UMS
UNIVERSITI MALAYSIA SABAH

KP14603 OBJECT ORIENTED PROGRAMMING
SEMESTER 2
SESSION 2019/2020

ASSIGNMENT 2

Secondary Student Performance Management System

LECTURE : MADAM SITI HASNAH TANALOL

NAME:NURHIDAYAH BINTI AMBOMAI
NO MATRICS:BI19110070

CONTENT

1.0 Introduction

For this project,i choose to create an Secondary Student performance system.This system is a great approach for teacher.It help to manage students records and keep a track of student result .Permission also provided for the teacher to insert student information,student marks,and update student co-curricular activities.This will make things easy because the teacher can update the student informatioon the system.This system also develop to gather what all records need to be stored regarding a student.

All the details of the student are added by teacher.In this system there is no chance of losing data because the system maintain a database to store all the information.Adding information by teacher is very easy which does not take much time and physical effort.

In this system teacher can login by entering username .This system only allowed teacher in the school to access the system which make things easy for teacher to update their student information.

2.0 Java code

Teacher_login.java

```
 */
public class Teacher_login extends javax.swing.JFrame {

    /**
     * Creates new form Teacher_login
     */
    public Teacher_login() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void Teacher_NameActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Secondary_student_performance s=new Secondary_student_performance();

        s.setVisible(true);
    }

    private void pnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        Teacher_Name.setText("");

        Teacher_Name.setText("");

        pn.setText("");
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Teacher_login().setVisible(true);
            }
        });
    }
```

Secondary_student_performance.java

```
import java.io.FileWriter;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Secondary_student_performance extends javax.swing.JFrame {

    /**
     * Creates new form Secondary_student_performance
     */
    public Secondary_student_performance() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

```
    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Student_Information s=new Student_Information();

        s.setVisible(true);
    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        Student_Name.setText("");
        ID.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField6.setText("");
        jTextField5.setText("");
        FORM.setSelectedItem(0);
        Class.setSelectedItem(0);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

        model.addRow(new Object[]{
            Student_Name.getText(),ID.getText(),jTextField4.getText(),FORM.getSelectedItem().toString(),Class.getSelectedItem().toString()
        });
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
        if(jTable1.getSelectedRow()==-1){
            if(jTable1.getSelectedRow()==0){
                JOptionPane.showMessageDialog(null,"Table is empty");
            }else{
```

```
            JOptionPane.showMessageDialog(null,"You must select which row to be updated");
        }
    }

    model.setValueAt(Student_Name.getText(), jTable1.getSelectedRow(), 0);
    model.setValueAt(ID.getText(), jTable1.getSelectedRow(), 1);

    model.setValueAt(jTextField4.getText(), jTable1.getSelectedRow(), 1);

    model.setValueAt(FORM.getSelectedItem().toString(), jTable1.getSelectedRow(), 2);
    model.setValueAt(Class.getSelectedItem().toString(), jTable1.getSelectedRow(), 3);

}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

    if(jTable1.getSelectedRow()==-1){
        if(jTable1.getSelectedRow()==0){
            JOptionPane.showMessageDialog(null,"Table is empty");
        }else{
            JOptionPane.showMessageDialog(null,"You must select which row to be deleted");
        }
    }else{
        model.removeRow(jTable1.getSelectedRow());
    }

}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
   DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
   Student_Name.setText(model.getValueAt(jTable1.getSelectedRow(), 0).toString());
   ID.setText(model.getValueAt(jTable1.getSelectedRow(), 1).toString());
   jTextField4.setText(model.getValueAt(jTable1.getSelectedRow(), 2).toString());
   FORM.setSelectedItem(model.getValueAt(jTable1.getSelectedRow(), 3).toString());
   Class.setSelectedItem(model.getValueAt(jTable1.getSelectedRow(), 4).toString());
```

```
private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
    String name=Student_Name.getText().toString();
    String id=ID.getText().toString();
    String pn=jTextField4.getText().toString();
    String fm=FORM.getSelectedItem().toString();
    String cl=Class.getSelectedItem().toString();

    try{
        FileWriter writer = new FileWriter("student Information.txt",true);

        writer.write(name);
        writer.write(" ");
        writer.write(id);
        writer.write(" ");
        writer.write(pn);
        writer.write(" ");
        writer.write(fm);
        writer.write(" ");
        writer.write(cl);
        writer.write(System.getProperty("line.separator"));

        writer.close();

        JOptionPane.showMessageDialog(rootPane,"Success");

