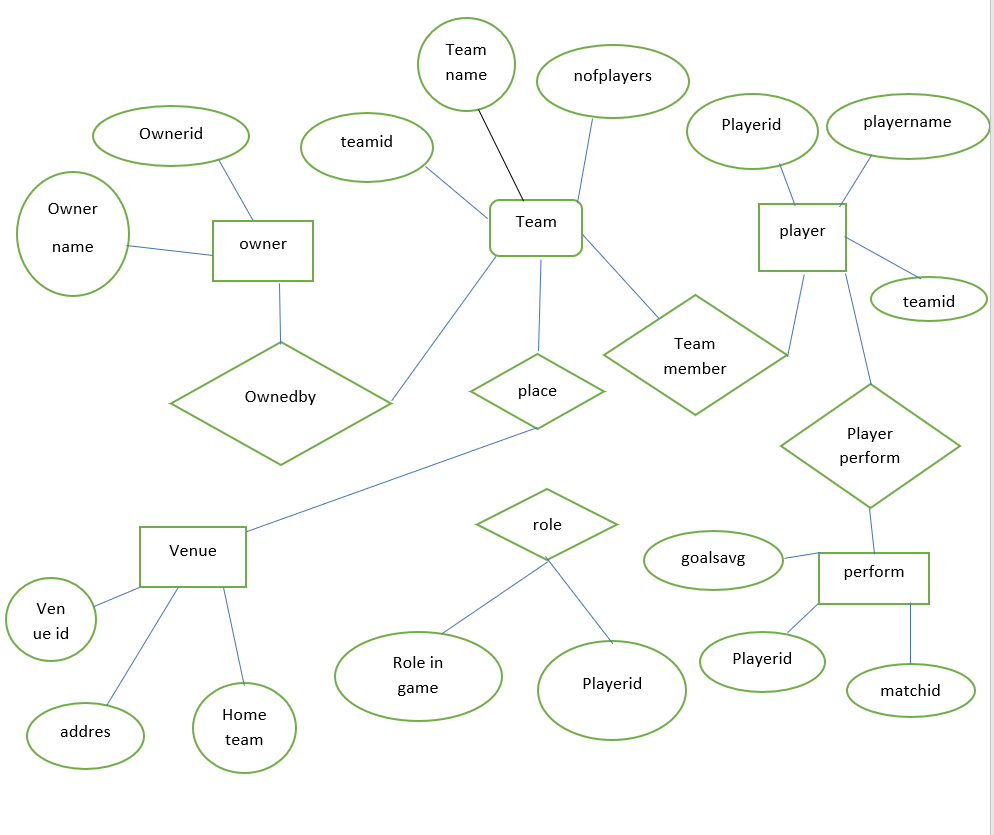
Table created.

7.create table performance(pid number(5) references player(playerid),matchid number(5) references match(goalsavg number(5));

Table created.

**`DESIGN:**

**ER DIAGRAM**



**Implementation:**

**Program:**

**Data base connection:**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class DatabaseConnection {

Connection connection = null;

private String userName = "root";

private String password = "";

String DATABASE\_URL = "jdbc:mysql://localhost/footballmanagement2";

// public DatabaseConnection(String DATABASE\_URL){

// this.DATABASE\_URL = DATABASE\_URL;

// }

public Connection startConnection(){

try {

// String DATABASE\_URL = "jdbc:mysql://localhost:3306/mynewdatabase";

Class.forName("com.mysql.jdbc.Driver").newInstance();

} catch (ClassNotFoundException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

try {

connection = DriverManager.getConnection(DATABASE\_URL,userName,password);

System.out.println("connected");

} catch (SQLException ex) {

Logger.getLogger(DatabaseConnection.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, ex);

}

return connection;

}

public void closeConnection(){

try {

connection.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseConnection.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

**Fixture**:

import java.util.ArrayList;

import java.util.Collections;

import java.util.Random;

public class Fixture {

public int[] group = new int[12];

public void createFixture(){

ArrayList<Integer> list = new ArrayList<Integer>();

for (int i=5; i<=16; i++) {

list.add(new Integer(i));

}

Collections.shuffle(list);

for (int i=0; i<12; i++) {

group[i]=list.get(i);

}

for(int i=0;i<12;i++){

System.out.println(group[i]);

}

}

}

**SearchBoxModel**:

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.util.ArrayList;

import javax.swing.AbstractListModel;

import javax.swing.ComboBoxEditor;

import javax.swing.ComboBoxModel;

import javax.swing.JComboBox;

import javax.swing.JTextField;

public class SearchBoxModel extends AbstractListModel

implements ComboBoxModel, KeyListener, ItemListener

{

ArrayList<String> db = new ArrayList<String>();

ArrayList<String> data = new ArrayList<String>();

String selection;

JComboBox cb;

ComboBoxEditor cbe;

int currPos = 0;

public SearchBoxModel(JComboBox jcb,ArrayList<String> db )

{

cb = jcb;

cbe = jcb.getEditor();

//here we add the key listener to the text field that the combobox is wrapped around

cbe.getEditorComponent().addKeyListener(this);

//set up our "database" of items - in practice you will usuallu have a proper db.

this.db = db;

}

public void updateModel(String in)

{

data.clear();

