POLYTECHNIC SULTAN MIZAN ZAINAL ABIDIN
DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

DFP50043
INTEGRATIVE PROGRAMMING AND TECHNOLOGY

TOPIC	CHAPTER 2-4
ASSESSMENT	PROBLEM BASED TASK
NAME	MUHAMAD ZULASRAF BIN ZULKIFLI MUHAMMAD FAREEZ DANIEL BIN MOHD KHALID
REG NO	13DDT19F2032 13DDT19F2036
PROGRAMME	DDT5S1

INSTRUCTIONS :
1. Answer ALL the questions
2. Submit the assessment on _____

MARKING SCHEME		
CLO 3P	PLO 8	/20
TOTAL		**/20**

"***THE ENTIRE QUESTION IS BASED ON JTMK'S QUESTION BANK APPROVED BY PROGRAMME LEADER. SIGNATURE IS NOT REQUIRED***."

PROBLEM BASED TASK PRESENTAION

CHAPTER 2 : SWING COMPONENTS
CHAPTER 3 : EVENT HANDLING
CHAPTER 4 : JAVA DATABASE CONNECTIVITY

Learning Outcomes:

By the end of this lab, students should be able to:

- Write program using Event Handling with GUI components.
- Construct Java programs using the swing component
- Arranges JDBC for database connectivity by applying the appropriate steps
- Construct SQL query statements using JDBC

Question:

Your team are required to develop **a management system** using SWING Components with Java Database Connectivity. The system should have login page, menu page, registration, update, delete, manipulating data and retrieve an information. You also should display any necessary report in a suitable format.

You must prepare report Project Report in PDF format

i. Cover Page (include name of group members and matrices number)
ii. Table of contents
iii. Introduction
iv. Coding.
v. Screenshots of interface and description of each.
vi. Conclusion

You must present your product and presentation will be evaluated based on :

a. Accountability (5 marks)
b. Integrity/Honesty (5 marks)
c. Social responsibility (5 marks)
d. Self-Discipline (5 marks)

Table of Contents

1.0 INTRODUCTION

This system is linked to the student information management system; staff members must enter the personal information of students to register data in the system. In addition, staff members can correct data entry errors and erase student information. Finally, the system may review all accessible information and search for the desired name. This system employs the SWING, GUI component, and JDBC database technologies.

2.0 CODING

i) MENUPAGE.java

```
import java.awt.*; //import abstract window toolkit package
import java.awt.event.*; //import event from awt
import javax.swing.*; //import swing
public class menupage extends Frame implements ActionListener
{ Button e = new Button ("Insert");
Button e1 = new Button("Delete");
Button e2 = new Button("Update");
Button e3 = new Button("Search");
Button e4 = new Button("View");
Button e5 = new Button("Close");
public menupage () //constructor
{
add(e); add(e1); add(e2); add(e3); add(e4); add(e5);
e.addActionListener(this); e1.addActionListener(this);
e2.addActionListener(this); e3.addActionListener(this);
e4.addActionListener(this); e5.addActionListener(this);
setLayout(new FlowLayout(FlowLayout.CENTER));
 setTitle("Menu Page");
setSize(350,100);
setVisible(true);
addWindowListener(new WindowEventHandler());
}
class WindowEventHandler extends WindowAdapter
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
```

```
}
}
public void actionPerformed(ActionEvent ae)
{
 Object select=ae.getSource();
 if(select==e)
 {
new form().setVisible(true);
dispose();
 }
 if (select==e1|| select==e2 || select==e3)
 {
 new formUpdate().setVisible(true);
 dispose();
 }
 if (select==e4)
 {
 School s = new School("School Students Details Table");
 dispose();
 }


if (select==e5)
 {
 dispose();
 }
}
public static void main(String arg[])
{
new menupage();
}
}
```

