**Student Management System Documentation**

**1. Introduction**

The Student Management System (SMS) is a Java Swing application that allows users to manage student records, enroll students in courses, and assign grades. This documentation provides an overview of the system's functionality, usage instructions, and code structure.

**2. System Overview**

The SMS provides the following key features:

**Adding Students:** Users can add new students to the system, providing their registration number, name, and age.

**Enrolling Students:** The system enables the enrollment of students in various courses. Students can be enrolled in one or more than one (multiple) courses.

**Assigning Grades:** Users can assign grades to students based on their performance in courses.

**Updating Student Information:** The system allows users to update student information, including name and age.

**3. How to Use the System**

**3.1 Adding a Student**

To add a new student to the system, follow these steps:

Launch the application.

Click on the "Add Student" tab.

Fill in the student's name and age.

Click the "Add Student" button.

**3.2 Enrolling Students in Courses**

To enroll students in courses, use the "Course Enrollment" tab:

Select a course from the dropdown menu.

Select one or more than one students from the list.

Click the "Enroll" button to enroll the selected students in the chosen course.

**3.3 Managing Grades**

To assign grades to students, follow these steps using the "Grades" tab:

Select a student from the dropdown menu.

Choose a course and a grade from the respective dropdowns.

Click the "Assign Grade" button to assign the selected grade to the student in the chosen course.

**3.4 Updating Student Information**

To update student information, follow these steps:

Select a student from the dropdown menu.

Update the student's name and age.

Click the "Update" button to save the changes.

**4. Code Structure**

The code is organized into several classes:

**SMS.java:** The main class that initializes the Swing application.

**GUI.java:** Manages the user interface, including tabs for adding students, enrolling students, and assigning grades.

**addStudent.java:** Handles the process of adding a new student.

**Course.java:** Represents a course in the system.

**CoursePanel.java:** Manages course-related operations, including enrolling students in courses.

**Grade.java:** Represents a grade in the system.

**GradePanel.java:** Manages grade-related operations, such as assigning grades to students.

**Student.java:** Represents a student in the system and handles student-related operations.

**StudentPanel.java:** Manages the student records, including adding and updating students.

**5. D­evelopment Environment**

The project is developed using Java with the Swing Framework for the graphical user interface.

**6. Program Codes**

**SMS.java file:**

package com.mycompany.sms;

import javax.swing.\*;

import javax.swing.plaf.nimbus.NimbusLookAndFeel;

public class SMS {

    public static void main(String[] args) {

        try {

            UIManager.setLookAndFeel(new NimbusLookAndFeel());

        } catch (UnsupportedLookAndFeelException e) {

            e.printStackTrace();

        }

        SwingUtilities.invokeLater(() -> {

            JFrame frame = new JFrame("Student Management System");

            frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

            frame.setSize(800, 600);

            GUI gui = new GUI();

            gui.initialize(frame);

            frame.setVisible(true);

        });

    }

}

**GUI.java file:**

package com.mycompany.sms;

import javax.swing.\*;

public class GUI {

    private JPanel studentPanel;

    private JPanel coursePanel;

    private JPanel gradePanel;

    public void initialize(JFrame frame) {

        JTabbedPane tabbedPane = new JTabbedPane();

        frame.add(tabbedPane);

        studentPanel = new StudentPanel(frame);

        coursePanel = new CoursePanel(frame);

        gradePanel = new GradePanel(frame);

        tabbedPane.addTab("Student Records", studentPanel);

        tabbedPane.addTab("Course Enrollment", coursePanel);

        tabbedPane.addTab("Grades", gradePanel);

    }

}

**Student Panel.java file:**

package com.mycompany.sms;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

public class StudentPanel extends JPanel {

    private DefaultTableModel tableModel;

    private JTable studentTable;

    public StudentPanel(JFrame frame) {

        try {

            UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");

        } catch (Exception e) {

            e.printStackTrace();

        }

        setLayout(new BorderLayout());

        // Create the student table

        studentTable = new JTable(tableModel);

        JScrollPane scrollPane = new JScrollPane(studentTable);