//lets find any items which start with the string the user typed, and add it to the popup list

//here you would usually get your items from a database, or some other storage...

for(String s:db)

if(s.startsWith(in))

data.add(s);

super.fireContentsChanged(this, 0, data.size());

//this is a hack to get around redraw problems when changing the list length of the displayed popups

cb.hidePopup();

cb.showPopup();

if(data.size() != 0)

cb.setSelectedIndex(-1);

}

public int getSize(){return data.size();}

public Object getElementAt(int index){return data.get(index);}

public void setSelectedItem(Object anItem)

{selection = (String) anItem;}

public Object getSelectedItem(){return selection;}

public void keyTyped(KeyEvent e){}

public void keyPressed(KeyEvent e){}

public void keyReleased(KeyEvent e)

{

String str = cbe.getItem().toString();

JTextField jtf = (JTextField)cbe.getEditorComponent();

currPos = jtf.getCaretPosition();

if(e.getKeyChar() == KeyEvent.CHAR\_UNDEFINED)

{

if(e.getKeyCode() != KeyEvent.VK\_ENTER )

{

cbe.setItem(str);

jtf.setCaretPosition(currPos);

}

}

else if(e.getKeyCode() == KeyEvent.VK\_ENTER)

cb.setSelectedIndex(cb.getSelectedIndex());

else

{

updateModel(cb.getEditor().getItem().toString());

cbe.setItem(str);

jtf.setCaretPosition(currPos);

}

}

public void itemStateChanged(ItemEvent e)

{

cbe.setItem(e.getItem().toString());

cb.setSelectedItem(e.getItem());

}

}

**Main Frame**:

import com.jtattoo.plaf.hifi.HiFiLookAndFeel;

import java.awt.CardLayout;

import java.awt.Image;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import java.time.LocalDate;

import java.time.ZoneId;

import java.util.ArrayList;

import java.util.Date;

import java.util.Properties;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.SwingUtilities;

import javax.swing.UIManager;

import net.proteanit.sql.DbUtils;

public class MainFrame extends javax.swing.JFrame {

// DatabaseConnection databaseConnection;

/\*\*

\* Creates new form MainFrame

\*/

public MainFrame() {

initComponents();

connection = startConnection();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

LoginPanel = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

userNameLabel = new javax.swing.JLabel();

userNameTextField = new javax.swing.JTextField();

passwordLabel = new javax.swing.JLabel();

signInButton = new javax.swing.JButton();

ExitButton = new javax.swing.JButton();

jPasswordField1 = new javax.swing.JPasswordField();

MainMenu = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

playerButton = new javax.swing.JButton();

teamButton = new javax.swing.JButton();

searchButton = new javax.swing.JButton();

fixtureButton = new javax.swing.JButton();

transferButton = new javax.swing.JButton();

exitButton = new javax.swing.JButton();

fixtureLabel = new javax.swing.JLabel();

SearchLabel = new javax.swing.JLabel();

managePlayerLabel = new javax.swing.JLabel();

transferLabel = new javax.swing.JLabel();

ExitLabel = new javax.swing.JLabel();

manageTeamLabel = new javax.swing.JLabel();

teamManagePanel = new javax.swing.JPanel();

teamManagementLabel = new javax.swing.JLabel();

addTeamButton = new javax.swing.JButton();

updateTeamButton = new javax.swing.JButton();

deleteTeamButton = new javax.swing.JButton();

backButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

teamManagementTable = new javax.swing.JTable();

teamComboBox = new javax.swing.JComboBox();

jButton15 = new javax.swing.JButton();

PlayerManagePanel = new javax.swing.JPanel();

playerManagementLabel = new javax.swing.JLabel();

addPlayerButton = new javax.swing.JButton();

updatePlayerButton = new javax.swing.JButton();

deletePlayerButton = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

playerTable = new javax.swing.JTable();

jButton2 = new javax.swing.JButton();

playerComboBox = new javax.swing.JComboBox();

jButton16 = new javax.swing.JButton();

addPlayerPanel = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jComboBox1 = new javax.swing.JComboBox();

jLabel9 = new javax.swing.JLabel();

jComboBox3 = new javax.swing.JComboBox();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jLabel14 = new javax.swing.JLabel();

jLabel15 = new javax.swing.JLabel();

jButton4 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

physicalComboBox = new javax.swing.JComboBox();

speedComboBox = new javax.swing.JComboBox();

mentalComboBox = new javax.swing.JComboBox();

defendingComboBox = new javax.swing.JComboBox();

attackingComboBox = new javax.swing.JComboBox();

technicalComboBox = new javax.swing.JComboBox();

jLabel79 = new javax.swing.JLabel();

jComboBox2 = new javax.swing.JComboBox();

jLabel80 = new javax.swing.JLabel();

teamPreferenceComboBox = new javax.swing.JComboBox();

addTeamPanel = new javax.swing.JPanel();

jLabel16 = new javax.swing.JLabel();

teamNameLabel = new javax.swing.JLabel();

teamNameTextField = new javax.swing.JTextField();

yearFoundedLabel = new javax.swing.JLabel();

yearFoundedTextField = new javax.swing.JTextField();

valueLabel = new javax.swing.JLabel();

valueTextField = new javax.swing.JTextField();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

