JAVA AWT BASED- TECH QUIZ MANAGEMENT SYSTEM FORM- SQL

CONNECTIVITY USING JDBC

A
Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By
B. Mahalaxmi <1602-18-737-080>

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020

BONAFIDE CERTIFICATE

Certified that this project report titled “Tech Quiz database management system” is bonafide work of Mrs B. Mahalaxmi, who carried out the mini project work under my supervision.

Certified further that, to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion or any other candidate.

Signature of the Examiner

B.LEELAVATHY

Lecturer

Department of Information Technology

ABSTRACT

This project is about “Techie Quiz”, which can accessed by students by providing their login id and password. New users can sign up giving their name, mail Id , student Id and college name. Each student have an option of attempting either a python quiz or C quiz or even both. Students can also attempt the quiz more than once. Score will be evaluated and displayed. Quiz consists of multiple choice and programming questions. This helps students to improve their programming skills and spontaneity. Since smart phones are used by almost all the students, it is even easy for them to take quiz even without pc.

INTRODUCTION

REQUIREMENTS FOR TECH QUIZ DATABASE MANAGEMENT SYSTEM:

List of Tables:

- Students
- Login
- Quiz
- Score
- Students_login
- Students_quiz
- Students_score

List of attributes with their domain types:

ENTITY	ATTRIBUTES	DOMAIN
Students	1. STUDENTID 2. SNAME 3. MAILID 4. COLLEGE	VARCHAR2(10) VARCHAR2(10) VARCHAR2(20) VARCHAR2(10)
Login	1. LOGINID 2. USER_PASSWORD 3. TYPE	VARCHAR2(10) VARCHAR2(15) VARCHAR2(5)
Quiz	1. QUIZID 2. QUIZLANG 3. QNAME	VARCHAR2(20) VARCHAR2(10) VARCHAR2(10)
Score	1. SCRID 2. MARKS	VARCHAR2(10) NUMBER(2)

Students_quiz	1. STUDENTID 2. QUIZID 3. NOOFATTEMPTS	VARCHAR2(10) VARCHAR2(10) NUMBER(2)
Students_login	1. STUDENTID 2. LOGINID	VARCHAR2(10) VARCHAR2(10)
Students_score	1. STUDENTID 2. SCRID	VARCHAR2(10) VARCHAR2(10)

AIM OF THE PROJECT:

To create a Java GUI based form for the project **Tech Quiz Database Management System** which takes the values like : student Id,student name, Mail ID, College,login ID, password from the student. These are the values to be updated in the database using JDBC connectivity .

The values entered (insertion,deletion,updation) by the user for the respective table in **GUI** should be updated in the database using **JDBC**.

ARCHITECTURE AND TECHNOLOGY USED:

Java Eclipse, Oracle 11g Database, Java SE version 7, SQL*Plus.

SQL PLUS is the most basic Oracle Database utility with a basic command-line interface, commonly used by users,administrators and programmers.

The interface of SQL Plus is used for creating the database. DDL and DML commands are implemented for operations being executed. The details of students, their logins, quiz, score are stored in the form of tables in the database.

Eclipse is an integrated development environment(IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in java and its primary use is for developing java applicatons, but it may also be used to develop applications in other programming languages via plug-ins, including Erlang, Javascripts etc.

The front end application code is written in **"Java"** using eclipse. The portal for front end application is designed through Eclipse, runs and has the capacity to connect with the database which has data inserted using SQL.

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window
based applications in java.