    }catch(Exception e){
        JOptionPane.showMessageDialog(rootPane,"Error");
    }
}
```

Student_Information.java

```
import java.io.FileWriter;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Student_Information extends javax.swing.JFrame {

    /**
     * Creates new form Student_Information
     */
    public Student_Information() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        int eng =Integer.parseInt(jTextField8.getText());
        int bm =Integer.parseInt(jTextField9.getText());
        int sej =Integer.parseInt(jTextField10.getText());
        int geo =Integer.parseInt(jTextField11.getText());
        int math =Integer.parseInt(jTextField12.getText());

        int tot=eng+bm+sej+geo+math;

        jTextField4.setText("" + tot);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        int tot=Integer.parseInt(jTextField4.getText());
        int avg=tot/5;

        jTextField5.setText("" + avg);
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        double avg=Double.parseDouble(jTextField5.getText());

        char Grade;

        if(avg>=90)
        Grade='A';
        else
        if(avg>=70)
        Grade='B';
        else
        if(avg>=50)
        Grade='C';
```

```
    if(avg>=50)
    Grade='C';
    else
    if(avg>=33)
    Grade='D';
    else
    Grade='F';

    jTextField6.setText(""+Grade);
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField13.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");
    jTextField11.setText("");
    jTextField12.setText("");
    jTextField4.setText("");
    jTextField6.setText("");
    jTextField5.setText("");
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {



    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

    model.addRow(new Object[]{
        jTextField13.getText(),jTextField4.getText(),jTextField5.getText(),jTextField6.getText()
    });
}

  private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
      DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

      if(jTable1.getSelectedRow()==-1){
          if(jTable1.getSelectedRow()==0){
              JOptionPane.showMessageDialog(null,"Table is empty");
          }else{
              JOptionPane.showMessageDialog(null,"You must select which row to be updated");
          }
      }
      model.setValueAt(jTextField13.getText(), jTable1.getSelectedRow(), 0);

      model.setValueAt(jTextField4.getText(), jTable1.getSelectedRow(), 1);
      model.setValueAt(jTextField5.getText(), jTable1.getSelectedRow(), 2);
      model.setValueAt(jTextField6.getText(), jTable1.getSelectedRow(), 3);
  }

  private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
     DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
      if(jTable1.getSelectedRow()==-1){
          if(jTable1.getSelectedRow()==0){
              JOptionPane.showMessageDialog(null,"Table is empty");
          }else{
              JOptionPane.showMessageDialog(null,"You must select which row to be deleted");
          }
      }else{
          model.removeRow(jTable1.getSelectedRow());
      }
  }

  private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
      String name=jTextField13.getText().toString();
      String total=jTextField4.getText().toString();
      String grade=jTextField6.getText().toString();
      String avg=jTextField5.getText().toString();
```

```
        try{
            FileWriter writer = new FileWriter("student List.txt",true);

            writer.write(name);
            writer.write(" ");
            writer.write(total);
            writer.write(" ");
            writer.write(grade);
            writer.write(" ");
            writer.write(avg);

            writer.write(System.getProperty("line.separator"));

            writer.close();

            JOptionPane.showMessageDialog(rootPane,"Success");

        }catch(Exception e){
            JOptionPane.showMessageDialog(rootPane,"Error");
        }
    }

    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Cocuriculum_update s=new Cocuriculum_update();

        s.setVisible(true);
    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
       DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
       jTextField13.setText(model.getValueAt(jTable1.getSelectedRow(), 0).toString());
       jTextField4.setText(model.getValueAt(jTable1.getSelectedRow(), 1).toString());
       jTextField6.setText(model.getValueAt(jTable1.getSelectedRow(), 2).toString());
       jTextField5.setText(model.getValueAt(jTable1.getSelectedRow(), 3).toString());
```

Cocurriculum_update.java

```
import java.io.FileWriter;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author User
 */
public class Cocuriculum_update extends javax.swing.JFrame {

    /**
     * Creates new form Cocuriculum_update
     */
    public Cocuriculum_update() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
```

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    Student_Name.setText("");
    Ass.setSelectedItem(0);
    sp.setSelectedItem(0);
    cl.setSelectedItem(0);
    m.setText("");
}

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

    model.addRow(new Object[]{
        Student_Name.getText(),Ass.getSelectedItem().toString(),sp.getSelectedItem().toString(),cl.getSelectedItem().toString(),m
        .getText()
    });
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
    if(jTable1.getSelectedRow()==-1){
        if(jTable1.getSelectedRow()==0){
            JOptionPane.showMessageDialog(null,"Table is empty");
        }else{
            JOptionPane.showMessageDialog(null,"You must select which row to be deleted");
        }
    }else{
        model.removeRow(jTable1.getSelectedRow());
    }
```