playerProfilePanel = new javax.swing.JPanel();

jLabel17 = new javax.swing.JLabel();

jLabel18 = new javax.swing.JLabel();

jLabel19 = new javax.swing.JLabel();

jLabel20 = new javax.swing.JLabel();

jLabel21 = new javax.swing.JLabel();

jLabel25 = new javax.swing.JLabel();

jLabel26 = new javax.swing.JLabel();

jLabel27 = new javax.swing.JLabel();

jLabel35 = new javax.swing.JLabel();

jLabel36 = new javax.swing.JLabel();

jLabel37 = new javax.swing.JLabel();

jLabel38 = new javax.swing.JLabel();

jLabel39 = new javax.swing.JLabel();

jLabel40 = new javax.swing.JLabel();

jLabel41 = new javax.swing.JLabel();

jLabel42 = new javax.swing.JLabel();

jLabel43 = new javax.swing.JLabel();

jButton3 = new javax.swing.JButton();

updatePlayerPanel = new javax.swing.JPanel();

jLabel22 = new javax.swing.JLabel();

jLabel23 = new javax.swing.JLabel();

jLabel24 = new javax.swing.JLabel();

jTextField4 = new javax.swing.JTextField();

jTextField5 = new javax.swing.JTextField();

jLabel28 = new javax.swing.JLabel();

jLabel29 = new javax.swing.JLabel();

jLabel30 = new javax.swing.JLabel();

jLabel31 = new javax.swing.JLabel();

jLabel32 = new javax.swing.JLabel();

jLabel33 = new javax.swing.JLabel();

jLabel34 = new javax.swing.JLabel();

jButton9 = new javax.swing.JButton();

jButton10 = new javax.swing.JButton();

physicalComboBox1 = new javax.swing.JComboBox();

speedComboBox1 = new javax.swing.JComboBox();

mentalComboBox1 = new javax.swing.JComboBox();

defendingComboBox1 = new javax.swing.JComboBox();

attackingComboBox1 = new javax.swing.JComboBox();

technicalComboBox1 = new javax.swing.JComboBox();

updateTeamPanel = new javax.swing.JPanel();

jLabel44 = new javax.swing.JLabel();

jLabel45 = new javax.swing.JLabel();

jTextField6 = new javax.swing.JTextField();

jLabel46 = new javax.swing.JLabel();

jTextField7 = new javax.swing.JTextField();

jLabel47 = new javax.swing.JLabel();

jTextField8 = new javax.swing.JTextField();

jButton11 = new javax.swing.JButton();

jButton12 = new javax.swing.JButton();

searchPanel = new javax.swing.JPanel();

jTextField9 = new javax.swing.JTextField();

transferPanel = new javax.swing.JPanel();

jLabel48 = new javax.swing.JLabel();

jLabel49 = new javax.swing.JLabel();

jLabel50 = new javax.swing.JLabel();

jLabel51 = new javax.swing.JLabel();

jLabel52 = new javax.swing.JLabel();

jLabel53 = new javax.swing.JLabel();

jLabel54 = new javax.swing.JLabel();

jLabel55 = new javax.swing.JLabel();

teamTransferComboBox = new javax.swing.JComboBox();

jButton8 = new javax.swing.JButton();

jButton13 = new javax.swing.JButton();

playerTransferComboBox = new javax.swing.JComboBox();

jButton14 = new javax.swing.JButton();

fixturePanel = new javax.swing.JPanel();

jLabel58 = new javax.swing.JLabel();

jLabel59 = new javax.swing.JLabel();

jLabel60 = new javax.swing.JLabel();

jLabel61 = new javax.swing.JLabel();

jLabel62 = new javax.swing.JLabel();

jLabel63 = new javax.swing.JLabel();

jLabel64 = new javax.swing.JLabel();

jLabel65 = new javax.swing.JLabel();

jLabel66 = new javax.swing.JLabel();

jLabel67 = new javax.swing.JLabel();

jLabel68 = new javax.swing.JLabel();

jLabel69 = new javax.swing.JLabel();

jLabel70 = new javax.swing.JLabel();

jLabel71 = new javax.swing.JLabel();

jLabel72 = new javax.swing.JLabel();

jLabel73 = new javax.swing.JLabel();

jLabel74 = new javax.swing.JLabel();

jLabel75 = new javax.swing.JLabel();

jLabel76 = new javax.swing.JLabel();

jLabel77 = new javax.swing.JLabel();

jLabel78 = new javax.swing.JLabel();

jButton17 = new javax.swing.JButton();

jButton18 = new javax.swing.JButton();

jButton19 = new javax.swing.JButton();

jButton25 = new javax.swing.JButton();

leagueTablePanel = new javax.swing.JPanel();

jLabel81 = new javax.swing.JLabel();

jScrollPane3 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton20 = new javax.swing.JButton();

matchResultPanel = new javax.swing.JPanel();

jLabel82 = new javax.swing.JLabel();

jLabel83 = new javax.swing.JLabel();

jLabel84 = new javax.swing.JLabel();

jComboBox4 = new javax.swing.JComboBox();

jLabel85 = new javax.swing.JLabel();

jComboBox5 = new javax.swing.JComboBox();

jLabel86 = new javax.swing.JLabel();