ii) School.java

```
import java.awt.*;
import java.sql.*;
import javax.swing.*;

public class School {

    private boolean status;

    public School(String title) {

            // Creating Window using JFrame
            JFrame frame = new JFrame();
            frame.setTitle(title);
            frame.setSize(800, 500);



            // Adding Table View
            frame.add(getTablePanel());

            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setVisible(true);
    }

    private JPanel getTablePanel() {

            JPanel tableJPanel = new JPanel();
            tableJPanel.setLayout(new BorderLayout());

            // Column Header
            String[] columns = {
```

```
        "Name", "Age","Gender","Phone","Address","DOB","Status" };

        // Getting Data for Table from Database
        Object[][] data = getEmployeeDetails();

        // Creating JTable object passing data and header
        JTable employeeTable = new JTable(data, columns);


        tableJPanel.add(employeeTable.getTableHeader(), BorderLayout.NORTH);
        tableJPanel.add(employeeTable, BorderLayout.CENTER);

        return tableJPanel;
}

private Object[][] getEmployeeDetails() {

        Object[][] data = null;
        Connection conn;
        try {

                // Loading the Driver
                Class.forName("com.mysql.jdbc.Driver");

                // Getting Database Connection Object by Passing URL, Username and Password
                conn=DriverManager.getConnection("jdbc:mysql://localhost:4306/ipt2","root","");
                Statement st=conn.createStatement();

                ResultSet rs = st.executeQuery("Select * from school");

                int rowCount = getRowCount(rs); // Row Count
                int columnCount = getColumnCount(rs); // Column Count

                data = new Object[rowCount][columnCount];

                // Starting from First Row for Iteration
                rs.beforeFirst();
```

```
                int i = 0;
                while (rs.next()) {

                        int j = 0;
                        data[i][j++] = rs.getString("name");
                        data[i][j++] = rs.getString("age");
                        data[i][j++] = rs.getString("gender");
                        data[i][j++] = rs.getString("phone");
                        data[i][j++] = rs.getString("address");
                        data[i][j++] = rs.getString("DOB");
                        data[i][j++] = rs.getString("Department");
                        i++;
                }

                status = true;

                // Closing the Resources;
                st.close();
                conn.close();

        } catch (Exception e) {
                        System.out.println("SQL code does not execute");
        }

        return data;
}

// Method to get Row Count from ResultSet Object
private int getRowCount(ResultSet rs) {

        try {

                if(rs != null) {

                        rs.last();
```

```
                return rs.getRow();
            }

        } catch (SQLException e) {

            System.out.println(e.getMessage());
            e.printStackTrace();
        }

        return 0;
    }

    // Method to get Column Count from ResultSet Object
    private int getColumnCount(ResultSet rs) {

        try {

            if(rs != null)
                return rs.getMetaData().getColumnCount();

        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }

        return 0;
    }

    public static void main(String[] args) {

        String title = "School Students Details Table";
        School School = new School(title);
        System.out.println(School);
    }
}
```

iii) Form.java

```
import java.awt.*; //import abstract window toolkit package
import java.awt.event.*; //import class event from awt package
import java.awt.FlowLayout;
import javax.swing.*;
import java.sql.*; //import required packages
import java.*;
public class form extends Frame implements ActionListener, ItemListener
{
JLabel title, pName, pAge, pGender, pHp, pAdd, pDOB, pStat;
JTextField pName1, pAge1, pHp1, pAdd1, pDOB1;
JRadioButton pGM, pGF, statM, statD, statS;
JButton bInsert, bReset,bSearch;
String getGen, getStat;
//declare variable used in database connection
Connection conn;
Statement st;
ResultSet rs;
String db;
private ButtonGroup group, group1;
 public form()
 {
 setLayout(new FlowLayout());
 title = new JLabel ("Please fill in the Student information" );
 bInsert = new JButton("INSERT");
 bReset = new JButton("RESET");
 bSearch = new JButton("SEARCH");

 //Student's Name
 pName = new JLabel("Name: ");
```

```
pName1 = new JTextField (15);

//student's Age
pAge = new JLabel("Age: ");
pAge1 = new JTextField (15);

//student's Gender
pGender = new JLabel("Gender: ");
pGM = new JRadioButton ("Male");
pGF = new JRadioButton ("Female");

//student's Phone Number
pHp = new JLabel("Phone No: ");
pHp1 = new JTextField (15);

//student's Address
pAdd = new JLabel ("Address: ");
pAdd1 = new JTextField (15);

//student's DOB
pDOB = new JLabel ("Date of Birth: ");
pDOB1 = new JTextField (15);

//student Status
pStat = new JLabel ("Department: ");
statM = new JRadioButton("JKE");
statD = new JRadioButton("JTMK");
statS = new JRadioButton("JKM");

group = new ButtonGroup();
statM.setActionCommand(statM.getText());
statD.setActionCommand(statD.getText());
statS.setActionCommand(statS.getText());
group1 = new ButtonGroup();
pGM.setActionCommand(statM.getText());
pGF.setActionCommand(statD.getText());
add(title);
```

```
 add(pName); add(pName1);
 add(pAge); add(pAge1);
 add(pGender); add(pGM); add(pGF);
 add(pHp); add(pHp1);
 add(pAdd); add(pAdd1);
 add(pDOB); add(pDOB1);
 add(pStat);
 group.add(statD);group.add(statS);
 add(statM); add(statD);
 group1.add(pGM);group1.add(pGF);
 add(statS);group.add(statM);
 add(bInsert); add(bReset);
 add(bSearch);
//button
 bInsert.addActionListener(this);
 bReset.addActionListener(this);
 bSearch.addActionListener(this);
 //gender
 pGM.addItemListener(this);
 pGF.addItemListener(this);
 //status
 statM.addItemListener(this);
 statD.addItemListener(this);
 statS.addItemListener(this);
 setSize(223,500);
 setVisible(true);
}
public static void main(String[] args)
{
form f = new form();
 f.addWindowListener(new WindowEventHandler());
}
public void itemStateChanged(ItemEvent ae)
{
if (pGM.isSelected())
getGen = "Male";
else
```

```
getGen = "Female";
if (statM.isSelected())
getStat = "JKE";
if(statD.isSelected())
getStat = "JTMK";
if(statS.isSelected())
getStat = "JKM";
}
public void actionPerformed(ActionEvent e)
{
 Object select=e.getSource();

 if (select==bInsert)
 {
 try
 {
//Register JDBC drver using mysql
Class.forName("com.mysql.jdbc.Driver");
//Open Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost:4306/ipt2","root","");
Statement st=conn.createStatement();
int age= Integer.parseInt(pAge1.getText());
// Execute Query (insert data using java code)
int i= st.executeUpdate("INSERT into School(name,age,gender,phone,address,DOB, department)values('"+
pName1.getText() + "','" +
pAge1.getText() + "','" + getGen + "','"+ pHp1.getText() + "','" + pAdd1.getText() + "','" +
pDOB1.getText() + "','" + getStat+ "')");
JOptionPane.showMessageDialog(null,"Item Successfully
Added","Confirmation",JOptionPane.INFORMATION_MESSAGE);
setVisible(false);
}
catch(Exception ei)
{
System.out.println("SQL code does not execute");
}
}
if (select==bSearch)
```

```
{
 new formSearch().setVisible(true);
dispose();
 }
 if (select==bReset)
 {
 pName1.setText("");
 pAge1.setText("");
 pGM.setSelected(false);
 pGF.setSelected(false);
 pHp1.setText("");
 pAdd1.setText("");
 pDOB1.setText("");
 statM.setSelected(false);
 statD.setSelected(false);
 statS.setSelected(false);
 JOptionPane.showMessageDialog(null,"The Form Has Already Been Reset");
 }
 }
}
class WindowEventHandler extends WindowAdapter
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
}
```

