Software Requirements Specification

for

A Visualization Tool for analysing Student scores(Analytics)

Version 1.0

Prepared by

Group Name: Data Wizards

Penumala Vishal	**SE22UARI126**	**Se22uari126@mahindrauniversity.edu.in**
Bollam Siddharth	**SE22UARI160**	**Se22uari160@mahindrauniversity.edu.in**
Mokila Harshitha Reddy	**SE22UARI097**	**Se22uari097@mahindrauniversity.edu.in**
T.Harshitha	**SE22UARI157**	**Se22uari157@mahindrauniversity.edu.in**
Abhiram	**SE22UARI004**	**Se22uari004@mahindrauniversity.edu.in**

Instructor:	**Avinash Arun Chauhan**
Course:	**Software Engineering**
Lab Section:	***Software Engineering***

Teaching Assistant: ***Narthanni .K***

Date: **28/02/2025**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

1.1 Document Purpose

This document defines the software requirements for the Student Score Visualization Tool. The tool will provide an interactive 3D visualization of student scores with live updates, analytics, and heatmaps to indicate subject strengths and weaknesses. The system will support two perspectives: students (who can view scores) and teachers (who can edit scores). It will include GPA calculation, data storage, a login system, and a student details page. AI integration for predictive analysis may be added in later versions.

1.2 Product Scope

The Student Score Visualization Tool is designed to modernize the way student performance data is analyzed and presented. It aims to:

- Enhance score analysis using dynamic 3D graphs.
- Provide heatmaps to visualize subject strengths/weaknesses.
- Offer real-time analytics for teachers and students.
- Enable teachers to input scores, automatically calculating GPAs.
- Allow students to track performance trends.
- Implement anti-cheating mechanisms to prevent manipulation of marks.
- Ensure seamless database connectivity for efficient data management.

Benefits:

- Provides better insights than traditional tabular grading systems.
- Engages users through modern, visually appealing 3D charts.
- Helps teachers track student progress and identify struggling students.
- Encourages students to improve performance using detailed analytics.
- Prevents unauthorized mark manipulation through secure anti-cheating measures.
- Ensures reliable database connection for data consistency and integrity.

1.3 Intended Audience and Document Overview

This document is intended for:

- Developers – To understand the system requirements and implementation details.
- Project Managers – To oversee progress and ensure system goals are met.
- Professors/Instructors – To review project feasibility and academic relevance.
- End Users (Students & Teachers) – To understand the tool’s functionality.

1.4 Definitions, Acronyms and Abbreviations

● 3D Visualization – A graphical representation of data using three dimensions.
● GPA (Grade Point Average) – A calculated metric representing a student's academic performance.
● Heatmap – A visual representation of data where colors indicate value intensity.
● AI (Artificial Intelligence) – Machine learning models for predictive analytics.
● Anti-Cheating Mechanism – A security feature that prevents unauthorized modifications of student scores.
● Database Connectivity – Ensures smooth communication between backend and database.

1.5 Document Conventions

- Uses Arial font size 11 for text.
- Section and subsection titles follow IEEE formatting.
- Italics are used for comments or placeholders.

1.6 References and Acknowledgments

- [IEEE Software Requirements Specification Template]
- [MongoDB Documentation]
- [D3.js Data Visualization Library]

2 Overall Description

2.1 Product Overview

The Student Score Visualization Tool is a web-based application designed to provide an interactive, data-driven experience for academic performance analysis. The system will include:

- 3D charts for score trends
- Heatmaps for performance insights
- GPA calculation
- Student & Teacher perspectives
- Secure login and data storage
- Anti-cheating mechanisms to ensure score integrity
- Seamless MongoDB database connectivity for real-time updates

2.2 Product Functionality

- User Login System (Teachers can edit, students can only view)
- 3D Score Visualization (Live updates)
- Heatmap-based Subject Analysis
- GPA Calculation (Based on entered scores)
- Student Profile Management
- Data Storage and Retrieval
- Analytics Dashboard for Teachers
- Anti-Cheating Mechanisms (Track edits, audit logs, role-based access control)
- Database Connectivity and Error Handling