jComboBox6 = new javax.swing.JComboBox();

jLabel87 = new javax.swing.JLabel();

jLabel88 = new javax.swing.JLabel();

jComboBox7 = new javax.swing.JComboBox();

jLabel89 = new javax.swing.JLabel();

jComboBox8 = new javax.swing.JComboBox();

jLabel90 = new javax.swing.JLabel();

jComboBox9 = new javax.swing.JComboBox();

jButton21 = new javax.swing.JButton();

jButton22 = new javax.swing.JButton();

jLabel91 = new javax.swing.JLabel();

jComboBox10 = new javax.swing.JComboBox();

jLabel92 = new javax.swing.JLabel();

jComboBox11 = new javax.swing.JComboBox();

setMatch = new javax.swing.JPanel();

jLabel56 = new javax.swing.JLabel();

jLabel57 = new javax.swing.JLabel();

jComboBox12 = new javax.swing.JComboBox();

jLabel93 = new javax.swing.JLabel();

jComboBox13 = new javax.swing.JComboBox();

jLabel94 = new javax.swing.JLabel();

jTextField10 = new javax.swing.JTextField();

jLabel95 = new javax.swing.JLabel();

jTextField11 = new javax.swing.JTextField();

jButton23 = new javax.swing.JButton();

jButton24 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

getContentPane().setLayout(new java.awt.GridBagLayout());

jPanel1.setLayout(new java.awt.CardLayout());

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel1.setText("Login");

userNameLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

userNameLabel.setText("User name ");

passwordLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

passwordLabel.setText("Password");

signInButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

signInButton.setText("Sign in");

signInButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

signInButtonActionPerformed(evt);

}

});

ExitButton.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

ExitButton.setText("Exit");

ExitButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

ExitButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(54, 54, 54)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(passwordLabel, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(userNameLabel, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(26, 26, 26)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(userNameTextField)

.addComponent(jPasswordField1, javax.swing.GroupLayout.DEFAULT\_SIZE, 142, Short.MAX\_VALUE)))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(62, 62, 62)

.addComponent(signInButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 98, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(96, 96, 96)

.addComponent(ExitButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 104, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(102, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(57, 57, 57)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(userNameTextField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(userNameLabel))

.addGap(57, 57, 57)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(passwordLabel)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 118, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(signInButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(ExitButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(28, 28, 28))

);

javax.swing.GroupLayout LoginPanelLayout = new javax.swing.GroupLayout(LoginPanel);

LoginPanel.setLayout(LoginPanelLayout);

LoginPanelLayout.setHorizontalGroup(

LoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(LoginPanelLayout.createSequentialGroup()

.addGroup(LoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(LoginPanelLayout.createSequentialGroup()

.addGap(55, 55, 55)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 79, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(LoginPanelLayout.createSequentialGroup()

.addGap(149, 149, 149)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(104, Short.MAX\_VALUE))

);

LoginPanelLayout.setVerticalGroup(

LoginPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(LoginPanelLayout.createSequentialGroup()

.addGap(59, 59, 59)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(190, Short.MAX\_VALUE))

);

jPanel1.add(LoginPanel, "loginPanelCard");

jLabel2.setFont(new java.awt.Font("Nirmala UI", 3, 36)); // NOI18N

jLabel2.setText("FOOTBALL MANAGEMENT SYSTEM");

playerButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/football/management/player.jpg"))); // NOI18N

playerButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

playerButtonActionPerformed(evt);

}

});

teamButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/football/management/team.png"))); // NOI18N

teamButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teamButtonActionPerformed(evt);

}

});

searchButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/football/management/search.png"))); // NOI18N

searchButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

searchButtonActionPerformed(evt);

}

});

fixtureButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/football/management/icon-fixtures-football.png"))); // NOI18N

fixtureButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

fixtureButtonActionPerformed(evt);

}

});

transferButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/football/management/rsz\_transfer.png"))); // NOI18N

transferButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

transferButtonActionPerformed(evt);

}

});

exitButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/football/management/denied\_128px.png"))); // NOI18N

exitButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

exitButtonActionPerformed(evt);

}

});

fixtureLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

fixtureLabel.setText("Fixtures\n");

SearchLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

SearchLabel.setText("Standing");

managePlayerLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

managePlayerLabel.setText("Manage Players");

transferLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

transferLabel.setText("Transfer");

ExitLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

ExitLabel.setText("Log out");

manageTeamLabel.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

manageTeamLabel.setText("Manage Teams");

javax.swing.GroupLayout MainMenuLayout = new javax.swing.GroupLayout(MainMenu);

MainMenu.setLayout(MainMenuLayout);

MainMenuLayout.setHorizontalGroup(

MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MainMenuLayout.createSequentialGroup()

.addGap(101, 101, 101)

.addComponent(fixtureLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 69, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(153, 153, 153)

.addComponent(transferLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 66, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(ExitLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(105, 105, 105))

.addGroup(MainMenuLayout.createSequentialGroup()

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MainMenuLayout.createSequentialGroup()

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(MainMenuLayout.createSequentialGroup()

.addGap(64, 64, 64)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(fixtureButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(searchButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(54, 54, 54)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(teamButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(transferButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(52, 52, 52))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, MainMenuLayout.createSequentialGroup()