iv) formUpdate.java

```
import java.awt.*; // import abstract window toolkit package
import java.awt.event.*; // import class event from awt package
import javax.swing.*;
import java.awt.FlowLayout;
// STEP 1.1 : import Required packages
import java.sql.*;

public class formUpdate extends Frame implements ActionListener
{
        JLabel tajuk, tajuk2,Lname, Lage,Lgender, Lphone, Laddress, LDOB,Ldepartment,Lsearch;
        JTextField Tname, Tage,Tgender, Tphone, Taddress,Tsearch,TDOB,Tdepartment;
        JButton Bupdate,Bsearch,Bdelete;

        // STEP 1.2 : Declare variable yang digunakan bagi connection database
                Connection conn;
                PreparedStatement pst;
                ResultSet rs;
                ResultSet rs1;
                ResultSet rs2;
   public formUpdate ()
   {
      setLayout(new FlowLayout( ));
                Lsearch = new JLabel (" Insert Name : " );
                Tsearch = new JTextField (15);
        Bsearch=new JButton("SEARCH");
                add(Lsearch);add(Tsearch);add(Bsearch);
```

```
        tajuk = new JLabel ("                Welcome To School                " );
            Bupdate=new JButton("UPDATE");
            Bdelete=new JButton("DELETE");
        Lname = new JLabel("        Name : ");
        Tname = new JTextField (20);
        Lage = new JLabel("        Age : ");
            Tage = new JTextField (20);
            Lgender = new JLabel("Gender : ");
            Tgender = new JTextField (20);
        Lphone= new JLabel("Phone No : ");
            Tphone = new JTextField (20);
                    Laddress = new JLabel("Address : ");
            Taddress = new JTextField (20);
            LDOB = new JLabel("Date Of Birth : ");
            TDOB = new JTextField (20);
            Ldepartment = new JLabel("Department : ");
            Tdepartment = new JTextField (20);
                    add(tajuk);add(Lname);add(Tname);add(Lage);
            add(Tage);add(Lphone);add(Tphone);
            add(Laddress);add(Taddress);add(LDOB);add(TDOB);
            add(Ldepartment);add(Tdepartment);
            add(Lgender);add(Tgender);
            add(Bupdate);add(Bdelete);
            Bsearch.addActionListener(this);
            Bupdate.addActionListener(this);
            Bdelete.addActionListener(this);
                    setSize(300,500);setVisible(true);
            addWindowListener(new WindowEventHandler());
}
            public static void main(String[] args)
            {
                    formUpdate  f = new formUpdate  ();
            }


            public void actionPerformed(ActionEvent e)
            {
```

```
                    Object pilihan=e.getSource();
    if (pilihan==Bsearch)
    {
        try{
                        //STEP 2 : Register JDBC  drver using mysql
                            Class.forName("com.mysql.jdbc.Driver");
                        //STEP 3 : Open Connection

        conn=DriverManager.getConnection("jdbc:mysql://localhost:4306/ipt2","root","");
                            Statement st=conn.createStatement();
                        // STEP 4.1 : Execute Query (insert data using java code)
                            rs=st.executeQuery("SELECT * FROM school where
name='"+Tsearch.getText()+"'");
                        if(rs.next())
                        {
                                Tname.setText(rs.getString("name"));
                                Tage.setEditable(false);
                        Tage.setText(rs.getString("age"));
                        Tage.setEditable(false);
                        Tgender.setText(rs.getString("gender"));
                        Tgender.setEditable(false);
                        Tphone.setText(rs.getString("phone"));
                        Taddress.setText(rs.getString("address"));
                        TDOB.setText(rs.getString("DOB"));
                        Tdepartment.setText(rs.getString("department"));
                        Tdepartment.setEditable(false);

                        JOptionPane.showMessageDialog(null,"Item Successfully
Retrieve","Confirmation",JOptionPane.INFORMATION_MESSAGE);;}
                        else
                            {       JOptionPane.showMessageDialog(null,"Data Not Found");}
        }
                        catch(Exception ei)
                        {
                            System.out.println("SQL code does not execute");
                        }
```

```
    }

    else if (pilihan==Bdelete)
    {
        try{
                        //STEP 2 : Register JDBC  drver using mysql
                                Class.forName("com.mysql.jdbc.Driver");
                        //STEP 3 : Open Connection

        conn=DriverManager.getConnection("jdbc:mysql://localhost:4306/ipt2","root","");
                                Statement st=conn.createStatement();
                        // STEP 4.1 : Execute Query (insert data using java code)
                                int rs2=st.executeUpdate("DELETE from school where
name='"+Tname.getText()+"'");
                                JOptionPane.showMessageDialog(null,"DataSuccessfully
Deleted","Confirmation",JOptionPane.INFORMATION_MESSAGE);

        }
                        catch(Exception ei)
                        {
                                System.out.println("SQL code does not execute");
                        }
    }

    else if (pilihan==Bupdate)
    {
        try{
                        //STEP 2 : Register JDBC  drver using mysql
                                Class.forName("com.mysql.jdbc.Driver");
                        //STEP 3 : Open Connection

        conn=DriverManager.getConnection("jdbc:mysql://localhost:4306/ipt2","root","");
                                Statement st=conn.createStatement();
                        // STEP 4.1 : Execute Query (insert data using java code)
                                int rs2=st.executeUpdate("UPDATE school set phone='"+Tphone.getText()+"',
DOB='"+TDOB.getText()+"', address='"+Taddress.getText()+"'where name='"+Tname.getText()+"'");
```

```
                    JOptionPane.showMessageDialog(null,"DataSuccessfully
Update","Confirmation",JOptionPane.INFORMATION_MESSAGE);

        }
                catch(Exception ei)
                {
                        System.out.println("SQL code does not execute");
                }
    }


        }

}
```