        JButton addButton = new JButton("Add Row");

        addButton.addActionListener(e -> {

            Student std = new Student();

            for (Student student : std.getStudentList()) {

                Object[] rowData = { student.getId(), student.getName(), student.getClass() };

                tableModel.addRow(rowData);

            }

        });

        add(scrollPane, BorderLayout.CENTER);

        JPanel buttonPanel = new JPanel();

        JButton addStudentButton = new JButton("Add Student");

        addStudentButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                JFrame frame = new JFrame("Student Registration");

                addStudent studentPanel = new addStudent();

                frame.dispose();

                frame.add(studentPanel);

                // Set frame properties

                frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                frame.setSize(800, 600);

                frame.setVisible(true);

            }

        });

        buttonPanel.add(addStudentButton);

        JButton updateStudentButton = new JButton("Update Student");

        updateStudentButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                new UpdateStudent();

            }

        });

        buttonPanel.add(updateStudentButton);

        JButton displayStudentDetailsButton = new JButton("Display Student Details");

        displayStudentDetailsButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                Student std = new Student();

                if (std.getStudentList().isEmpty()) {

                    JOptionPane.showMessageDialog(StudentPanel.this, "No student records to display.");

                    return;

                }

                JFrame detailsFrame = new JFrame("Student Details");

                detailsFrame.setSize(600, 400);

                String[] columnNames = { "Reg #", "Name", "Age" };

                DefaultTableModel detailsTableModel = new DefaultTableModel(columnNames, 0);

                JTable detailsTable = new JTable(detailsTableModel);

                for (Student student : std.getStudentList()) {

                    Object[] rowData = { student.getId(), student.getName(), student.getAge(), };

                    detailsTableModel.addRow(rowData);

                }

                JScrollPane tableScrollPane = new JScrollPane(detailsTable);

                detailsFrame.add(tableScrollPane);

                detailsFrame.setVisible(true);

            }

        });

        buttonPanel.add(displayStudentDetailsButton);

        add(buttonPanel, BorderLayout.SOUTH);

    }

    // Method to update the student table with a list of students

    public void updateStudentTable(ArrayList<Student> students) {

        while (tableModel.getRowCount() > 0) {

            tableModel.removeRow(0);

        }

        // Add students to the table

        for (Student student : students) {

            Object[] rowData = { student.getClass(), student.getName() };

            tableModel.addRow(rowData);

        }

    }

}

**Course Panel.java file:**

package com.mycompany.sms;

import javax.swing.\*;

import javax.swing.plaf.nimbus.NimbusLookAndFeel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.List;

public class CoursePanel extends JPanel {

    private JComboBox<String> courseComboBox;

    private JList<String> studentList;

    private JButton enrollButton;

    private List<Student> enrolledStudents;

    public CoursePanel(JFrame frame) {

        try {

            // Set the Nimbus Look and Feel

            UIManager.setLookAndFeel(new NimbusLookAndFeel());

        } catch (UnsupportedLookAndFeelException e) {

            e.printStackTrace();

        }

        setLayout(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();

        constraints.insets = new Insets(5, 5, 5, 5);

        // Course selection dropdown

        courseComboBox = new JComboBox<>();

        courseComboBox.addItem("Programming Fundemental");

        courseComboBox.addItem("Artifitial Intelligence");

        courseComboBox.addItem("Object Oriented Programming");

        addField(courseComboBox, constraints, 0, 1);

        // Student list

        DefaultListModel<String> studentListModel = new DefaultListModel<>();

        studentList = new JList<>(studentListModel);

        studentList.setSelectionMode(ListSelectionModel.MULTIPLE\_INTERVAL\_SELECTION);

        addField(new JScrollPane(studentList), constraints, 0, 2);

        // Enroll button

        enrollButton = new JButton("Enroll");

        addField(enrollButton, constraints, 0, 3);

        enrolledStudents = new ArrayList<>();

        Student std = new Student();

        for (Student student : std.getStudentList()) {

            studentListModel.addElement(Integer.toString(student.getId()) + " " + student.getName());

        }

        // Enroll button action

        enrollButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                try {

                    String selectedCourse = (String) courseComboBox.getSelectedItem();