.addGap(93, 93, 93)

.addComponent(SearchLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(manageTeamLabel)

.addGap(98, 98, 98)))

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(playerButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(exitButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(MainMenuLayout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(managePlayerLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 135, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGroup(MainMenuLayout.createSequentialGroup()

.addGap(64, 64, 64)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 606, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

MainMenuLayout.setVerticalGroup(

MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MainMenuLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(64, 64, 64)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(teamButton, javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(searchButton)

.addComponent(playerButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 129, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(SearchLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(managePlayerLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(manageTeamLabel))

.addGap(54, 54, 54)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(transferButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(exitButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(fixtureButton, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 121, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(fixtureLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(MainMenuLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(transferLabel)

.addComponent(ExitLabel)))

.addGap(64, 64, 64))

);

jPanel1.add(MainMenu, "mainMenuCard");

teamManagementLabel.setFont(new java.awt.Font("Nirmala UI", 1, 36)); // NOI18N

teamManagementLabel.setText("Team Management");

addTeamButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

addTeamButton.setText("Add Team ");

addTeamButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addTeamButtonActionPerformed(evt);

}

});

updateTeamButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

updateTeamButton.setText("Update Team");

updateTeamButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

updateTeamButtonActionPerformed(evt);

}

});

deleteTeamButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

deleteTeamButton.setText("Match Result");

deleteTeamButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteTeamButtonActionPerformed(evt);

}

});

backButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

backButton.setText("Back");

backButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

backButtonActionPerformed(evt);

}

});

teamManagementTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Team ID", "Team Name", "Year Founded", "Value"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.Integer.class, java.lang.Integer.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(teamManagementTable);

if (teamManagementTable.getColumnModel().getColumnCount() > 0) {

teamManagementTable.getColumnModel().getColumn(0).setResizable(false);

teamManagementTable.getColumnModel().getColumn(1).setResizable(false);

teamManagementTable.getColumnModel().getColumn(3).setResizable(false);

try {

statement=connection.createStatement();

resultSet = statement.executeQuery("SELECT Team\_Name FROM groupC");

resultSet.next();

jLabel70.setText(resultSet.getString("Team\_Name"));

resultSet.next();

jLabel71.setText(resultSet.getString("Team\_Name"));

resultSet.next();

jLabel72.setText(resultSet.getString("Team\_Name"));

resultSet.next();

jLabel73.setText(resultSet.getString("Team\_Name"));

} catch (SQLException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

try {

statement=connection.createStatement();

resultSet = statement.executeQuery("SELECT Team\_Name FROM groupD");

resultSet.next();

jLabel75.setText(resultSet.getString("Team\_Name"));

resultSet.next();

jLabel76.setText(resultSet.getString("Team\_Name"));

resultSet.next();

jLabel77.setText(resultSet.getString("Team\_Name"));

resultSet.next();

jLabel78.setText(resultSet.getString("Team\_Name"));

} catch (SQLException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

count=1;

String teamOne=null;

String teamTwo=null;

String category=null;

for(int a=1;a<=4;a++){

String groupName=null;

if(a==1){

groupName="groupa";

category="A";

}else if(a==2){

groupName="groupb";

category="B";

}else if(a==3){

groupName="groupc";

category="C";

}else{

groupName="groupd";

category="D";

}

String alu="SELECT Group\_ID FROM "+groupName+" WHERE Team\_ID="+a;

System.out.println("alu="+alu);

int groupId1=0;

try {

statement=connection.createStatement();

resultSet=statement.executeQuery(alu);

resultSet.next();

groupId1=resultSet.getInt(1);

} catch (SQLException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

int d=1;

for(int i=1;i<=4;i++){

for(int j=i+1;j<=4;j++){

// int groupId1=a\*10 +i;

int groupId2=groupId1 +j -d;

try {

String abc="SELECT Team\_Name FROM "

+groupName+" WHERE Group\_ID IN("+groupId2+", "+groupId1+")";

System.out.println("abc="+abc);

statement =connection.createStatement();

resultSet=statement.executeQuery(abc);

resultSet.next();

teamOne=resultSet.getString("Team\_Name").toString();

resultSet.next();

teamTwo=resultSet.getString("Team\_Name").toString();

String query="INSERT INTO result (Team\_One,Team\_two,Category,Year) values("

+"'"+teamOne+"'"+","+"'"+teamTwo+"' ,"+"'"+category+"' ,"+year+" )";

System.out.println("groupId1="+groupId1+" groupId2="+groupId2);

System.out.println("\n");

System.out.println("query="+query+" \n");

statement.execute(query);

} catch (SQLException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

d++;

groupId1++;

}

}

}//GEN-LAST:event\_jButton17ActionPerformed

private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton18ActionPerformed

// TODO add your handling code here:

if(count == 0){

JOptionPane.showMessageDialog(this, "There is no Fixture to clear");

return;

}

Date date = new Date();

LocalDate localDate = date.toInstant().atZone(ZoneId.systemDefault()).toLocalDate();

int year = localDate.getYear();

try {

String delete="delete from result where Year="+year;

statement = connection.createStatement();

statement.executeUpdate("delete from groupA");

statement.executeUpdate("delete from groupB");

statement.executeUpdate("delete from groupC");

statement.executeUpdate("delete from groupD");

statement.executeUpdate(delete);