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
    if(jTable1.getSelectedRow()==-1){
        if(jTable1.getSelectedRow()==0){
            JOptionPane.showMessageDialog(null,"Table is empty");
        }else{
            JOptionPane.showMessageDialog(null,"You must select which row to be updated");
        }
    }
    model.setValueAt(Student_Name.getText(), jTable1.getSelectedRow(), 0);
    model.setValueAt(Ass.getSelectedItem().toString(), jTable1.getSelectedRow(), 1);
    model.setValueAt(sp.getSelectedItem().toString(), jTable1.getSelectedRow(), 2);
    model.setValueAt(cl.getSelectedItem().toString(), jTable1.getSelectedRow(), 3);
    model.setValueAt(m.getText(), jTable1.getSelectedRow(), 4);



}

private void AssActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
    String name=Student_Name.getText().toString();
    String ass=Ass.getSelectedItem().toString();
    String sport=sp.getSelectedItem().toString();;
    String marks=m.getText().toString();

    try{
        FileWriter writer = new FileWriter("Student Co-curricular activities.txt",true);

        writer.write(name);
        writer.write(" ");
```

```
            writer.write(name);
            writer.write(" ");
            writer.write(ass);
            writer.write(" ");
            writer.write(sport);
            writer.write(" ");
            writer.write(marks);

            writer.write(System.getProperty("line.separator"));

            writer.close();

            JOptionPane.showMessageDialog(rootPane,"Success");

        }catch(Exception e){
            JOptionPane.showMessageDialog(rootPane,"Error");
        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Cocuriculum_update().setVisible(true);
            }
        });
    }
```

3.0 Object oriented concept implementation

a)Encapsulation.

By providing only a setter or getter method, make the class read-only or write-only.

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    Student_Name.setText("");
    ID.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField6.setText("");
    jTextField5.setText("");
    FORN.setSelectedItem(0);
    Class.setSelectedItem(0);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

    model.addRow(new Object[]{
        Student_Name.getText(),ID.getText(),jTextField4.getText(),FORN.getSelectedItem().toString(),Class.getSelectedItem().toString()
    });
}
```

b)Inheritance

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.**extends** is the keyword used to inherit the properties of a class.

```
 */
public class Secondary_student_performance extends javax.swing.JFrame {

    /**
     * Creates new form Secondary_student_performance
     */
    public Secondary_student_performance() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
```

c)Dynamic Polymorphism

Dynamic Polymorphism is the mechanism by which multiple methods can be defined with same name and signature in the superclass and subclass. The call to an overridden method are resolved at run time.

```
this.setVisible(false);
Student_Information s=new Student_Information();

s.setVisible(true);
```

```
DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
if(jTable1.getSelectedRow()==-1){
    if(jTable1.getSelectedRow()==0){
        JOptionPane.showMessageDialog(null,"Table is empty");
    }else{
        JOptionPane.showMessageDialog(null,"You must select which row to be updated");
    }
}

model.setValueAt(Student_Name.getText(), jTable1.getSelectedRow(), 0);
model.setValueAt(ID.getText(), jTable1.getSelectedRow(), 1);

model.setValueAt(jTextField4.getText(), jTable1.getSelectedRow(), 1);

model.setValueAt(FORM.getSelectedItem().toString(), jTable1.getSelectedRow(), 2);
model.setValueAt(Class.getSelectedItem().toString(), jTable1.getSelectedRow(), 3);
```

d)Class and object

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.Ways to create object of a class is using new keyboard.

```
private void Teacher_NameActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Secondary_student_performance s=new Secondary_student_performance();

    s.setVisible(true);
```

e)Method

Non static methods in Java are used more than static methods. Those methods can only be run on objects and not on the whole class.

Non static methods can access and alter the field of the object.