3) INTERFACE AND DESCRIPTION

3.1 Insert Button

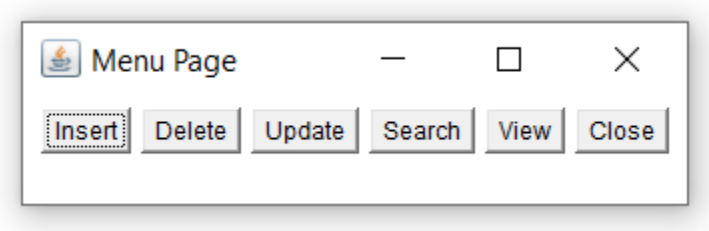

Figure 1: menu page for Student Information

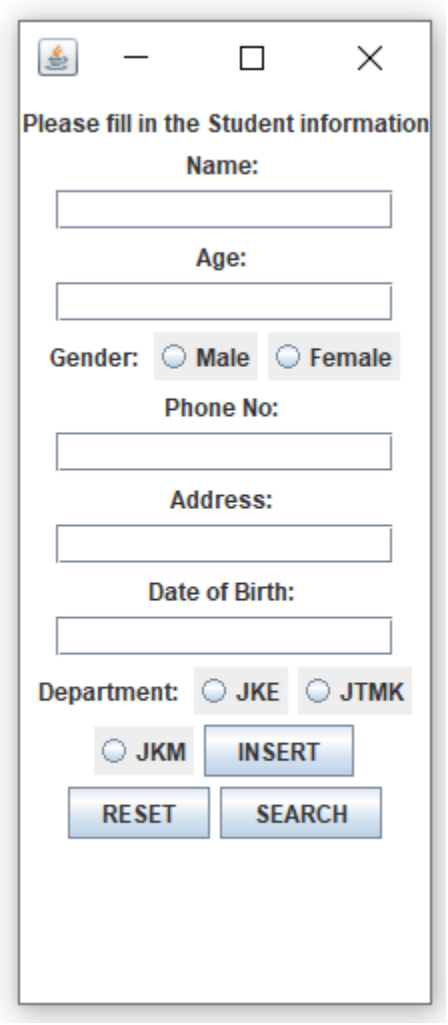

Figure 2: input form for student information

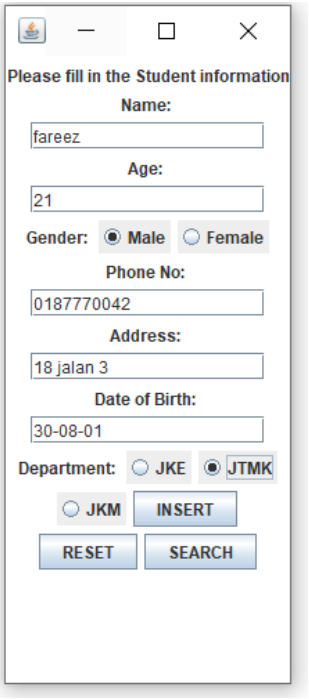

Figure 3: data to insert into the system

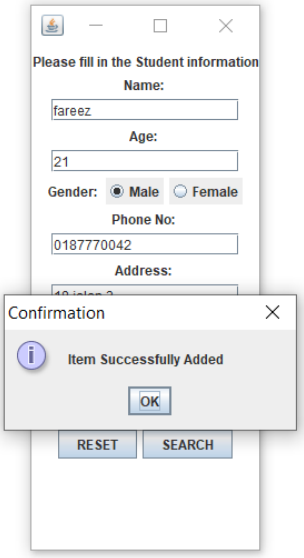

Figure 4: after click insert

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

name	age	gender	phone	address	DOB	department
zul	21	Male	0198112665	19 Tgp Jerasntut	17-04-2021	JKM
fareez	21	Male	0187770042	18 jalan 3	30-08-01	JTMK