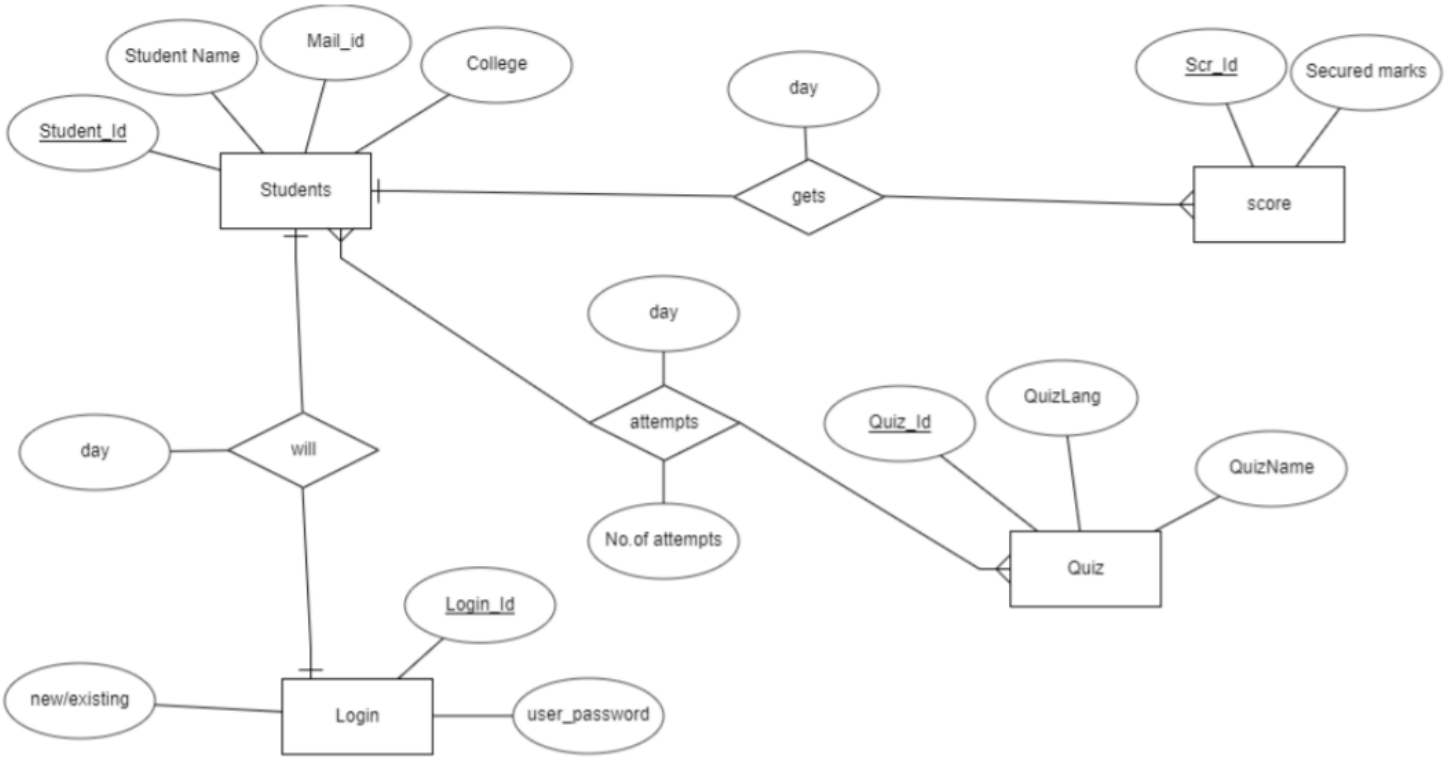
Java AWT components are platform-dependent i.e. components are displayed according
to the view of operating system. AWT is heavyweight i.e. its components are using the
resources of OS.

The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

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.

ER Diagram


Student Name
Mail_id
College
Student_Id
Students
day
gets
Scr_Id
Secured marks
score
day
attempts
No.of attempts
Quiz_Id
QuizLang
QuizName
Quiz
day
will
Login_Id
new/existing
Login
user_password