                    System.out.println(selectedCourse);

                    int[] selectedIndices = studentList.getSelectedIndices();

                    for (int index : selectedIndices) {

                        String selectedStudentName = studentListModel.get(index);

                        String[] parts = selectedStudentName.split(" ");

                        if (parts.length > 0) {

                            int id = Integer.parseInt(parts[0]);

                            std.setEnrolledCourse(selectedCourse, id);

                            enrolledStudents.add(new Student());

                        }

                    }

                    if (enrolledStudents.size() == 0) {

                        JOptionPane.showMessageDialog(null, "Please Select Student first");

                    } else {

                        JOptionPane.showMessageDialog(null, "Students Succesfully Enrolled");

                    }

                } catch (Exception error) {

                    JOptionPane.showMessageDialog(null, "Error! " + error.getMessage());

                }

            }

        });

    }

    private void addField(JComponent component, GridBagConstraints constraints, int gridx, int gridy) {

        constraints.gridx = gridx;

        constraints.gridy = gridy;

        constraints.fill = GridBagConstraints.HORIZONTAL;

        add(component, constraints);

    }

}

**Grade panel.java file:**

package com.mycompany.sms;

import javax.swing.\*;

import javax.swing.plaf.nimbus.NimbusLookAndFeel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.table.DefaultTableModel;

public class GradePanel extends JPanel {

    private JComboBox<String> studentComboBox;

    private JComboBox<String> courseComboBox;

    private JComboBox<String> gradeComboBox;

    private JTable gradeTable;

    private JButton assignGradeButton;

    private JButton cancelButton;

    public GradePanel(JFrame frame) {

        try {

            // Set the Nimbus Look and Feel

            UIManager.setLookAndFeel(new NimbusLookAndFeel());

        } catch (UnsupportedLookAndFeelException e) {

            e.printStackTrace();

        }

        setLayout(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();

        constraints.insets = new Insets(5, 5, 5, 5);

        setLayout(new GridBagLayout());

        // Create a container to hold the JComboBox components in one line

        JPanel comboBoxContainer = new JPanel();

        comboBoxContainer.setLayout(new FlowLayout(FlowLayout.CENTER)); // Adjust alignment as needed

        // Student selection dropdown

        studentComboBox = new JComboBox<>();

        comboBoxContainer.add(studentComboBox);

        studentComboBox.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                updateCourseComboBox();

            }

        });

        courseComboBox = new JComboBox<>();

        comboBoxContainer.add(courseComboBox);

        // Grade selection dropdown

        gradeComboBox = new JComboBox<>();

        gradeComboBox.addItem("Select Grade");

        gradeComboBox.addItem("A");

        gradeComboBox.addItem("B");

        gradeComboBox.addItem("C");

        gradeComboBox.addItem("D");

        gradeComboBox.addItem("F");

        comboBoxContainer.add(gradeComboBox);

        addField(comboBoxContainer, constraints, 0, 0);

        // Create the grade table

        String[] columns = { "Student", "Course", "Grade" };

        DefaultTableModel model = new DefaultTableModel(columns, 0);

        gradeTable = new JTable(model);

        JScrollPane scrollPane = new JScrollPane(gradeTable);

        addField(scrollPane, constraints, 0, 1);

        // Assign grade button

        assignGradeButton = new JButton("Assign Grade");

        assignGradeButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                // Get the selected student and grade

                String selectedStudent = (String) studentComboBox.getSelectedItem();

                String selectedCourse = (String) courseComboBox.getSelectedItem();

                String selectedGrade = (String) gradeComboBox.getSelectedItem();

                if (selectedStudent != null && selectedGrade != null) {

                    model.addRow(new Object[] { selectedStudent, selectedCourse, selectedGrade });

                }

            }

        });

        addField(assignGradeButton, constraints, 0, 2);

        // Cancel button

        cancelButton = new JButton("Close");

        cancelButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                frame.dispose();

                JFrame frame = new JFrame("Student Management System");

                frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                frame.setSize(800, 600);

                GUI gui = new GUI();

                gui.initialize(frame);

                frame.setVisible(true);