Show all | Number of rows: 25 Filter rows: Search this table

Figure 5: the data successfully retrieve into database

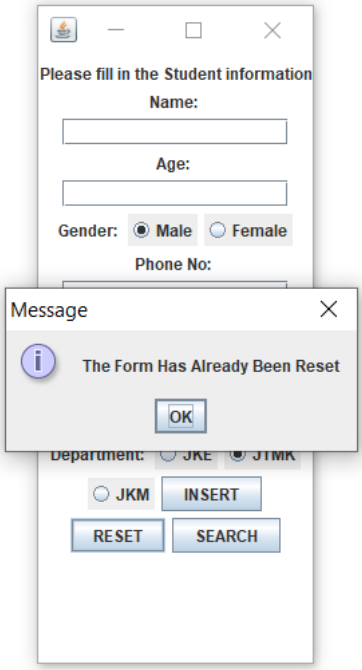

Figure 6: after click reset button the input form will be reset

3.2 Delete, Update and Search Button

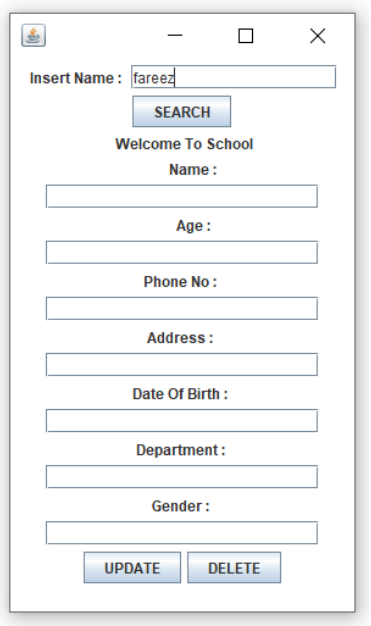

Figure 7: insert name to search

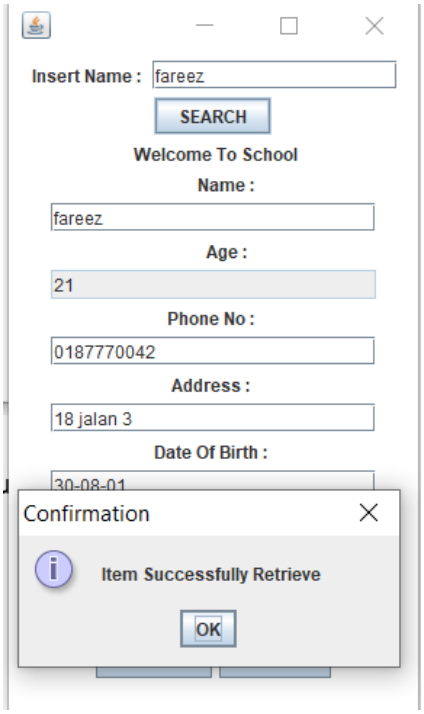

Figure 8: data will be retrieve

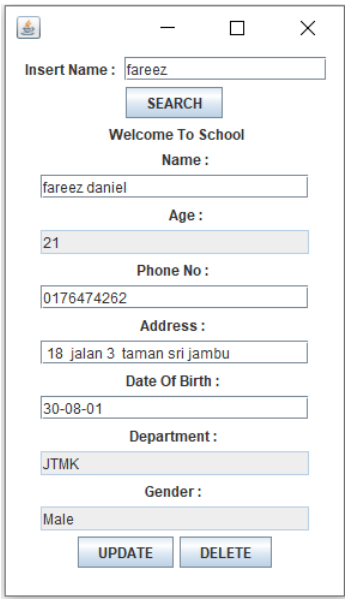

Figure 9 :put new information to update data

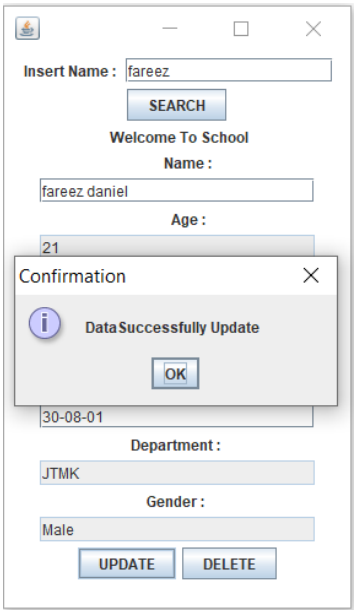

Figure 10: message box will popup and say data successfully update

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

name	age	gender	phone	address	DOB	department
fareez daniel	21	Male	0176474262	18 jalan 3 taman sri jambu	30-08-01	JTMK