JDBC Connectivity:

```
private void connToDb(){
try {

Class.forName(“oracle.jdbc.driver.OracleDriver”);
connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:xe","rachana","vasavi
");
statement = connection.createStatement();
} catch (SQLException connectException) {
System.out.println(connectException.getMessage());
System.out.println(connectException.getSQLState());
System.out.println(connectException.getErrorCode());
System.exit(1);
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
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.

IMPLEMENTATION:

Below is the code for the table Students

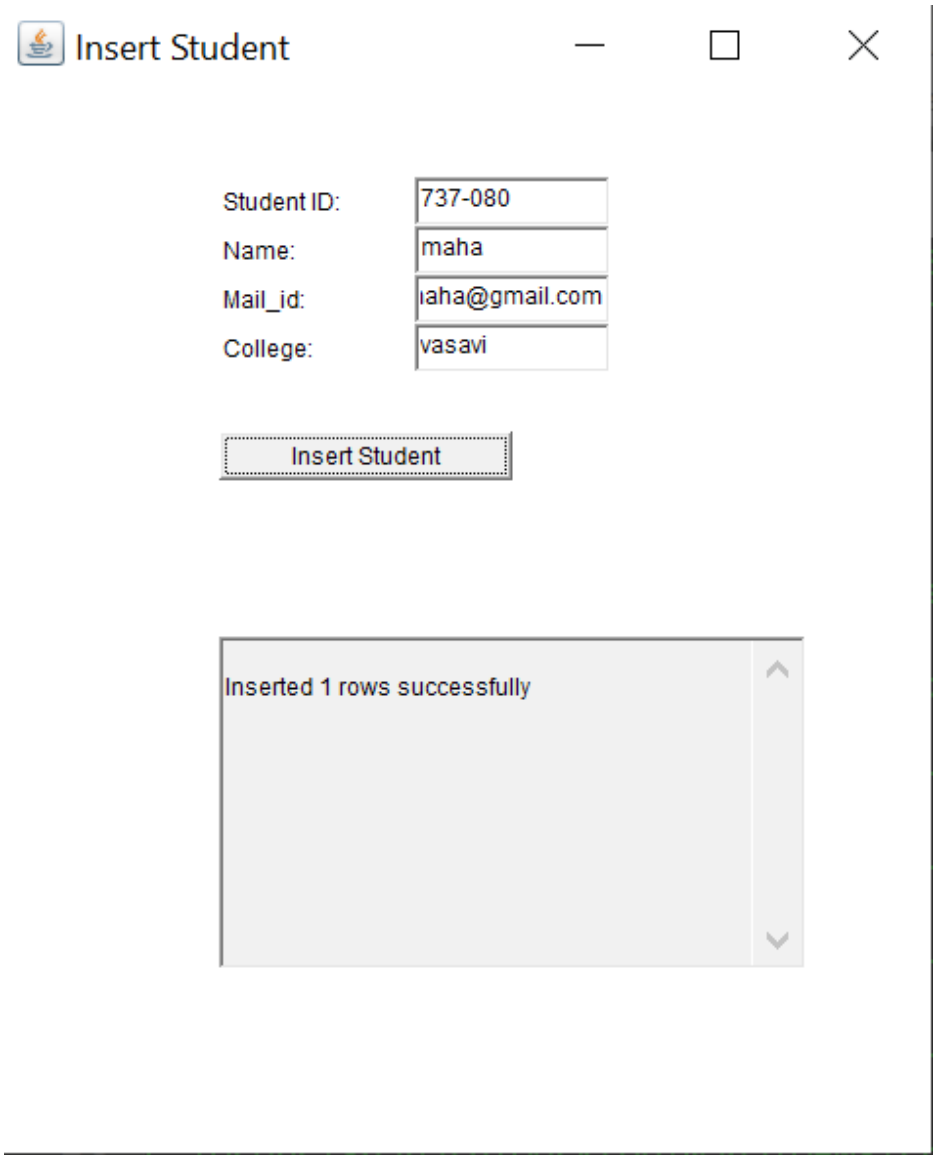
Insert Student:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class InsertStudent extends Frame
{
	Button insertStudentButton;
	TextField sidText, snameText, mailText, collegeText;
	TextArea errorText;
	Connection connection;
	Statement statement;
	public InsertStudent()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
	}

	public void connectToDB()
   {
		try
		{
```

```
                connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:SHARATH","maha","
2000");
                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 buildGUI()
        {
            //Handle Insert Account Button
            insertStudentButton = new Button("Insert Student");
            insertStudentButton.addActionListener(new ActionListener()
            {
                public void actionPerformed(ActionEvent e)
                {
                    try
                    {
                      //String query = "INSERT INTO sailors (SID,SNAME,
RATING, AGE) VALUES (2,'Divya',7,20)";
                      String query= "INSERT INTO students VALUES("+"'" +
sidText.getText() + "','" + snameText.getText() + "','" + mailText.getText() + "','" +
collegeText.getText() + "'" +")";
                      int i = statement.executeUpdate(query);
                      errorText.append("\nInserted " + i + " rows successfully");
                    }
                    catch (SQLException insertException)
                    {
                      displaySQLErrors(insertException);
                    }
                }
            });


            sidText = new TextField(15);
            snameText = new TextField(15);
```

```
        mailText = new TextField(15);
        collegeText = new TextField(15);


        errorText = new TextArea(10, 40);
        errorText.setEditable(false);

        Panel first = new Panel();
        first.setLayout(new GridLayout(4, 2));
        first.add(new Label("Student ID:"));
        first.add(sidText);
        first.add(new Label("Name:"));
        first.add(snameText);
        first.add(new Label("Mail_id:"));
        first.add(mailText);
        first.add(new Label("College:"));
        first.add(collegeText);
        first.setBounds(125,90,200,100);

        Panel second = new Panel(new GridLayout(4, 1));
        second.add(insertStudentButton);
second.setBounds(125,220,150,100);

        Panel third = new Panel();
        third.add(errorText);
        third.setBounds(125,320,300,200);

        setLayout(null);

        add(first);
        add(second);
        add(third);

        setTitle("Insert Gets");
        setSize(500, 600);
        setVisible(true);
  }



  private void displaySQLErrors(SQLException e)
  {
```

```
            errorText.append("\nSQLException: " + e.getMessage() + "\n");
            errorText.append("SQLState:     " + e.getSQLState() + "\n");
            errorText.append("VendorError:  " + e.getErrorCode() + "\n");
        }



}
```



```
SQL> select * from students;

STUDENTID  SNAME      MAILID               COLLEGE
---------- ---------- -------------------- ----------
737-121    Jay        Jay@gmail.com        vasavi
737-122    hari       hari@gmail.com       vasavi
737-123    sharath    sharath@gmail.com    vasavi
737-125    arjun      arjun@gmail.com      vasavi
737-124    aadhi      aadhi@gmail.com      vasavi

SQL> commit;

Commit complete.

SQL> select * from students;

STUDENTID  SNAME      MAILID               COLLEGE
---------- ---------- -------------------- ----------
737-080    maha       maha@gmail.com       vasavi
737-121    Jay        Jay@gmail.com        vasavi
737-122    hari       hari@gmail.com       vasavi
737-123    sharath    sharath@gmail.com    vasavi
737-125    arjun      arjun@gmail.com      vasavi
737-124    aadhi      aadhi@gmail.com      vasavi

6 rows selected.
```