2.3 Design and Implementation Constraints

- Web-Based (Compatible with modern browsers)
- Must support real-time data updates
- Uses secure authentication mechanisms
- Data stored in a scalable database
- Score changes logged and reviewed to prevent cheating
- MongoDB connectivity should be error-free and optimized

2.4 Assumptions and Dependencies

- Requires internet connectivity.
- May depend on third-party libraries for data visualization.
- AI integration is planned but not a primary requirement for the first version.
- Reliable database connection must be maintained for accurate data retrieval.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Intuitive, web-based dashboard.
- Secure login with multi-role support.
- Interactive data visualization and analysis.

3.1.2 Hardware Interfaces

- Cloud-hosted servers for scalability.
- Support for mobile and desktop access.

3.1.3 Software Interfaces

- Django REST API for backend communication.
- PostgreSQL for managing student data.
- AI model integration with TensorFlow/PyTorch.

3.2 Functional Requirements

F1: User Authentication

- The system shall provide a login page for teachers and students.
- Teachers shall have editing permissions, while students shall have view-only access.

F2: 3D Score Visualization

- The system shall display student scores using 3D charts.
- The graphs shall update live as scores are entered.

F3: Heatmap-based Subject Analysis

- The system shall generate heatmaps to indicate strong and weak subjects.
- Colors shall range from green (high performance) to red (low performance).

F4: GPA Calculation

- The system shall automatically calculate GPAs when scores are entered.

F5: Student Details Page

- The system shall have a student profile page displaying performance history.

F6: Analytics Dashboard for Teachers

- Teachers shall have an overview dashboard to compare students' scores.
- The dashboard shall support filters and data export.

F7: Anti-Cheating Mechanism

- The system shall track all edits made to student scores.
- The system shall store an audit log of changes for review.
- Teachers must provide a reason when modifying scores.
- The system shall restrict unauthorized users from modifying marks.

F8: Database Connectivity and Handling

- The system shall maintain a stable connection with MongoDB.
- The system shall handle connection errors gracefully and log them for debugging.

3.3 Use Case Model

3.3.1 Use Case #1 – Student Performance Analysis (U1)

Author – Penumala Vishal

Purpose - Enable administrators and teachers to view student performance analytics through interactive dashboards.

Requirements Traceability – Related to **Functional Requirements 3.2.3 (Visualization Dashboard)** and **3.2.4 (AI Predictions)**.

Priority - High

Preconditions -

- The user must be authenticated as a teacher or administrator.
- Student performance data must exist in the database.

Post conditions –

- The user receives a graphical representation of student performance.
- AI-generated insights and reports are available for download.

Actors – Administrators,Teacher

Flow of Events

1. Basic Flow
    - The user logs into the system.
    - Navigates to the dashboard section.
    - Selects a student or class to view analytics.
    - System fetches and displays data visualization and reports.

2. Alternative Flow
    - If no data exists, the system prompts the user to import student scores.

3. Exceptions - Exceptions that may happen during the execution of the use case
    - Database connectivity issues.
    - Unauthorized access attempts.

3.3.2 Use Case #2 – Student Data Import (U2)

Author – Bollam Siddharth

Purpose - Allow administrators to upload and manage student performance data..

Requirements Traceability – Related to **Functional Requirements 3.2.2 (Data Import).**

Priority - High

Preconditions -

- The user must be an administrator.
- CSV file format must match predefined structure.

Post conditions –

- Student scores are stored in the database.
- The system confirms successful data upload.

