**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide a comprehensive outline of the functional and non-functional requirements for a Student Management System (SMS). This system aims to streamline the management of student information, course enrollment, and reporting processes within educational institutions, thereby enhancing operational efficiency and data accuracy.

**1.2 Scope**

The SMS will encompass the following key functionalities:

Student Information Management: Capture and maintain detailed records of students.

Course Management: Manage course offerings, including descriptions, schedules, and credit allocations.

Enrollment Management: Facilitate student enrollment in courses and track their academic progress.

Reporting Functions: Generate various reports for administrative purposes, including enrollment statistics and academic performance evaluations.

**2. Functional Requirements**

**2.1 Student Management**

Add, Edit, and Delete Student Records

The system shall allow administrators to create new student profiles with fields for first name, Last name, student ID, date of birth, contact details ( phone number).

Administrators shall be able to edit existing student records to update any relevant information.

The system shall provide functionality to delete student records while ensuring compliance with data retention policies.

**Search Functionality**

The system shall implement a robust search feature that allows users to locate student records using various criteria such as:

Name

Student ID

Student Name

**View Student Details**

Users shall be able to view detailed profiles of students that include:

Personal information (name, date of birth, contact)

Enrollment history (Enrollment date, grades received)

Current course enrollments

Attendance records

**Report Generation**

The system shall enable the generation of reports on:

Total number of students enrolled

Student performance

Enrollment trends over specified periods

**2.2 Course Management**

**Course Management Functions**

Administrators shall have the ability to create new courses with fields for course name, description, credits awarded, prerequisites, and instructor assignments.

The system shall allow editing of existing course details as well as the ability to deactivate or delete courses that are no longer offered.

**View Course Details**

Users shall be able to access detailed information about each course, including:

List of enrolled students with their respective statuses

Student Enrollment in Courses

The system shall provide functionality for enrolling students in courses through a user-friendly interface.

Administrators should be able to manage wait lists for courses that reach capacity.

Course Performance Reports

**The system shall generate reports reflecting:**

Enrollment numbers per course

Average grades per course

Course completion rates

**2.3 User Management**

Administrators shall be able to assign specific roles and permissions based on user responsibilities. Roles may include:

Administrator: Full access to all features.

**Authentication and Authorization**

The system shall implement secure login procedures using encryption protocols.

Multi-factor authentication (MFA) should be available for added security.

**2.4 Report Generation**

**Comprehensive Reporting Features**

The system shall support the generation of various types of reports including but not limited to:

Student enrollment reports by program or semester

Academic performance reports by course or department

Export Functionality

Reports generated by the system should be exportable in multiple formats such as PDF, CSV, and Excel.

Users should have the ability to customize report parameters before generating them.

**3. Non-Functional Requirements**

**3.1 Performance**

The system must be capable of supporting a minimum of 1 concurrent users without performance degradation.

Average response time for any operation should not exceed 60 seconds under normal load conditions.

**3.2 Security**

All sensitive data is encrypted both at rest and in transit using industry-standard encryption protocols (e.g., AES, bcrypt).

User sessions must have timeouts after periods of inactivity to prevent unauthorized access.

Regular security audits must be conducted to identify and mitigate vulnerabilities.

**3.3 Usability**

The user interface must adhere to accessibility standards (e.g., WCAG) to ensure usability for all users.

Contextual help features should be integrated into the interface to assist users in navigating the system effectively.

**3.4 Reliability**

The system must achieve an uptime of at least [Z]% over any given month.

A disaster recovery plan must be established to restore service within [A] hours in case of major failures.

**3.5 Maintainability**

Code should follow best practices for readability and structure (e.g., modular design).

Comprehensive documentation should accompany the codebase, including setup instructions and API documentation for future developers.

**Conclusion**

The School Management System helps minimize the use of paperwork and keeps data safe for the future unless it’s a natural hazard that happens but in most cases data is stored somewhere else as backup, and also simplifies the processes of admitting and enrolling students to the refered courses.