Update Student:

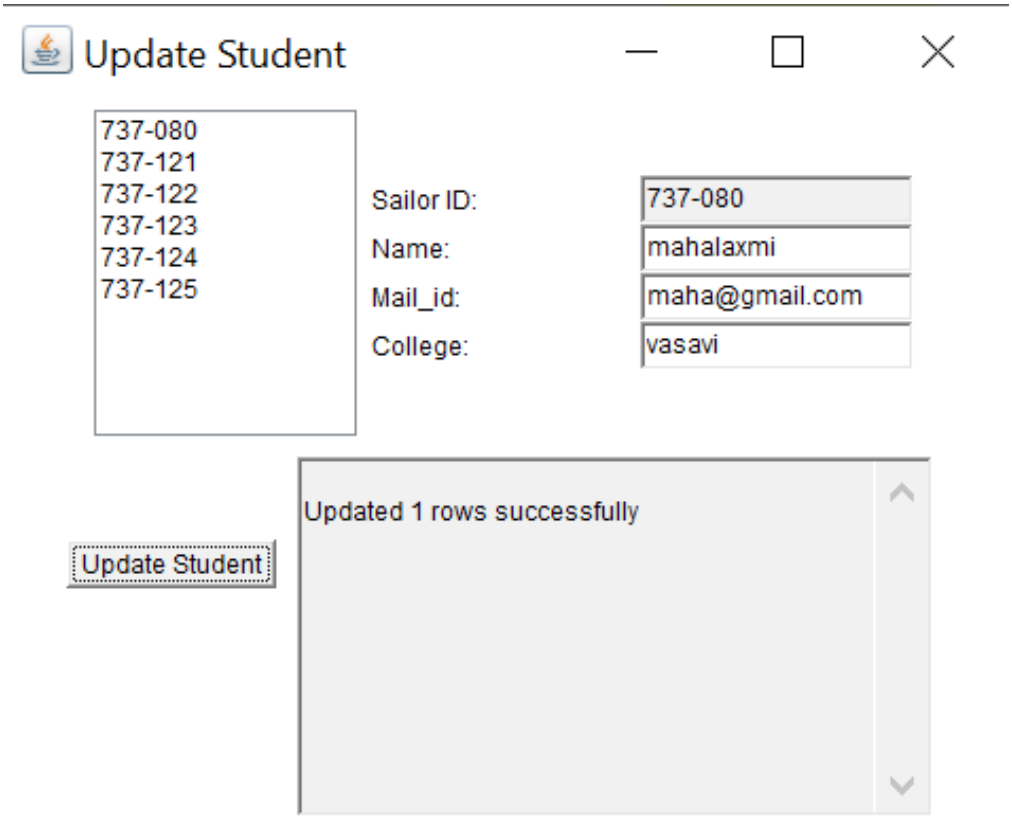
```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class UpdateStudent extends Frame
{
	Button updateStudentButton;
	List studentIDList;
	TextField sidText, snameText, mailText, collegeText;
	TextArea errorText;
	Connection connection;
	Statement statement;
	ResultSet rs;

	public UpdateStudent()
	{
```

```
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
	}

	public void connectToDB()
	{
		try
		{
		  connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:SHARATH","maha","
2000");
		  statement = connection.createStatement();

		}
		catch (SQLException connectException)
		{
		  System.out.println(connectException.getMessage());
		  System.out.println(connectException.getSQLState());
		  System.out.println(connectException.getErrorCode());
		  System.exit(1);
		}
	}

	private void loadStudents()
	{
		try
		{
		  rs = statement.executeQuery("SELECT STUDENTID FROM students");
		  while (rs.next())
		  {
			studentIDList.add(rs.getString("STUDENTID"));
		  }
		}
		catch (SQLException e)
```

```
            {
              displaySQLErrors(e);
            }
        }

        public void buildGUI()
        {
           studentIDList = new List(10);
                loadStudents();
                add(studentIDList);

                //When a list item is selected populate the text fields
                studentIDList.addItemListener(new ItemListener()
                {
                        public void itemStateChanged(ItemEvent e)
                        {
                                try
                                {
                                        rs = statement.executeQuery("SELECT * FROM
students where STUDENTID ='"+studentIDList.getSelectedItem()+"'");
                                        rs.next();
                                        sidText.setText(rs.getString("STUDENTID"));
                                        snameText.setText(rs.getString("SNAME"));
                                        mailText.setText(rs.getString("MAILID"));
                                        collegeText.setText(rs.getString("COLLEGE"));

                                }
                                catch (SQLException selectException)
                                {
                                        displaySQLErrors(selectException);
                                }
                        }
                });


                //Handle Update Sailor Button
                updateStudentButton = new Button("Update Student");
                updateStudentButton.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent e)
                        {
                                try
```

```
                        {
                                Statement statement = connection.createStatement();
                                int i = statement.executeUpdate("UPDATE students "
                                + "SET sname='" + snameText.getText() + "', "
                                + "mailid='" + mailText.getText() + "', "
