STUDENT MANAGEMENT SYSTEM PLUS

Object Oriented Programming Using Java

Name: Bassam Sinare

Course: Object Oriented Programming

Level: 400

Date: February 2026

1. Introduction

This project presents the design and implementation of the Student Management System Plus. The system replaces spreadsheet-based record management with a structured desktop application built using Java, JavaFX, SQLite, JDBC, and Maven. It runs offline on Windows systems.

2. Problem Statement

Manual spreadsheet management causes duplicate IDs, missing records, inconsistent data, and weak reporting. The system aims to provide validated data storage, secure database access, and reliable reporting features.

3. Objectives

- Apply object-oriented programming principles.
- Build a JavaFX graphical user interface.
- Implement SQLite database using JDBC.
- Enforce strong validation rules.
- Generate academic performance reports.
- Support CSV import and export.
- Implement unit testing with Maven.

4. System Architecture

The application follows a layered architecture:
- UI Layer – JavaFX controllers and views.
- Domain Layer – Student model class.
- Service Layer – Business logic and validation.
- Repository Layer – SQLite database operations.
- Utility Layer – Logging and file handling.

5. Functional Features

- Add, update, delete, and view students.
- Search by ID or full name.
- Filter by programme, level, and status.
- Sort by GPA and full name.
- Dashboard with student statistics.

6. Reports

- Top 10 students by GPA.
- At-risk students below configurable GPA threshold.
- GPA distribution summary.
- Programme summary with averages.

All reports can be exported to CSV format.

7. Data Validation

Validation is applied in both UI and Service layers.
Rules include unique student ID, GPA between 0.0 and 4.0, valid email format, correct phone number length, and valid academic level.

8. Database Design

SQLite is used with primary key constraints, NOT NULL constraints, CHECK constraints, and prepared statements to prevent SQL injection.

9. Testing

At least 12 unit tests were written covering validation logic, report calculations, and repository methods. Tests were executed using Maven (mvn test).

10. Logging

Application logs include system start and shutdown, CRUD operations, import/export summaries, and database errors. Logs are saved in the data folder.

11. Challenges

- Managing background threads during import.
- Maintaining clean architecture.
- Implementing flexible report filtering.
- Handling duplicate IDs during import.

12. Conclusion

The Student Management System Plus successfully provides a structured, validated, and user-friendly solution for managing student records. The project demonstrates strong understanding of OOP, database integration, testing, and professional software delivery.