Student Management System

Course: CS310 Advanced Programming (Java)

Semester: Spring 2024/2025

Team Members:

- Yaqeen Altobat

-

-

# Table of Contents

1. Introduction and Motivation

2. Application Overview

3. Functionalities List

4. User Interface Screens (Mockups)

5. Class Diagrams

6. Code Structure Explanation

7. Exception Handling Strategy

8. Conclusion & Lessons Learned

9. Division of Work Table

10. Code Link on GitHub

11. References

# 1. Introduction and Motivation

This project aims to apply core Java programming principles by building a desktop-based Student Management System. The system will allow users to add, view, and manage student records through an interactive GUI. The project showcases OOP design, exception handling, GUI building using Swing, and GitHub collaboration.

# 2. Application Overview

The application is structured with a model-view-manager architecture. Users can perform key operations such as adding new students, listing all students, and reviewing data in a user-friendly GUI.

# 3. Functionalities List

- Add new students with ID, Name, Major, GPA

- Display list of all added students

- Delete a student by ID (planned future feature)

- GUI with 3 separate screens

- Input validation and exception handling

- Polymorphic access via Admin and Guest roles

# 4. User Interface Screens (Mockups)

The following mockups show the main application windows:

- Main Menu Screen

- Add Student Screen

- View Student List Screen

# 5. Class Diagrams

See the UML class diagram attached in the project folder. It represents the structure and relationships between classes: User, Admin, Guest, Student, StudentManager, and GUI frames.

# 6. Code Structure Explanation

- model/: contains Student and User-related classes

- manager/: contains logic for managing student data

- ui/: GUI components using Java Swing

- Main.java: entry point for the application

# 7. Exception Handling Strategy

The application uses try-catch blocks in the AddStudentFrame to handle invalid input (non-numeric ID or GPA). Exception messages are shown using JOptionPane to guide the user.

# 8. Conclusion & Lessons Learned

This project helped us reinforce object-oriented programming concepts, GUI design using Swing, and the importance of input validation. It also improved our ability to work with modular Java structures.

# 9. Division of Work Table

- Ashraf Awwad: All components (solo project)

# 10. Code Link on GitHub

GitHub Repository: https://github.com/yourusername/StudentManagementSystem

# 11. References

- Java API Docs: https://docs.oracle.com/javase/8/docs/api/

- Java Swing Tutorial: https://docs.oracle.com/javase/tutorial/uiswing/

- NetBeans + GitHub Integration: https://www.youtube.com/watch?v=6U7\_Om4zffM