                                + "college ='"+ collegeText.getText() + "' WHERE
studentid = '"
                                + studentIDList.getSelectedItem()+"'");

                                errorText.append("\nUpdated " + i + " rows
successfully");
                                studentIDList.removeAll();
                                loadStudents();
                        }
                        catch (SQLException insertException)
                        {
                                displaySQLErrors(insertException);
                        }
                }
        });

        sidText = new TextField(15);
        sidText.setEditable(false);
        snameText = new TextField(15);
        mailText = new TextField(15);
        collegeText = new TextField(15);

        errorText = new TextArea(10, 40);
        errorText.setEditable(false);

        Panel first = new Panel();
        first.setLayout(new GridLayout(4, 2));
        first.add(new Label("Sailor ID:"));
        first.add(sidText);
        first.add(new Label("Name:"));
        first.add(snameText);
        first.add(new Label("Mail_id:"));
        first.add(mailText);
        first.add(new Label("College:"));
        first.add(collegeText);

        Panel second = new Panel(new GridLayout(4, 1));
```

```
        second.add(updateStudentButton);

        Panel third = new Panel();
        third.add(errorText);

        add(first);
        add(second);
        add(third);

        setTitle("Update Student");
        setSize(500, 600);
        setLayout(new FlowLayout());
        setVisible(true);

    }

    private void displaySQLErrors(SQLException e)
    {
        errorText.append("\nSQLException: " + e.getMessage() + "\n");
        errorText.append("SQLState:     " + e.getSQLState() + "\n");
        errorText.append("VendorError:  " + e.getErrorCode() + "\n");
    }

    public static void main(String[] args)
    {
        UpdateStudent ups = new UpdateStudent();

        ups.addWindowListener(new WindowAdapter(){
          public void windowClosing(WindowEvent e)
          {
              System.exit(0);
          }
        });

        ups.buildGUI();
    }
}
```




```
SQL> select * from students;

STUDENTID  SNAME      MAILID               COLLEGE
---------- ---------- -------------------- ----------
737-080    maha       maha@gmail.com       vasavi
737-121    Jay        Jay@gmail.com        vasavi
737-122    hari       hari@gmail.com       vasavi
737-123    sharath    sharath@gmail.com    vasavi
737-125    arjun      arjun@gmail.com      vasavi
737-124    aadhi      aadhi@gmail.com      vasavi

6 rows selected.

SQL> select * from students;

STUDENTID  SNAME      MAILID               COLLEGE
---------- ---------- -------------------- ----------
737-121    Jay        Jay@gmail.com        vasavi
737-122    hari       hari@gmail.com       vasavi
737-123    sharath    sharath@gmail.com    vasavi
737-125    arjun      arjun@gmail.com      vasavi
737-124    aadhi      aadhi@gmail.com      vasavi
737-080    mahalaxmi  maha@gmail.com       vasavi

6 rows selected.
```