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    Student_Name.setText("");
    Ass.setSelectedItem(0);
    sp.setSelectedItem(0);
    cl.setSelectedItem(0);
    m.setText("");
}

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

    model.addRow(new Object[]{
        Student_Name.getText(),Ass.getSelectedItem().toString(),sp.getSelectedItem().toString(),cl.getSelectedItem().toString(),m
        .getText()
    });
}
```

4.0 Read and write implimentation

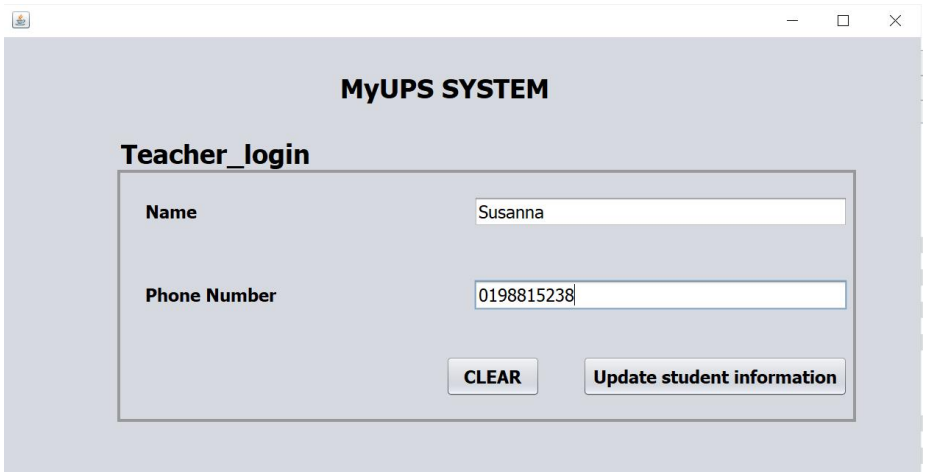
Read and read implementation that I used is JTextField.JTextField class is used to create a textfield control, which allows a user to enter a single line text and edit it.

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        int eng =Integer.parseInt(jTextField8.getText());
        int bm =Integer.parseInt(jTextField9.getText());
        int sej =Integer.parseInt(jTextField10.getText());
        int geo =Integer.parseInt(jTextField11.getText());
        int math =Integer.parseInt(jTextField12.getText());

        int tot=eng+bm+sej+geo+math;

        jTextField4.setText("" + tot);
    }
```

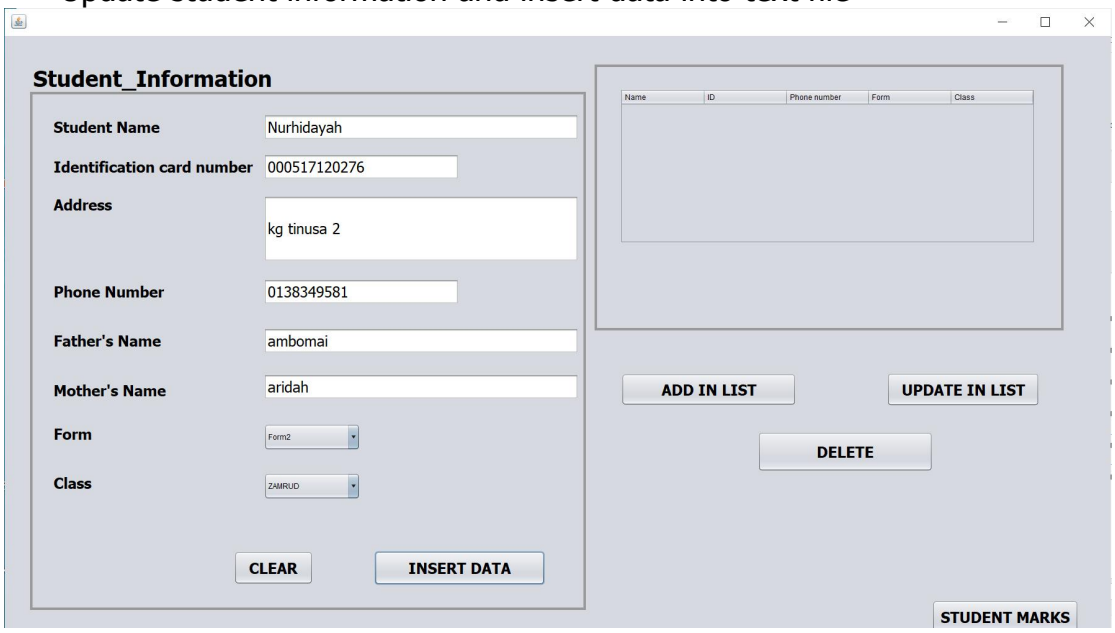
```
private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
    String name=jTextField13.getText().toString();
    String total=jTextField4.getText().toString();
    String grade=jTextField6.getText().toString();
    String avg=jTextField5.getText().toString();

    try{
        FileWriter writer = new FileWriter("student List.txt",true);

        writer.write(name);
        writer.write(" ");
        writer.write(total);
        writer.write(" ");
        writer.write(grade);
        writer.write(" ");
        writer.write(avg);

        writer.write(System.getProperty("line.separator"));

        writer.close();

        JOptionPane.showMessageDialog(rootPane,"Success");

    }catch(Exception e){
        JOptionPane.showMessageDialog(rootPane,"Error");
    }
}
```