            }

        });

        addField(cancelButton, constraints, 1, 2);

        updateStudentComboBox();

    }

    private void addField(JComponent component, GridBagConstraints constraints, int gridx, int gridy) {

        constraints.gridx = gridx;

        constraints.gridy = gridy;

        constraints.fill = GridBagConstraints.HORIZONTAL;

        add(component, constraints);

    }

    private void updateStudentComboBox() {

        Student std = new Student();

        studentComboBox.addItem("Select Student");

        for (Student student : std.getStudentList()) {

            studentComboBox.addItem(Integer.toString(student.getId()) + " " + student.getName());

        }

    }

    private void updateCourseComboBox() {

        String selectedStudent = (String) studentComboBox.getSelectedItem();

        if (selectedStudent != null && !selectedStudent.equals("Select Student")) {

            String[] parts = selectedStudent.split(" ");

            int id = Integer.parseInt(parts[0]);

            Student std = new Student();

            for (Student student : std.getStudentList()) {

                if (student.getId() == id) {

                    for (String selectedSTD : student.getEnrolledCourse()) {

                        courseComboBox.addItem(selectedSTD);

                    }

                }

            }

        }

    }

}

**Student.java file:**

package com.mycompany.sms;

import java.util.ArrayList;

public class Student {

    private static ArrayList<Student> studentList = new ArrayList<>();

    private int id;

    private String name;

    private int age;

    private ArrayList<String> enrolledcourse = new ArrayList<>();

    public ArrayList<Student> getStudentList() {

        return studentList;

    }

    public void setStudentList(Student studentList) {

        Student.studentList.add(studentList);

    }

    public void setEnrolledCourse(String course, int id) {

        for (int i = 0; i < studentList.size(); i++) {

            if (studentList.get(i).id == id) {

                studentList.get(i).enrolledcourse.add(course);

            }

        }

    }

    public int getAge() {

        return age;

    }

    public ArrayList<String> getEnrolledCourse() {

        return enrolledcourse;

    }

    public void setAge(int age) {

        this.age = age;

    }

    public void setId(int id) {

        this.id = id;

    }

    public void setName(String name) {

        this.name = name;

    }

    public Student() {

    }

    public Student(int id, String name, int age) {

        this.id = id;

        this.name = name;

        this.age = age;

        System.out.println(studentList);

    }

    public String getName() {

        return name;

    }

    public int getId() {

        return id;

    }

}

**Update Student.java file:**

package com.mycompany.sms;

import javax.swing.\*;

import javax.swing.plaf.nimbus.NimbusLookAndFeel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class UpdateStudent extends JPanel {

    private JTextField regNoField;

    private JTextField nameField;

    private JTextField ageField;

    private JComboBox<String> studentComboBox;

    private JButton updateButton;

    private JButton resetButton;

    Student std = new Student();

    public UpdateStudent() {

        try {

            // Set the Nimbus Look and Feel

            UIManager.setLookAndFeel(new NimbusLookAndFeel());

        } catch (UnsupportedLookAndFeelException e) {

            e.printStackTrace();

        }

        JFrame frame = new JFrame("Student Management System");

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        frame.setSize(800, 600);

        setLayout(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();

        constraints.insets = new Insets(5, 5, 5, 5);

        // Create a helper method to simplify adding components with constraints

        ActionListener resetListener = new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                frame.dispose();

                JFrame frame = new JFrame("Student Management System");

                frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                frame.setSize(800, 600);

                GUI gui = new GUI();

                gui.initialize(frame);

                frame.setVisible(true);

            }

        };

        Student student = new Student();

        studentComboBox = new JComboBox<>();

        for (Student std : student.getStudentList()) {

            studentComboBox.addItem(String.valueOf(std.getId()));

        }

        studentComboBox.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                // When a student is selected from the combo box, display their information

                int selectedIndex = studentComboBox.getSelectedIndex();

                if (selectedIndex >= 0) {

                    Student student = std.getStudentList().get(selectedIndex);

                    regNoField.setText(Integer.toString(student.getId()));

                    nameField.setText(student.getName());

                    ageField.setText(Integer.toString(student.getAge()));