Delete Student:

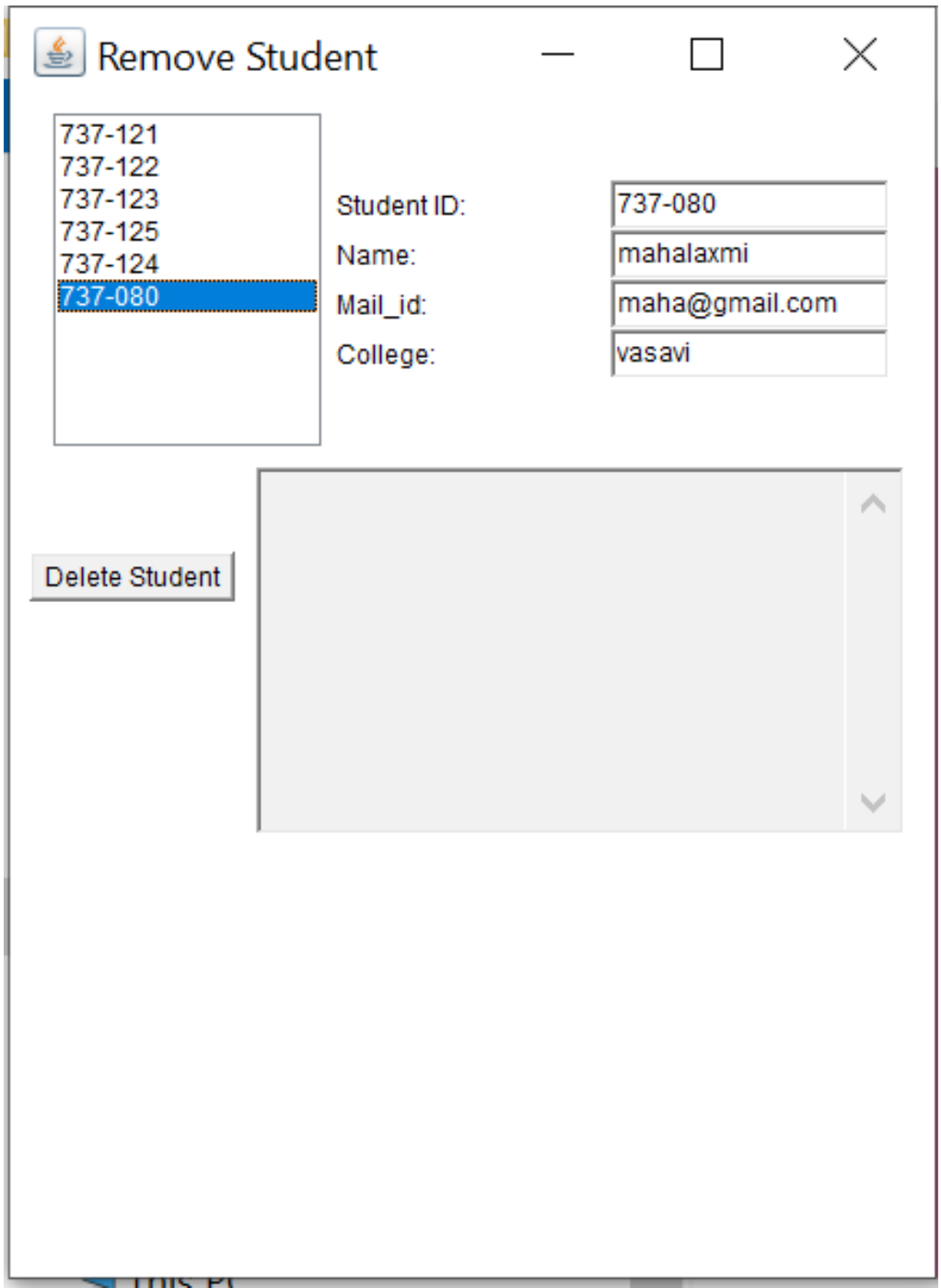
```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class DeleteStudent extends Frame
{

	Button deleteStudentButton;
	List studentIDList;
	TextField sidText, snameText, mailText, collegeText;
	TextArea errorText;
	Connection connection;
	Statement statement;
	ResultSet rs;

	public DeleteStudent()
	{
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
	}

	public void connectToDB()
   {
		try
		{
		  connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:SHARATH","maha","
2000");
		  statement = connection.createStatement();
```

```
            }
            catch (SQLException connectException)
            {
              System.out.println(connectException.getMessage());
              System.out.println(connectException.getSQLState());
              System.out.println(connectException.getErrorCode());
              System.exit(1);
            }
    }

        private void loadStudents()
        {
            try
            {
              rs = statement.executeQuery("SELECT * FROM Students");
              while (rs.next())
              {
                    studentIDList.add(rs.getString("STUDENTID"));
              }
            }
            catch (SQLException e)
            {
              displaySQLErrors(e);
            }
        }

        public void buildGUI()
        {
           studentIDList = new List(10);
            loadStudents();
            add(studentIDList);

            //When a list item is selected populate the text fields
            studentIDList.addItemListener(new ItemListener()
            {
                    public void itemStateChanged(ItemEvent e)
                    {
                            try
                            {
                                    rs = statement.executeQuery("SELECT * FROM
students");
```

```
                                while (rs.next())
                                {
                                        if
(rs.getString("STUDENTID").equals(studentIDList.getSelectedItem()))
                                        break;
                                }
                                if (!rs.isAfterLast())
                                {
                                        sidText.setText(rs.getString("STUDENTID"));
                                        snameText.setText(rs.getString("SNAME"));
                                        mailText.setText(rs.getString("MAILID"));
                                        collegeText.setText(rs.getString("COLLEGE"));
