Student Management System Documentation

# Table of Contents

1. Problem Statement

2. Introduction

3. Objectives

4. Features

5. System Design

6. Class Diagrams

7. Implementation Details

8. Limitations

9. Future Enhancements

10. Conclusion

# Problem Statement

Managing student records, exams, and results manually can lead to errors, inconsistency, and inefficiency. An automated system is required to handle student registration and examination details efficiently.

# Introduction

This Student Management System is a console-based Java application that allows users to register students, store their course and exam details, and display examination results with performance grades and percentage.

# Objectives

- To create a system that handles student records and their examination performance.

- To provide a simple user interface for entering and retrieving data.

- To calculate grades and percentages based on exam marks.

# Features

- Register new students with details like name, roll number, email, DOB, course, etc.

- Add and store examination results for each student.

- Automatically calculate grades and percentages for each subject and overall performance.

- View student profiles and their exam results by roll number.

- Menu-driven interface with simple input prompts.

# System Design

The system follows an object-oriented approach:

- Student class: Stores personal and academic information.

- Subject class: Holds individual subject marks and grade.

- Examination class: Stores a list of subjects, computes overall results.

- StudentManagementSystem: Main class containing user menu and control logic.

# Class Diagrams

Class: Student

-------------------------------

- name: String

- rollNumber: int

- dob: String

- email: String

- studentId: String

- course: String

+ displayDetails(): void

Class: Subject

-------------------------------

- name: String

- marks: int

- grade: String

- percentage: double

+ displaySubjectDetails(): void

Class: Examination

-------------------------------

- subjects: List<Subject>

- overallPercentage: double

- overallGrade: String

+ addSubject(subject): void

+ displayExamResults(): void

# Implementation Details

Java collections like ArrayList and HashMap are used for dynamic data storage. Subjects and grades are entered and calculated within the Examination class. Roll number is used as a unique identifier for students. Main menu gives five options: Add Student, Add Exam, View Student, View Results, Exit.

# Limitations

- No data persistence (e.g., file or database storage), data is lost after program ends.

- No login/authentication mechanism.

- Only 5 subjects are supported per exam entry, not customizable.

# Future Enhancements

- Integrate file/database for persistent storage.

- Add GUI for better usability.

- Enable editing and deleting records.

- Add authentication and role-based access (e.g., Admin, Student).

- Generate printable reports or transcripts.

# Conclusion

The Student Management System is a simple, efficient, and modular solution for managing student records and examination results. It lays a solid foundation for further development into a more robust and full-featured academic management application.