                }

            }

        });

        // Add the combo box

        addStyledLabel("Select Student:", constraints, 0, 0);

        addField(studentComboBox, constraints, 1, 0);

        addStyledLabel("Registration Number:", constraints, 0, 2);

        regNoField = createStyledTextField();

        addField(regNoField, constraints, 1, 2);

        regNoField.setEditable(false);

        addStyledLabel("Name:", constraints, 0, 3);

        nameField = createStyledTextField();

        addField(nameField, constraints, 1, 3);

        addStyledLabel("Age:", constraints, 0, 4);

        ageField = createStyledTextField();

        addField(ageField, constraints, 1, 4);

        updateButton = new JButton("Update");

        updateButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                // Update the selected student's information

                int selectedIndex = studentComboBox.getSelectedIndex();

                if (selectedIndex >= 0) {

                    try {

                        int regNo = Integer.parseInt(regNoField.getText());

                        String name = nameField.getText();

                        int age = Integer.parseInt(ageField.getText());

                        Student student = std.getStudentList().get(selectedIndex);

                        student.setId(regNo);

                        student.setName(name);

                        student.setAge(age);

                        updateStudentComboBox();

                        JOptionPane.showMessageDialog(null, "Update SucessFully.");

                        System.out.println("Check");

                        frame.dispose();

                        JFrame frame = new JFrame("Student Management System");

                        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                        frame.setSize(800, 600);

                        GUI gui = new GUI();

                        gui.initialize(frame);

                        frame.setVisible(true);

                    } catch (NumberFormatException ex) {

                        JOptionPane.showMessageDialog(null, "Invalid input. Please enter valid data.");

                    }

                }

            }

        });

        addField(updateButton, constraints, 0, 5);

        resetButton = new JButton("Cancel");

        resetButton.addActionListener(resetListener);

        addField(resetButton, constraints, 1, 5);

        frame.add(this);

        frame.setVisible(true);

    }

    // Helper method to create styled labels and add to panel with constraints

    private void addStyledLabel(String text, GridBagConstraints constraints, int gridx, int gridy) {

        JLabel label = createStyledLabel(text);

        constraints.gridx = gridx;

        constraints.gridy = gridy;

        constraints.anchor = GridBagConstraints.WEST;

        add(label, constraints);

    }

    // Helper method to add components to panel with constraints

    private void addField(JComponent component, GridBagConstraints constraints, int gridx, int gridy) {

        constraints.gridx = gridx;

        constraints.gridy = gridy;

        constraints.fill = GridBagConstraints.HORIZONTAL;

        add(component, constraints);

    }

    // Helper method to create styled labels

    private JLabel createStyledLabel(String text) {

        JLabel label = new JLabel(text);

        label.setFont(new Font("Arial", Font.BOLD, 14));

        label.setForeground(new Color(0, 102, 204)); // Blue color

        return label;

    }

    // Helper method to create styled text fields

    private JTextField createStyledTextField() {

        JTextField textField = new JTextField();

        textField.setFont(new Font("Arial", Font.PLAIN, 14));

        textField.setPreferredSize(new Dimension(200, 30));

        return textField;

    }

    // Helper method to update the student combo box

    private void updateStudentComboBox() {

        studentComboBox.removeAllItems();

        for (Student student : std.getStudentList()) {

            studentComboBox.addItem(student.getName());

        }

    }

}

**Add Student.java file:**

package com.mycompany.sms;

import javax.swing.\*;

import javax.swing.plaf.nimbus.NimbusLookAndFeel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class addStudent extends JPanel {

    private JTextField regNoField;

    private JTextField nameField;

    private JTextField ageField;

    private JButton addButton;

    private JButton resetButton;

    public addStudent() {

        try {

            // Set the Nimbus Look and Feel

            UIManager.setLookAndFeel(new NimbusLookAndFeel());

        } catch (UnsupportedLookAndFeelException e) {

            e.printStackTrace();

        }

        JFrame frame = new JFrame("Student Management System");

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        frame.setSize(800, 600);

        frame.setVisible(true);

        setLayout(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();

        constraints.insets = new Insets(5, 5, 5, 5);