5.0 User manual

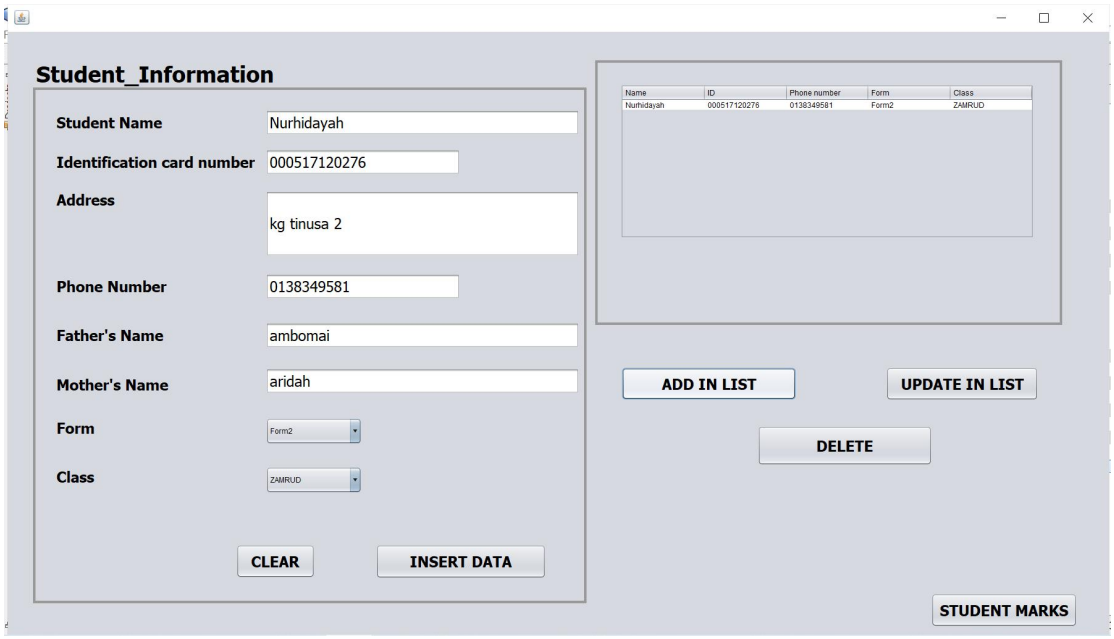
- Write teacher name and phone number.Then update student information



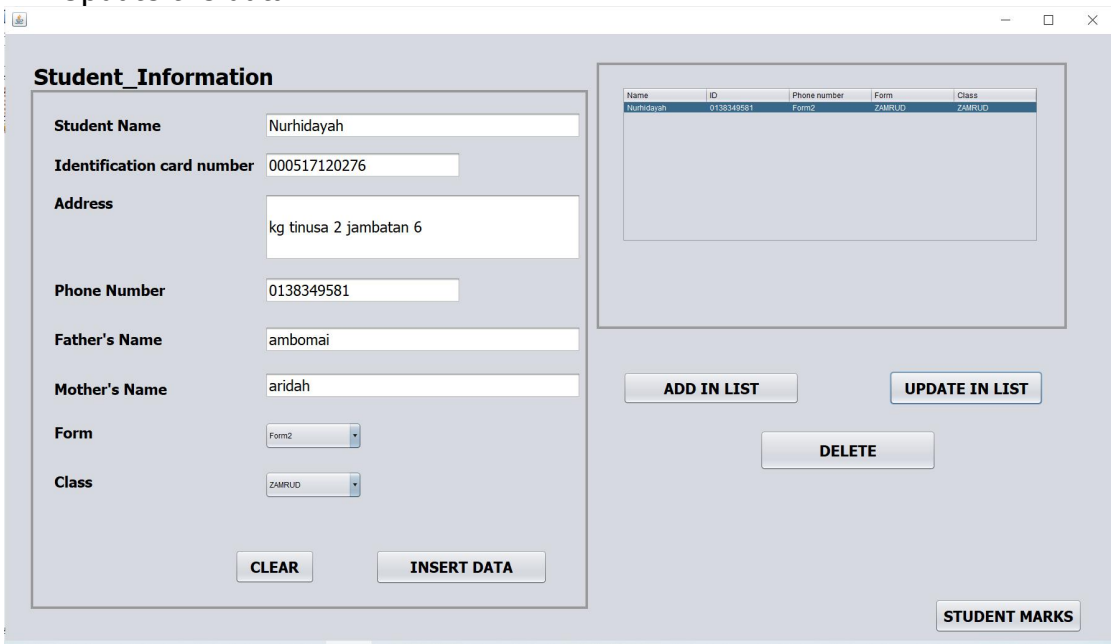
- Update student information and insert data into text file