Show all | Number of rows: 25 Filter rows: Search this table

Figure 11: the database will show new information after the update on GUI

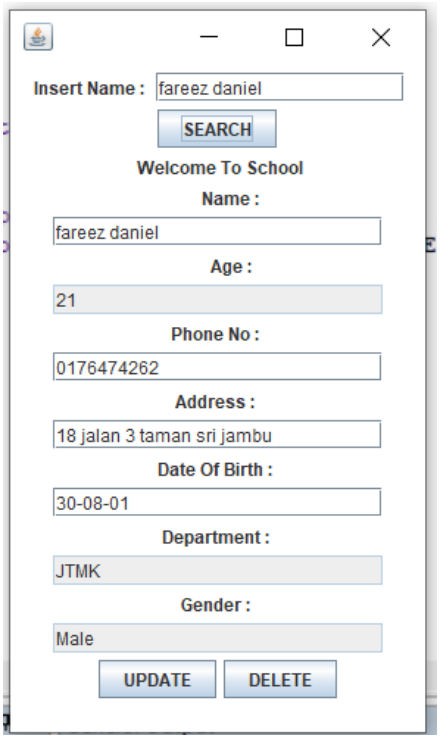

Figure 12: next is, delete button

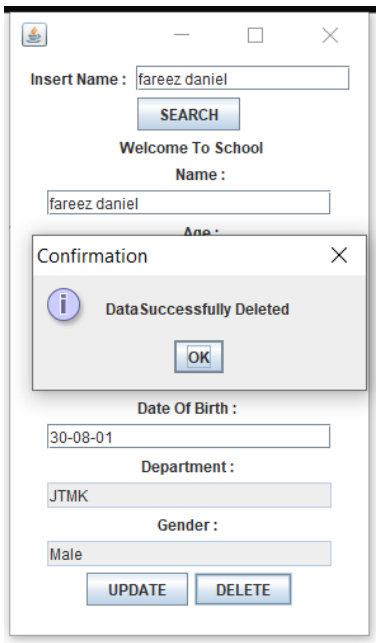

Figure 13: after click the delete button the message will popup

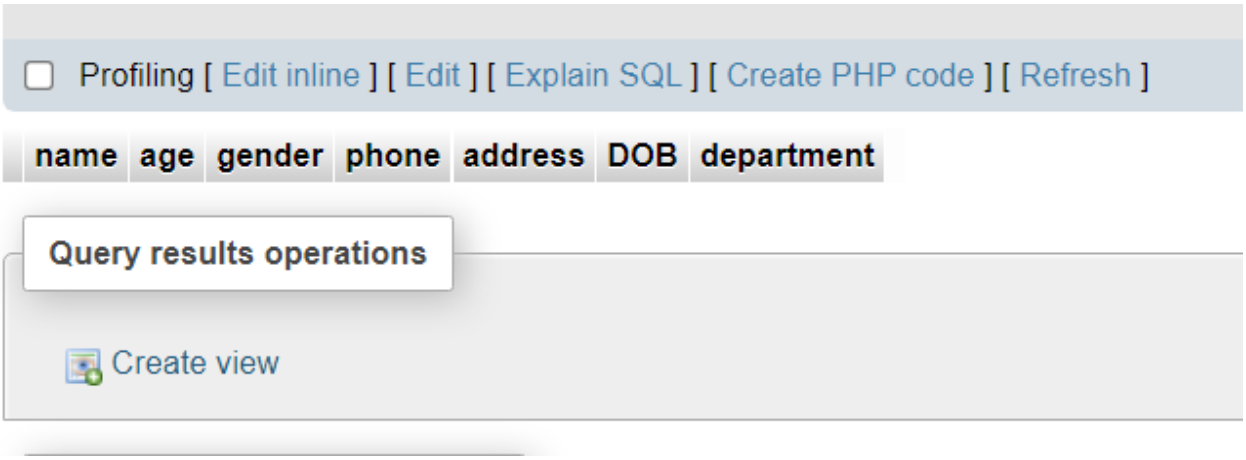

Figure 14: the user name fareez daniel will be deleted

3.3 View Button

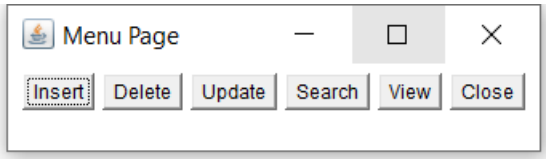

Figure 15: go to menupage and click view

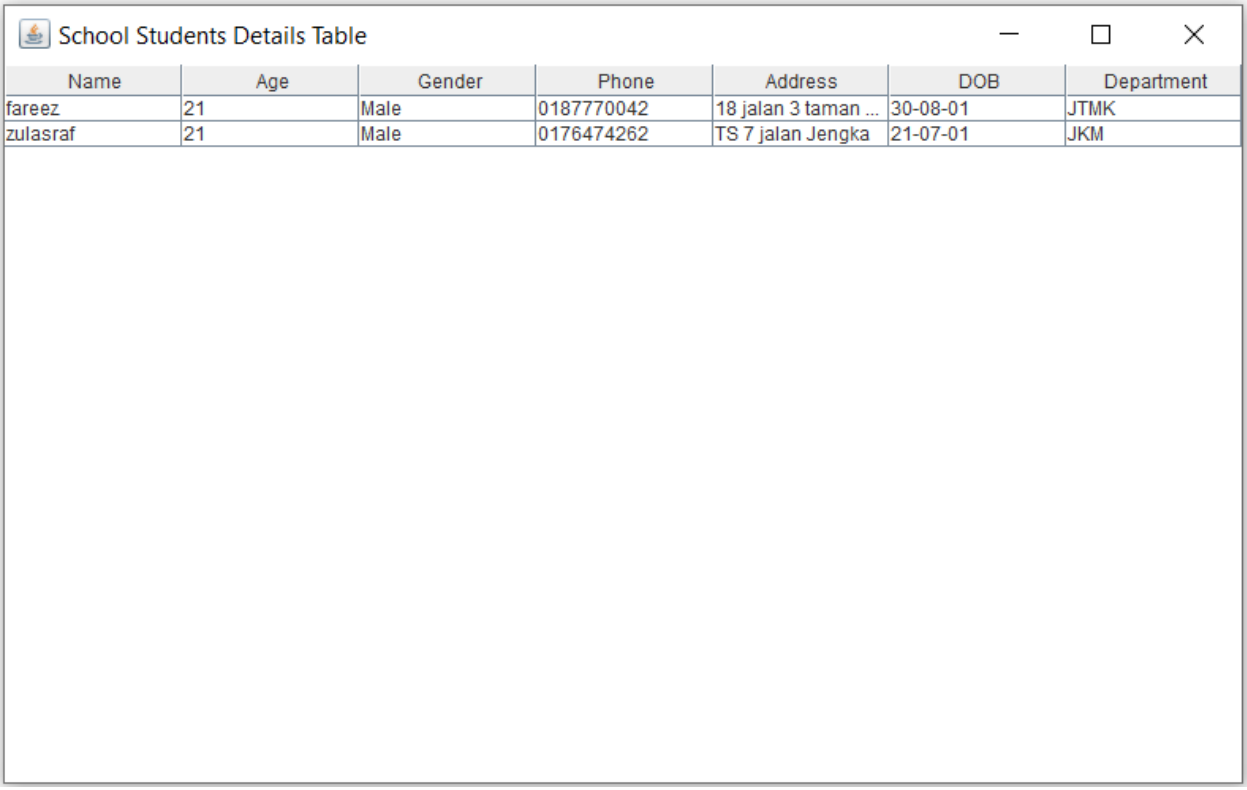

Figure 16: the data will show same as database

Figure 17 : the View is retrieve from database

3.4 Close button

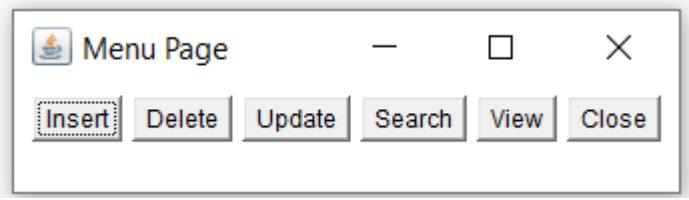

Figure 18: click the close button and the gui will exit

4.0) Conclusion

In conclusion, compared to the manual technique, which required writing student data on paper using a written system, this student information system will make it simpler for employees to enter student data. In contrast, the manual method needed employees to use a written system. We anticipate that the mechanism for entering, and processing data will be sped up with the implementation of this student information system, and that previously collected data will be brought up to date. If we continue to use manual processes like this one, data entry will be quite slow, and it will take a significant amount of time for each individual to enter their personal data.