        // Create a helper method to simplify adding components with constraints

        ActionListener resetListener = new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                // Reset the input fields

                clearInputFields();

            }

        };

        Student std = new Student();

        addStyledLabel("Registration Number:", constraints, 0, 0);

        regNoField = createStyledTextField();

        regNoField.setText(String.valueOf(std.getStudentList().size() + 1));

        regNoField.setEditable(false);

        addField(regNoField, constraints, 1, 0);

        addStyledLabel("Name:", constraints, 0, 1);

        nameField = createStyledTextField();

        addField(nameField, constraints, 1, 1);

        addStyledLabel("Age:", constraints, 0, 2);

        ageField = createStyledTextField();

        addField(ageField, constraints, 1, 2);

        addButton = new JButton("Add Student");

        addButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                try {

                    int regNo = Integer.parseInt(regNoField.getText());

                    String name = nameField.getText();

                    int age = Integer.parseInt(ageField.getText());

                    Student student = new Student(regNo, name, age);

                    student.getStudentList();

                    // student.studentList.add(student);

                    student.setStudentList(student);

                    System.out.println(student.getStudentList());

                    clearInputFields();

                    JOptionPane.showMessageDialog(null, "Added Sucessfully");

                    frame.dispose();

                    JFrame frame = new JFrame("Student Management System");

                    frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

                    frame.setSize(800, 600);

                    GUI gui = new GUI();

                    gui.initialize(frame);

                    frame.setVisible(true);

                } catch (NumberFormatException ex) {

                    // Handle the case where parsing fails (e.g., invalid input)

                    JOptionPane.showMessageDialog(null, "Invalid input. Please enter valid data.");

                }

            }

        });

        addField(addButton, constraints, 0, 3);

        resetButton = new JButton("Reset");

        resetButton.addActionListener(resetListener);

        addField(resetButton, constraints, 1, 3);

        frame.add(this);

    }

    // Helper method to create styled labels and add to panel with constraints

    private void addStyledLabel(String text, GridBagConstraints constraints, int gridx, int gridy) {

        JLabel label = createStyledLabel(text);

        constraints.gridx = gridx;

        constraints.gridy = gridy;

        constraints.anchor = GridBagConstraints.WEST;

        add(label, constraints);

    }

    // Helper method to add components to panel with constraints

    private void addField(JComponent component, GridBagConstraints constraints, int gridx, int gridy) {

        constraints.gridx = gridx;

        constraints.gridy = gridy;

        constraints.fill = GridBagConstraints.HORIZONTAL;

        add(component, constraints);

    }

    // Helper method to create styled labels

    private JLabel createStyledLabel(String text) {

        JLabel label = new JLabel(text);

        label.setFont(new Font("Arial", Font.BOLD, 14));

        label.setForeground(new Color(0, 102, 204));

        return label;

    }

    // Helper method to create styled text fields

    private JTextField createStyledTextField() {

        JTextField textField = new JTextField();

        textField.setFont(new Font("Arial", Font.PLAIN, 14));

        textField.setPreferredSize(new Dimension(200, 30));

        return textField;

    }

    // Helper method to clear input fields

    private void clearInputFields() {

        nameField.setText("");

        ageField.setText("");

    }

}

**Course.java file:**

package com.mycompany.sms;

import java.util.ArrayList;

public class Course {

    private static ArrayList<String[]> courseList = new ArrayList<>();

    private String name;

    public Course(String[] name) {

        Course.courseList.add(name);

    }

    public String getName() {

        return name;

    }

}

**Grade.java file:**

package com.mycompany.sms;

public class Grade {

    private String value;

    public Grade(String value) {

        this.value = value;