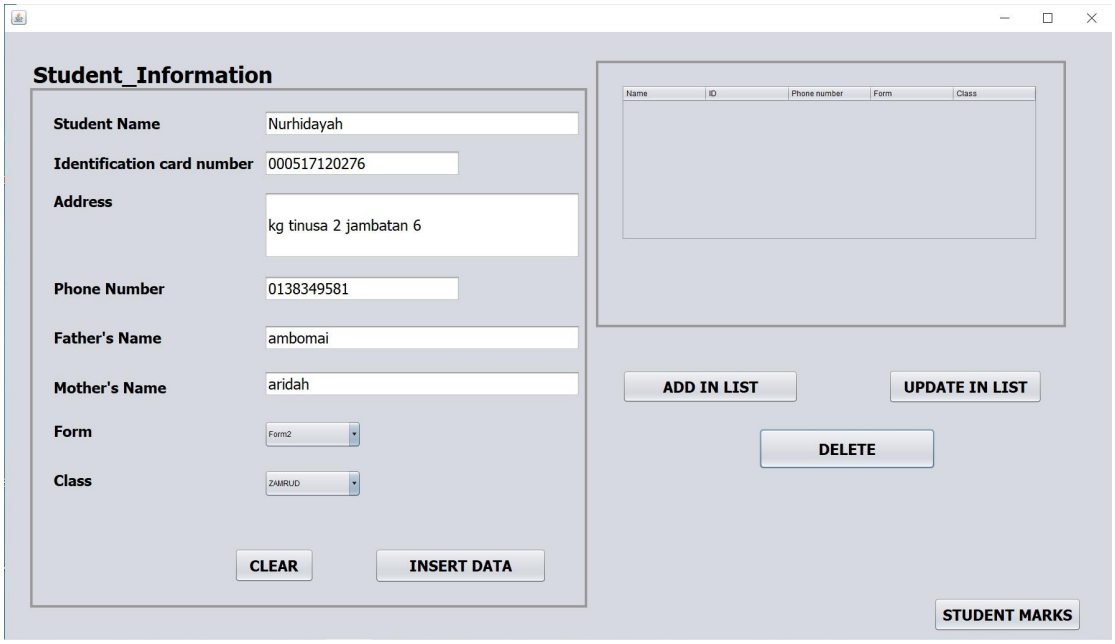
- Add data in table



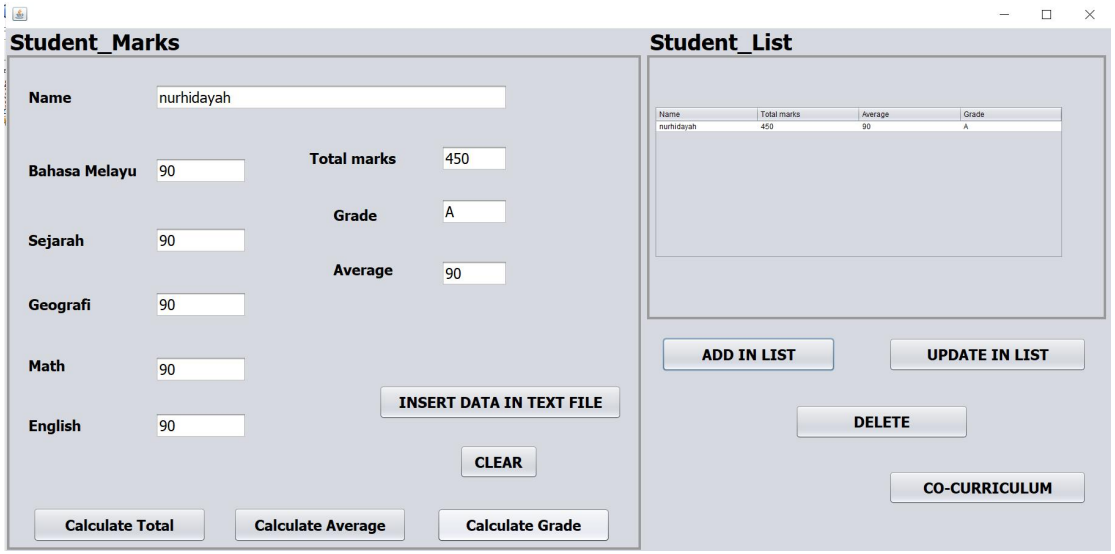
- Update the data



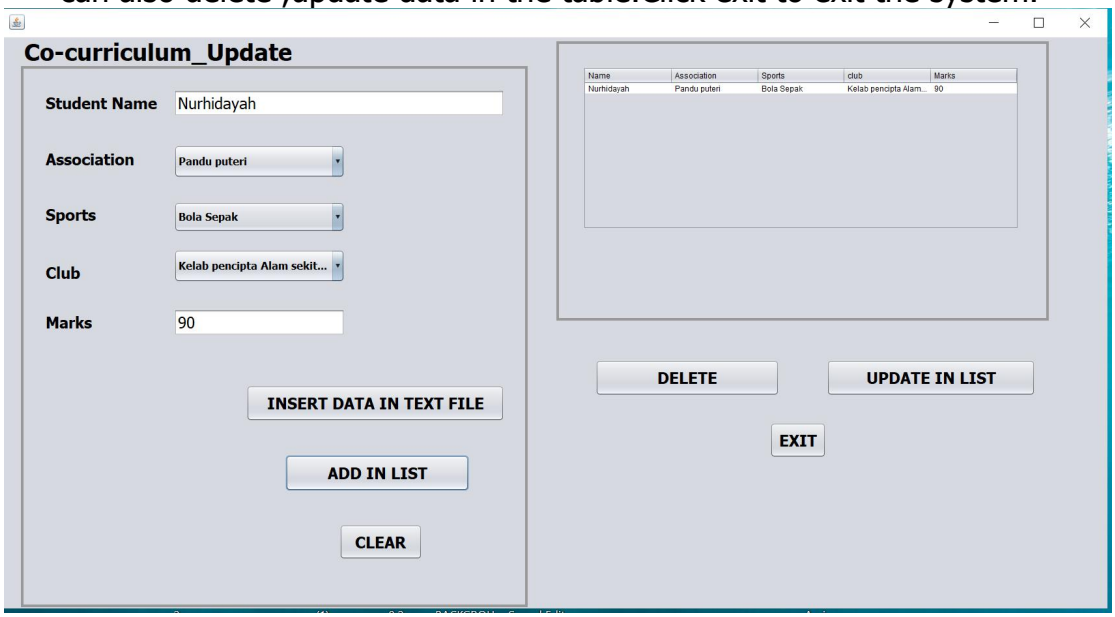
- Delete data in table