                                }
                        }
                        catch (SQLException selectException)
                        {
                                displaySQLErrors(selectException);
                        }
                }
        });


        //Handle Delete Sailor Button
        deleteStudentButton = new Button("Delete Student");
        deleteStudentButton.addActionListener(new ActionListener()
        {
                public void actionPerformed(ActionEvent e)
                {
                        try
                        {
                                Statement statement = connection.createStatement();
                                int i = statement.executeUpdate("DELETE FROM
students WHERE STUDENTID = '"+studentIDList.getSelectedItem()+"' and
sname='"+snameText.getText()+"' and mailid='"+mailText.getText()+"' and
college='"+collegeText.getText()+"'");
                                errorText.append("\nDeleted " + i + " rows
successfully");
                                sidText.setText(null);
                                snameText.setText(null);
                                mailText.setText(null);
                                collegeText.setText(null);
                                studentIDList.removeAll();
```

```
                        loadStudents();
                }
                catch (SQLException insertException)
                {
                        displaySQLErrors(insertException);
                }
        }
});

sidText = new TextField(15);
snameText = new TextField(15);
mailText = new TextField(15);
collegeText = new TextField(15);

errorText = new TextArea(10, 40);
errorText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Student ID:"));
first.add(sidText);
first.add(new Label("Name:"));
first.add(snameText);
first.add(new Label("Mail_id:"));
first.add(mailText);
first.add(new Label("College:"));
first.add(collegeText);

Panel second = new Panel(new GridLayout(4, 1));
second.add(deleteStudentButton);

Panel third = new Panel();
third.add(errorText);

add(first);
add(second);
add(third);

setTitle("Remove Student");
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
```

```
        }



        private void displaySQLErrors(SQLException e)
        {
                errorText.append("\nSQLException: " + e.getMessage() + "\n");
                errorText.append("SQLState:     " + e.getSQLState() + "\n");
                errorText.append("VendorError:  " + e.getErrorCode() + "\n");
        }

}
```