    }

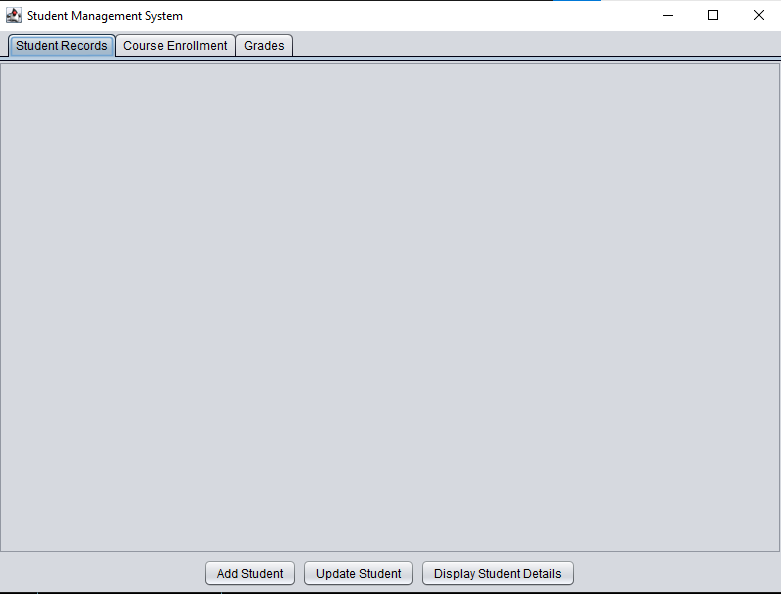
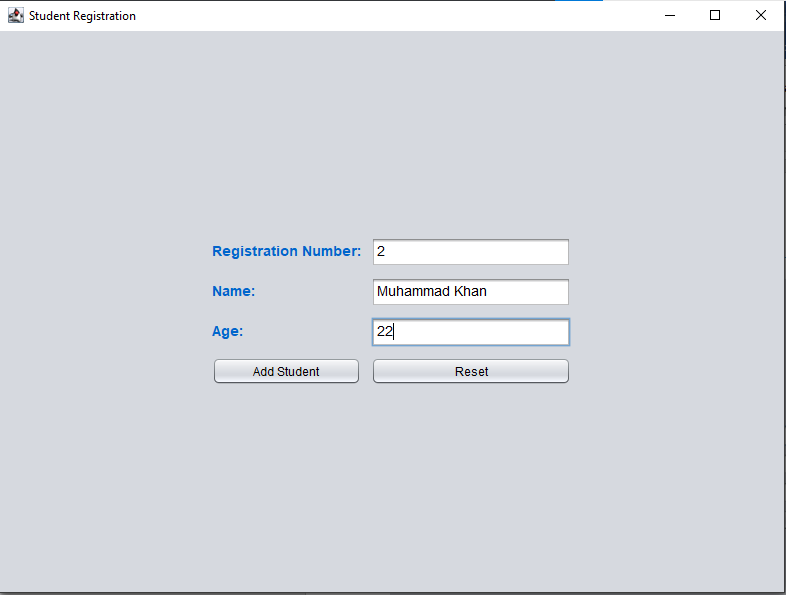
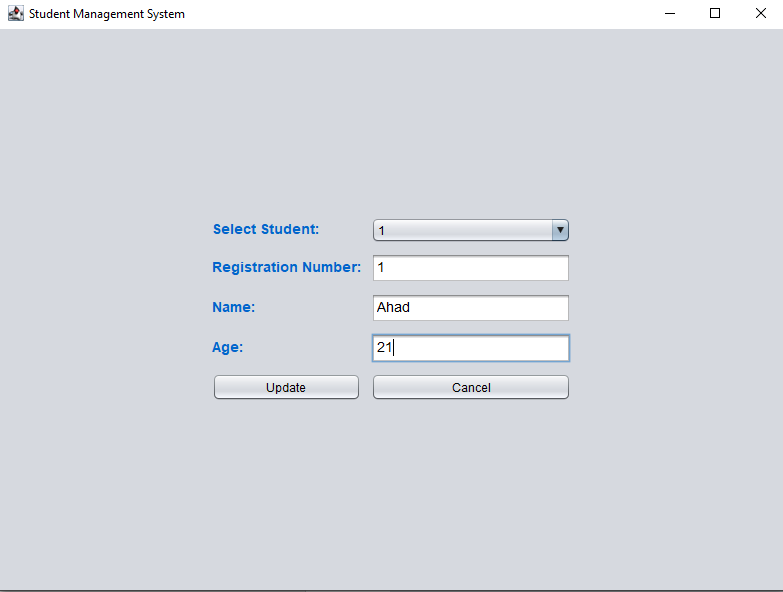
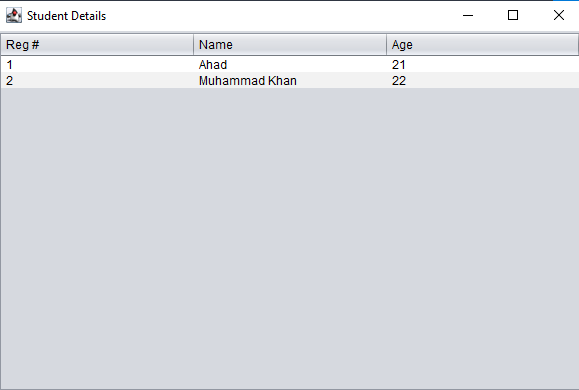
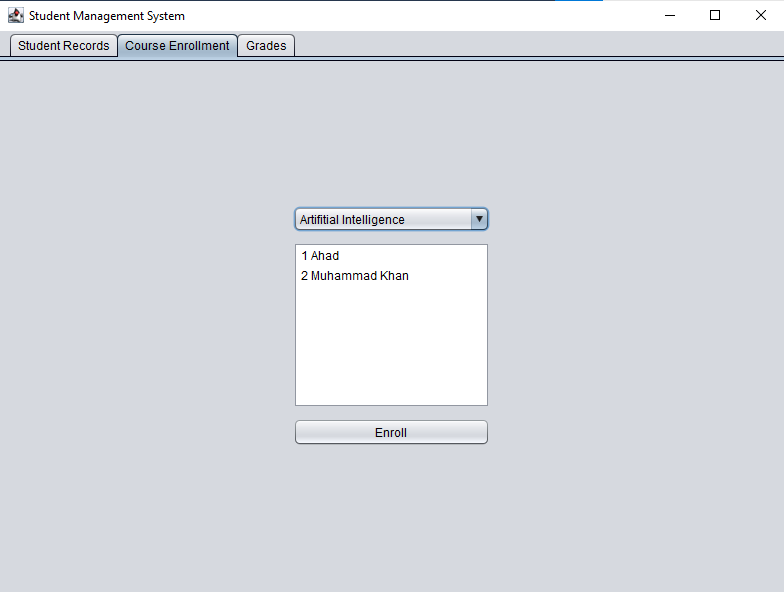
    public String getValue() {