JOptionPane.showMessageDialog(this, "Fixture has been cleared");

} catch (SQLException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

// Group A

jLabel60.setText("Group A1");

jLabel61.setText("Group A2");

jLabel62.setText("Group A3");

jLabel63.setText("Group A4");

//Group B

jLabel65.setText("Group B1");

jLabel66.setText("Group B2");

jLabel67.setText("Group B3");

jLabel68.setText("Group B4");

//Group C

jLabel70.setText("Group C1");

jLabel71.setText("Group C2");

jLabel72.setText("Group C3");

jLabel73.setText("Group C4");

//Group D

jLabel75.setText("Group D1");

jLabel76.setText("Group D2");

jLabel77.setText("Group D3");

jLabel78.setText("Group D4");

count=0;

}//GEN-LAST:event\_jButton18ActionPerformed

private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton19ActionPerformed

// TODO add your handling code here:

CardLayout card = (CardLayout) jPanel1.getLayout();

card.show(jPanel1, "mainMenuCard");

}//GEN-LAST:event\_jButton19ActionPerformed

private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton20ActionPerformed

// TODO add your handling code here:

CardLayout card = (CardLayout) jPanel1.getLayout();

card.show(jPanel1, "mainMenuCard");

}//GEN-LAST:event\_jButton20ActionPerformed

private void jButton22ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton22ActionPerformed

try {

// TODO add your handling code here:

int pointHome,pointAway;

int resultHome,resultAway;

String matchResult;

statement = connection.createStatement();

//HOME

String updateHome=null;

if(jComboBox5.getSelectedItem().toString().equals("Win")){

pointHome=3;

matchResult=jComboBox4.getSelectedItem().toString();

resultSet=statement.executeQuery("SELECT Win From league\_standing WHERE Team\_Name ="+"'"+jComboBox4.getSelectedItem().toString()+"'");

resultSet.next();

resultHome=resultSet.getInt("Win") + 1;

updateHome ="UPDATE league\_standing SET Win="+resultHome+" WHERE Team\_Name ="+"'"+jComboBox4.getSelectedItem().toString()+"'";

}else if(jComboBox5.getSelectedItem().toString().equals("Draw")){

pointHome=1;

matchResult="Draw";

resultSet=statement.executeQuery("SELECT Draw From league\_standing WHERE Team\_Name ="+"'"+jComboBox4.getSelectedItem().toString()+"'");

resultSet.next();

resultHome=resultSet.getInt("Draw") + 1;

updateHome ="UPDATE league\_standing SET Draw="+resultHome+" WHERE Team\_Name ="+"'"+jComboBox4.getSelectedItem().toString()+"'";

}else{

pointHome=0;

matchResult =jComboBox7.getSelectedItem().toString();

resultSet=statement.executeQuery("SELECT Loss From league\_standing WHERE Team\_Name ="+"'"+jComboBox4.getSelectedItem().toString()+"'");

resultSet.next();

resultHome=resultSet.getInt("Loss") + 1;

updateHome ="UPDATE league\_standing SET Loss="+resultHome+" WHERE Team\_Name ="+"'"+jComboBox4.getSelectedItem().toString()+"'";

}

statement.executeUpdate(updateHome);

// AWAY

String updateAway=null;

if(jComboBox8.getSelectedItem().toString().equals("Win")){

pointAway=3;

resultSet=statement.executeQuery("SELECT Win From league\_standing WHERE Team\_Name ="+"'"+jComboBox7.getSelectedItem().toString()+"'");

resultSet.next();

resultHome=resultSet.getInt("Win") + 1;

updateAway ="UPDATE league\_standing SET Win="+resultHome+" WHERE Team\_Name ="+"'"+jComboBox7.getSelectedItem().toString()+"'";

}else if(jComboBox8.getSelectedItem().toString().equals("Draw")){

pointAway=1;

resultSet=statement.executeQuery("SELECT Draw From league\_standing WHERE Team\_Name ="+"'"+jComboBox7.getSelectedItem().toString()+"'");

resultSet.next();

resultHome=resultSet.getInt("Draw") + 1;

updateAway ="UPDATE league\_standing SET Draw="+resultHome+" WHERE Team\_Name ="+"'"+jComboBox7.getSelectedItem().toString()+"'";

}else{

pointAway=0;

resultSet=statement.executeQuery("SELECT Loss From league\_standing WHERE Team\_Name ="+"'"+jComboBox7.getSelectedItem().toString()+"'");

resultSet.next();

resultHome=resultSet.getInt("Loss") + 1;

updateAway ="UPDATE league\_standing SET Loss="+resultHome+" WHERE Team\_Name ="+"'"+jComboBox7.getSelectedItem().toString()+"'";

}

statement.executeUpdate(updateAway);

System.out.println("pointHome:"+pointHome+"updateHome:"+updateHome+"pointAway:"+pointAway+"updateAway:"+updateAway);

//UPDATE result TABLE

String result = "INSERT INTO result (Match\_Result,Goal\_T1,Goal\_T2) values("+"'"+matchResult+"'"+

