Student Management System - Project Documentation

1. Project Overview

The Student Management System is a GUI-based Java application developed using the Swing framework. It allows users to add student records consisting of names and roll numbers. The application focuses on key development aspects such as core feature implementation, event handling, robust data validation, and clear code structure to demonstrate technical proficiency.

2. Features Implemented

- Add student records (Name and Roll Number).

- Input validation for completeness and numeric constraints.

- Clear input fields functionality.

- Display all student records in a text area.

3. Technologies Used

- Java SE 8+

- Swing GUI Toolkit

4. How to Run

1. Open the source code in an IDE like IntelliJ IDEA or Eclipse.
2. Compile and run the `StudentManagementSystem` class.
3. Use the GUI to add and view student records.

5. Evaluation Criteria (Rubric Alignment)

- Core Feature Implementation: Functionality for adding and displaying students is fully implemented.

- Error Handling and Robustness: Validations and error dialogs ensure stability and usability.

- Integration of Components: UI components, event listeners, and logic are tightly integrated.

- Event Handling and Processing: Uses ActionListeners for button click events.

- Data Validation: Checks for empty fields and numeric roll number format.

- Code Quality and Innovation: Clean, readable code with scope for expansion (e.g., edit/delete features).

- Documentation: This document serves as the project's technical documentation.

Prepared by: Javed Khan
Date: 13th June 2025

Source code

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.util.ArrayList;


public class StudentManagementSystem extends JFrame {

   private JTextField nameField, rollField;

   private JTextArea displayArea;

   private ArrayList<Student> students;


   public StudentManagementSystem() {

      setTitle("Student Management System");

      setSize(500, 400);

      setDefaultCloseOperation(EXIT_ON_CLOSE);

      setLayout(new BorderLayout());
```

```
students = new ArrayList<>();

JPanel inputPanel = new JPanel(new GridLayout(3, 2));
inputPanel.add(new JLabel("Name:"));
nameField = new JTextField();
inputPanel.add(nameField);

inputPanel.add(new JLabel("Roll Number:"));
rollField = new JTextField();
inputPanel.add(rollField);

JButton addButton = new JButton("Add Student");
inputPanel.add(addButton);

JButton clearButton = new JButton("Clear");
inputPanel.add(clearButton);

add(inputPanel, BorderLayout.NORTH);

displayArea = new JTextArea();
displayArea.setEditable(false);
add(new JScrollPane(displayArea), BorderLayout.CENTER);

// Event Handlers
addButton.addActionListener(e -> addStudent());
```

```
        clearButton.addActionListener(e -> clearFields());
    }

    private void addStudent() {
        String name = nameField.getText().trim();
        String roll = rollField.getText().trim();

        if (name.isEmpty() || roll.isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields must be filled!", "Validation Error",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        try {
            int rollNum = Integer.parseInt(roll);
            Student s = new Student(name, rollNum);
            students.add(s);
            updateDisplay();
            clearFields();
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Roll number must be an integer!", "Input
Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void clearFields() {
        nameField.setText("");
```

```
        rollField.setText("");
    }

    private void updateDisplay() {
        StringBuilder sb = new StringBuilder();
        for (Student s : students) {
            sb.append("Name: ").append(s.getName())
              .append(", Roll: ").append(s.getRollNumber())
              .append("\n");
        }
        displayArea.setText(sb.toString());
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new StudentManagementSystem().setVisible(true));
    }
}

class Student {
    private String name;
    private int rollNumber;

    public Student(String name, int rollNumber) {
        this.name = name;
        this.rollNumber = rollNumber;
    }
```

```
    public String getName() {
        return name;
    }

    public int getRollNumber() {
        return rollNumber;
    }
}
```

- https://github.com/you -javed khan /lib a y management-system
- https://github.com/khanjaved786/Review-2-