END

CRITERIA	(Excellent) 4	(Good) 3	(Fair) 2	(Poor) 1	Mark	Weightage (%)	Score
User Interface a. Content b. Purpose c. Theme	The interface has a very clear content, purpose and theme throughout the interface.	The interface has a clear content, purpose and theme throughout interface.	The interface has fewer clear content, purposes and themes throughout the interface.	The interface has a content, purpose and theme that is not clear throughout the interface.		10	score/4* 10
Dialog	Use all the dialog box for confirmation from user	Use most of the dialog box for confirmation user	Use partially the dialog box for confirmation from user	Did not user dialog box for confirmation from user		5	score/4* 5
Menu	Use an at least 6 an appropriate Menu and Menu Item	Use only 3 to 4 an appropriate Menu and Menu Item	Use only 2 an appropriate Menu and Menu Item	Did not use Menu and Menu Item		5	score/4* 5
Event Handling	Use at least 4 Event classes and 4 Event listeners with GUI components. All Events are SUCCESSFUL function.	Use 3 to 4 Event classes and 3 to 4 Event listeners with GUI components. PARTS OF Events are UNSUCCESSFUL function.	Use 3 and below of Event classes and 3 and below Event listeners with GUI components. ALL Events are SUCCESSFUL function.	Use 3 and below of Event classes and 3 and below Event listeners with GUI components. ALL Events are UNSUCCESSFUL function.		5	score/4* 5
connection to database	SUCCESSFULLY connect GUI with database (data can be updated)	Successfully connect GUI with database but with warnings(data can't be updated)	Has a database include table but UNSUCCESSFUL connection.	Has a database only and UNSUCCESSFUL connection.		5	score/4* 5

SQL statement	Able to do ALL SQL statement and SUCCESSFULLY operate.	Able to do 3 SQL statements and SUCCESSFULLY operate.	Able to do 2 or 1 SQL statements and SUCCESSFULLY operate.	Able to do SQL statement but NOT FULFILL operation requirement.		10	score/4* 10
Execution	Executes without errors and the program produces a correct output	Executes without errors and the program produces an incorrect output	Executes with some errors and the program not produces a output	Does not execute due to errors and the program not produces a output		5	score/4* 5
Output	Student can answer and delivering the output before the time	Student can answer and delivering the output on time	Student can answer and delivering the output after the due date	Student cannot answer and delivering the output after the due date		5	score/4* 5
TOTAL						50	

RUBRIC PRESENTATION

Generic Student Attributes (GSA) / Learning Domain (LD)	Skills / Aspects	Excellent	Good	Unsatisfactory	Mark	Weightage (%)	Score
		3	2	1			
CLS5 : Ethics & Professionalism (PLO8)	**A. Accountability**: Able to adapt and implement the concept of accountability in carrying out the task towards achieving related goals.	Able to focus on tasks that need to be completed before the deadlines. Able to produce high quality work according to proper procedures.	Able to focus on tasks that need to be completed on time and be able to produce work according to proper procedures.	Unable to focus on tasks that need to be completed on time and unable to produce work according to proper procedures.		5	score/3* 5
	B. Integrity/Honesty: Trustworthy; display high standard of ethical conduct and understand the impact of violating integrity in oneself, others and an organization.	Fully demonstrate a pattern of professional behavior such as promptness, task completion and academic integrity.	Demonstrate a partial pattern of professional behavior such as promptness, task completion and academic integrity.	Demonstrate a pattern of unprofessional behavior such as absence, tardiness, failure to complete tasks, inappropriately dressed or inappropriate personal behavior, violation of confidentiality, violation of academic integrity (e.g. plagiarism, cheating and etc).		5	score/3* 5
	C. Social responsibility: An obligation to act for the benefit of society at large.	Fully aware of the relation between individuals and society; demonstrate concern and active involvement in society.	Show some awareness of the relation between individuals and society.	Lack awareness of an individual's relation to society.		5	score/3* 5

	D. Self Discipline: Willing to obey all orders, respect authorities, involve in teamwork; self-reliance.	Display excellent effort and commitment in performing tasks, such as attendance, punctuality, enthusiasm, vitality, and optimism in performing and completing tasks.	Display effort and commitment in performing tasks, such as attendance, punctuality, enthusiasm, vitality, and optimism in performing and completing tasks.	Display minimal effort and commitment in performing tasks, such as attendance, punctuality, enthusiasm, vitality, and optimism in performing and completing tasks.		5	score/3* 5
TOTAL							