","+Integer.parseInt(jComboBox10.getSelectedItem().toString())+","+Integer.parseInt(jComboBox11.getSelectedItem().toString())

+") WHERE Team\_One="+"'"+jComboBox4.getSelectedItem().toString()+"'"+" AND Team\_Two="+"'"+jComboBox7.getSelectedItem().toString()+"'";

System.out.println("result:"+result);

statement.execute(result);

//UPDATE score TABLE

//Home

int goalDefHome,goalScoredHome,goalConcededHome;

resultSet=statement.executeQuery("SELECT Goal\_Scored, Goal\_Conceded,Goal\_Def FROM score WHERE Team\_Name="+

"'"+jComboBox4.getSelectedItem().toString()+"'" );

resultSet.next();

goalScoredHome=resultSet.getInt("Goal\_Scored") +(Integer.parseInt(jComboBox10.getSelectedItem().toString()));

goalConcededHome=resultSet.getInt("Goal\_Conceded") +(Integer.parseInt(jComboBox11.getSelectedItem().toString()));

goalDefHome=resultSet.getInt("Goal\_Def")+(Integer.parseInt(jComboBox10.getSelectedItem().toString()) - Integer.parseInt(jComboBox11.getSelectedItem().toString()));

String scoreHome="UPDATE score SET Goal\_Scored="

+goalScoredHome+", Goal\_Conceded="

+ goalConcededHome+", Goal\_Def="+goalDefHome+" WHERE Team\_Name="+

"'"+jComboBox4.getSelectedItem().toString()+"'" ;

System.out.println("scoreHome:"+scoreHome);

statement.executeUpdate(scoreHome);

// Away

int goalDefAway,goalScoredAway,goalConcededAway;

resultSet=statement.executeQuery("SELECT Goal\_Scored, Goal\_Conceded,Goal\_Def FROM score WHERE Team\_Name="+

"'"+jComboBox7.getSelectedItem().toString()+"'" );

resultSet.next();

goalScoredAway=resultSet.getInt("Goal\_Scored") +(Integer.parseInt(jComboBox11.getSelectedItem().toString()));

goalConcededAway=resultSet.getInt("Goal\_Conceded") +(Integer.parseInt(jComboBox10.getSelectedItem().toString()));

goalDefAway=resultSet.getInt("Goal\_Def")+(Integer.parseInt(jComboBox11.getSelectedItem().toString()) - Integer.parseInt(jComboBox10.getSelectedItem().toString()));

String scoreAway="UPDATE score SET Goal\_Scored="

+ goalScoredAway+", Goal\_Conceded="

+goalConcededAway+", Goal\_Def="+goalDefAway+" WHERE Team\_Name="

+ "'"+jComboBox4.getSelectedItem().toString()+"'" ;

System.out.println("scoreAway:"+scoreAway);

statement.executeUpdate(scoreAway);

} catch (SQLException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

jComboBox4.setSelectedItem("");

jComboBox7.setSelectedItem("");

JOptionPane.showMessageDialog(this, "Match Result has been updated");

}//GEN-LAST:event\_jButton22ActionPerformed

private void jButton21ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton21ActionPerformed

// TODO add your handling code here:

CardLayout card = (CardLayout) jPanel1.getLayout();

card.show(jPanel1, "teamManageCard");

}//GEN-LAST:event\_jButton21ActionPerformed

private void teamTransferComboBoxActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teamTransferComboBoxActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_teamTransferComboBoxActionPerformed

private void jButton23ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton23ActionPerformed

try {

// TODO add your handling code here:

statement = connection.createStatement();

String query="INSERT INTO result(Match\_Date,Match\_Time) values("

+"'"+jTextField10.getText()+"'"

+", "+"'"+jTextField11.getText()+"' )"+

"WHERE Team\_One="+"'"+jComboBox12.getSelectedItem().toString()+"' "+" AND Team\_Two="

+"'"+jComboBox13.getSelectedItem().toString()+"'";

System.out.println("query:"+query);

statement.execute(query);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "There no match between these team or invalid data");

}

jComboBox12.setSelectedItem("");

jComboBox13.setSelectedItem("");

jTextField10.setText("yy-mm-dd");

jTextField11.setText("hr:min-am/pm");

JOptionPane.showMessageDialog(this, "Match date and time have been set");

}//GEN-LAST:event\_jButton23ActionPerformed

private void jButton25ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton25ActionPerformed

// TODO add your handling code here:

CardLayout card = (CardLayout) jPanel1.getLayout();

card.show(jPanel1, "card16");

}//GEN-LAST:event\_jButton25ActionPerformed

private void jButton24ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton24ActionPerformed

// TODO add your handling code here:

CardLayout card = (CardLayout) jPanel1.getLayout();

card.show(jPanel1, "card13");

}//GEN-LAST:event\_jButton24ActionPerformed

Connection connection = null;

Statement statement = null;

ResultSet resultSet = null;

private String userName = "root";

private String password = null;

String DATABASE\_URL = "jdbc:mysql://localhost/footballmanagement2";

public Connection startConnection(){

try {

// String DATABASE\_URL = "jdbc:mysql://localhost:3306/mynewdatabase";

Class.forName("com.mysql.jdbc.Driver").newInstance();