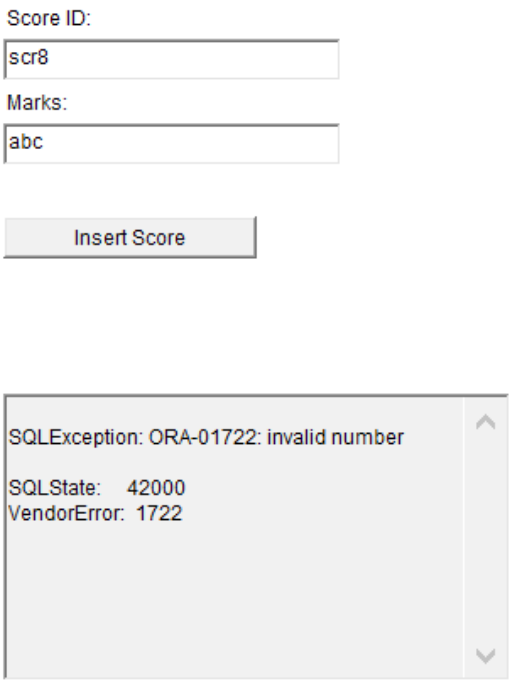
```
SQL> select * from students;

STUDENTID  SNAME      MAILID               COLLEGE
---------- ---------- -------------------- ----------
737-121    Jay        Jay@gmail.com        vasavi
737-122    hari       hari@gmail.com       vasavi
737-123    sharath    sharath@gmail.com    vasavi
737-125    arjun      arjun@gmail.com      vasavi
737-124    aadhi      aadhi@gmail.com      vasavi
737-080    mahalaxmi  maha@gmail.com       vasavi

6 rows selected.

SQL> select * from students;

STUDENTID  SNAME      MAILID               COLLEGE
---------- ---------- -------------------- ----------
737-121    Jay        Jay@gmail.com        vasavi
737-122    hari       hari@gmail.com       vasavi
737-123    sharath    sharath@gmail.com    vasavi
737-125    arjun      arjun@gmail.com      vasavi
737-124    aadhi      aadhi@gmail.com      vasavi
```

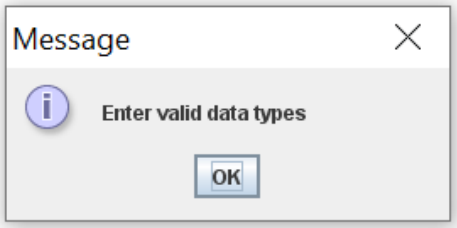
Testing:

The code written for building GUI and connecting with database ensures that the values entered by the user are of correct data types. It prompts an error message if the values entered are not of the specified data types.

Example:

In this example the domain of the marks is number , whereas the user entered characters. So it prompted ann error message.





RESULT:

1. Connection with database is established
2. The values given for tables in the GUI components by the user are saved in the database.

REFERENCES:

https://docs.oracle.com/javase/7/docs/api/
https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/
https://www.javatpoint.com/java-awt