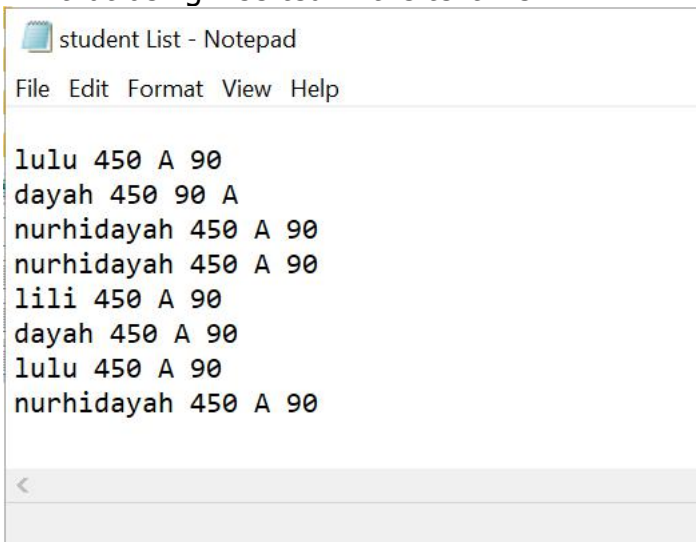
- Click student marks to update studentt marks.in the program you can also add data in text file,add data in table,update data in table,and delete data in table.