} catch (ClassNotFoundException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

try {

connection = DriverManager.getConnection(DATABASE\_URL,userName,password);

System.out.println("connected");

} catch (SQLException ex) {

Logger.getLogger(DatabaseConnection.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, ex);

}

return connection;

}

public void closeConnection(){

try {

connection.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseConnection.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

public void autoComplete(){

final JTextField textfield = (JTextField) playerTransferComboBox.getEditor().getEditorComponent();

textfield.addKeyListener(new KeyAdapter() {

public void keyReleased(KeyEvent ke) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

comboFilter(textfield.getText());

try {

Thread.sleep(1000);

} catch (InterruptedException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

}

});

}

public void comboFilter(String enteredText) {

ArrayList<String> filterArray= new ArrayList<String>();

String str1="";

try

{

String str="SELECT First\_Name FROM player\_info WHERE First\_Name LIKE '"+enteredText+"%'";

//Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

//Connection con=DriverManager.getConnection("jdbc:odbc:ds1", "sa" , "creative");

Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(str);

while(rs.next())

{

str1=rs.getString("First\_Name");

filterArray.add(str1);

}

}

catch(Exception ex)

{

System.out.println("error"+ex);

}

}

private void searchPlayer() {

//starting--

SearchBoxModel sbm = new SearchBoxModel(playerTransferComboBox,playerName);

//set the model on the combobox

playerTransferComboBox.setModel(sbm);

//set the model as the item listener also

playerTransferComboBox.addItemListener(sbm);

SearchBoxModel sbm2 = new SearchBoxModel(playerComboBox,playerName);

playerComboBox.setModel(sbm2);

playerComboBox.addItemListener(sbm2);

}

public void searchCompany(){

SearchBoxModel sbmc = new SearchBoxModel(teamTransferComboBox, teamName);

teamTransferComboBox.setModel(sbmc);

teamTransferComboBox.addItemListener(sbmc);

SearchBoxModel sbmc1 = new SearchBoxModel(teamComboBox, teamName);

teamComboBox.setModel(sbmc1);

teamComboBox.addItemListener(sbmc1);

SearchBoxModel sbmc2 = new SearchBoxModel(teamPreferenceComboBox, teamName);

teamPreferenceComboBox.setModel(sbmc2);

teamPreferenceComboBox.addItemListener(sbmc2);

SearchBoxModel sbmc3 = new SearchBoxModel(jComboBox3, teamName);

jComboBox3.setModel(sbmc3);

jComboBox3.addItemListener(sbmc3);

SearchBoxModel sbmc4 = new SearchBoxModel(jComboBox4, teamName);

jComboBox4.setModel(sbmc4);

jComboBox4.addItemListener(sbmc4);

SearchBoxModel sbmc5 = new SearchBoxModel(jComboBox7, teamName);

jComboBox7.setModel(sbmc5);

jComboBox7.addItemListener(sbmc5);

SearchBoxModel sbmc6 = new SearchBoxModel(jComboBox13, teamName);

jComboBox13.setModel(sbmc6);

jComboBox13.addItemListener(sbmc6);

SearchBoxModel sbmc7 = new SearchBoxModel(jComboBox12, teamName);

jComboBox12.setModel(sbmc7);

jComboBox12.addItemListener(sbmc7);

}

private void createArrayListDB(){

try {

String sql = "select First\_Name from player\_info";

Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

while(rs.next())

{

playerName.add(rs.getString("First\_Name"));

System.out.println(rs.getString("First\_Name"));

}

} catch (SQLException ex) {

}

}

public void createArrayListDB2(){

try {

String sql = "SELECT Team\_Name FROM team";

Statement stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

while(rs.next())

{

teamName.add(rs.getString("Team\_Name"));

}

} catch (SQLException ex) {

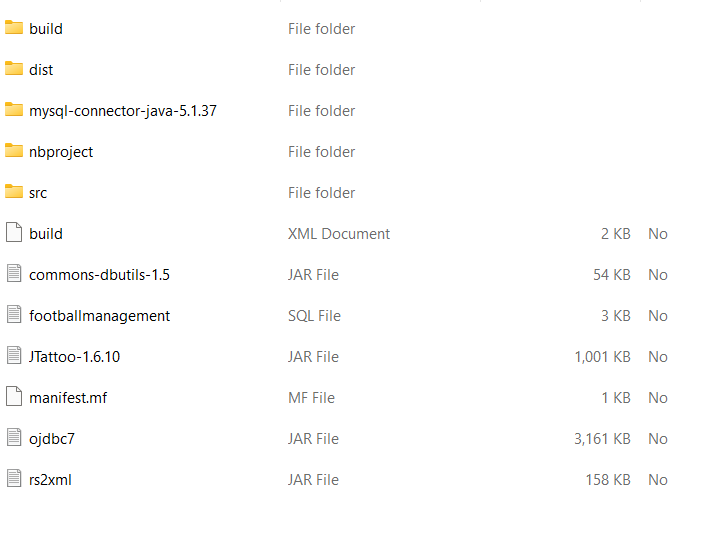
}

}

}

## GitHub links:

<https://github.com/sr376/football-management>



**Results:**

I successfully completed this PROJECT “Soccer League Database Management system”.

## Discussion and Future work

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create a android app by which I can control my project on my hand and connect to it.

**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>