Actors – Administrators

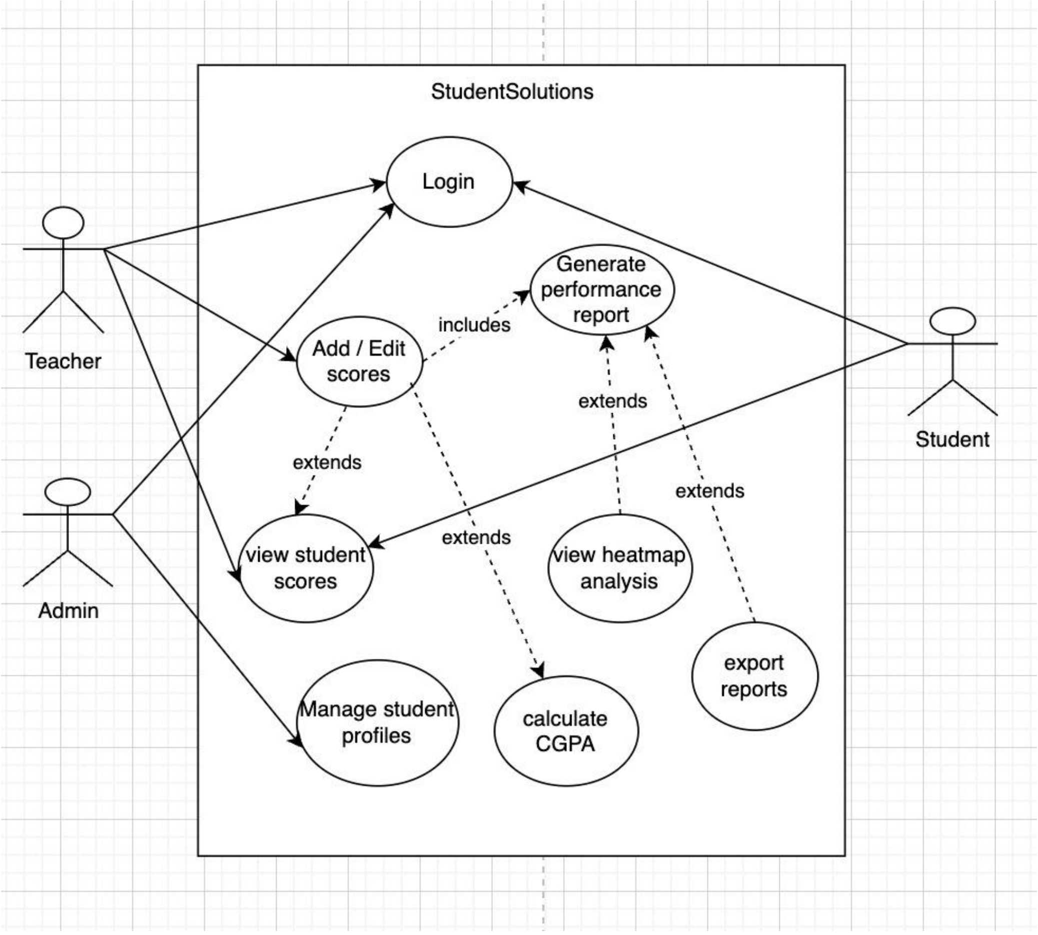
Flow of Events

1. Basic Flow
    - The administrator logs in.
    - Navigates to the data upload section.
    - Uploads a CSV file.
    - System validates and stores the dat.

2. Alternative Flow

- If incorrect data format is detected, the system displays an error message.

3. Exceptions - Exceptions that may happen during the execution of the use case
   - File upload failure due to incorrect format.
   - Database error during data insertion.



4 Other Non-functional Requirements

4.1 Performance Requirements

- The system should handle **100 concurrent users** without significant delay.
- 3D visualization should update within **2 seconds** of data entry.

4.2 Safety and Security Requirements

- Secure login using **hashed passwords**.
- Role-based access control for teachers and students.
- Logs maintained for **all score modifications**.

4.3 Software Quality Attributes

Scalability: Should support future AI integration
Reliability: The system must ensure 99.9% uptime**.**
Maintainability: Clean codebase for ease of future updates.

5 Other Requirements

- Must support **real-time data processing**.
- The UI must be **mobile-friendly and responsive**.

Appendix A – Data Dictionary

Variable	Description	Type
Student_ID	Unique student identifier	Integer
Score	Student's performance score	Float

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>