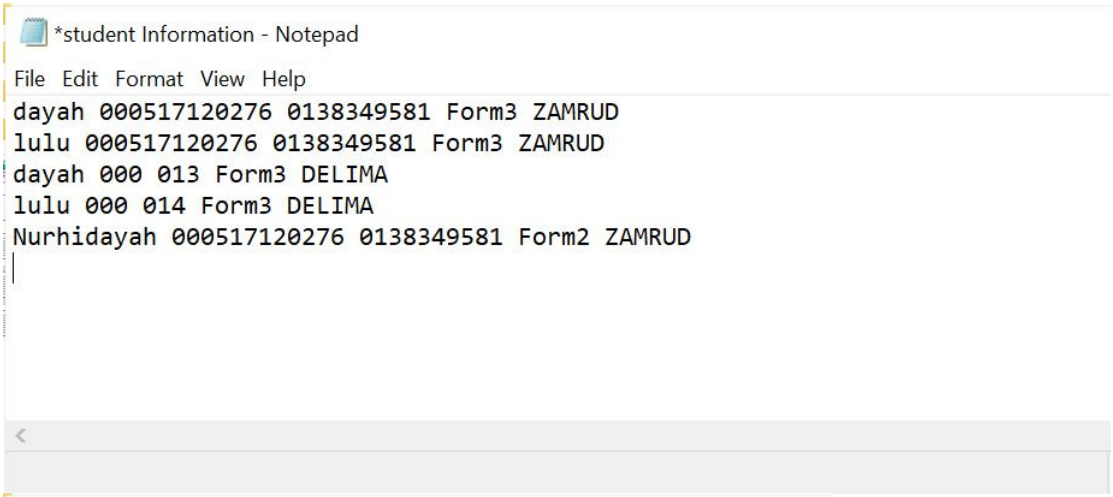


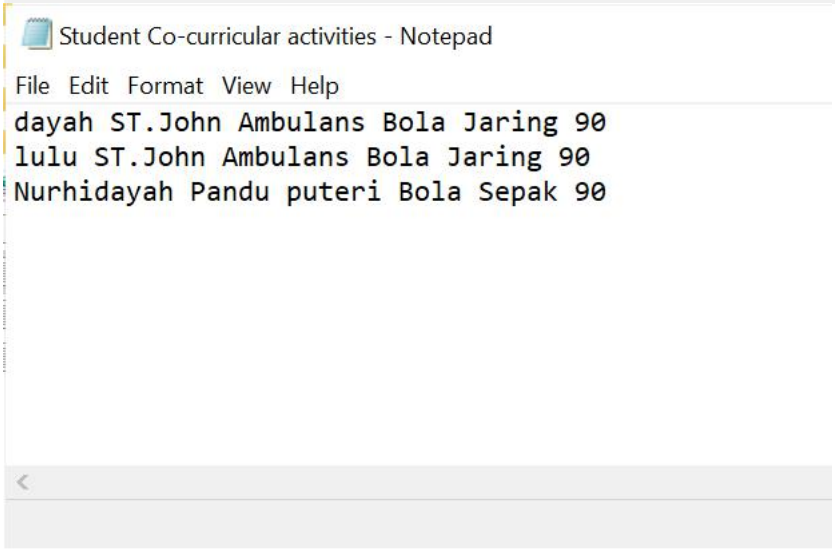
- Click co-curriculum button to update student co-curricular activities .Update student co-curricular activities and insert data in text file.In this system you can also delete ,update data in the table.Click exit to exit the system.



- See student list ,student information and student co-curricular activities that being inserted in the text file




*student Information - Notepad
File Edit Format View Help
dayah 000517120276 0138349581 Form3 ZAMRUD
lulu 000517120276 0138349581 Form3 ZAMRUD
dayah 000 013 Form3 DELIMA
lulu 000 014 Form3 DELIMA
Nurhidayah 000517120276 0138349581 Form2 ZAMRUD


Student Co-curricular activities - Notepad
File Edit Format View Help
dayah ST.John Ambulans Bola Jaring 90
lulu ST.John Ambulans Bola Jaring 90
Nurhidayah Pandu puteri Bola Sepak 90

6.0 Code explanation

- Teacher _login.java(main class)

In the main class ,teacher will need to insert their nama and phone number.Then click update student information button to update student information.

- Secondary_student_performance.java

In the class,teacher will need to update student information .Then register student by click submit button.Then,the teacher need to add the data into text file,add data into table,update data,and delete data if it was mistakenly inputted.

- Student_information.java

In this class,teacher will need to update 5 student marks.Then the teacher will need to click calculate total button to calculate the total of student marks.The teacher also need to click calculate average to calculate average marks.Then the teacher need to clik calculate grade to show student grade.Then,the teacher need to add the data into text file,add data into table,update data,and delete data if it was mistakenly inputted.

- Cocurriculum_update.java

In this class,the teacher need to update student co-curricular activities.Then,the teacher need to add the data into text file,add data into table,update data,and delete data if it was mistakenly inputted.