        return value;

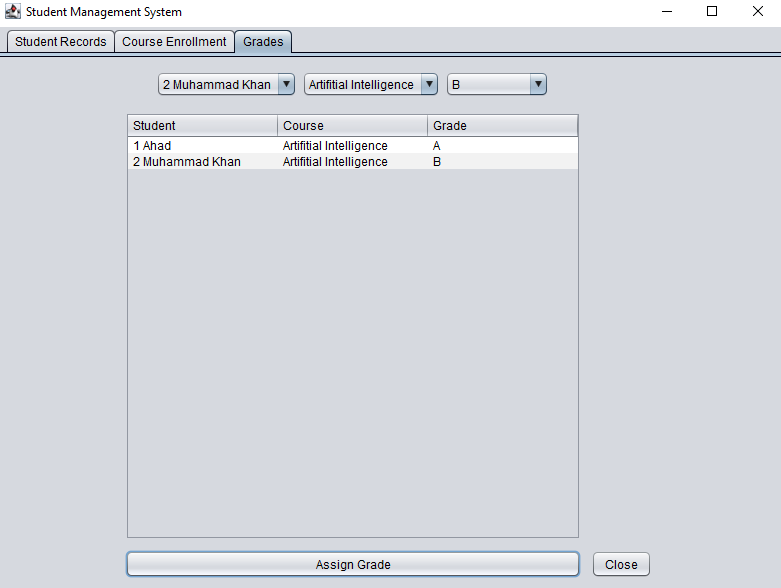
    }

}

**7. Outputs**

Use **ctr(control)** key to select multiple students



**The End**