**Software Requirements Specification (SRS)**

**Student Information System (SIS)**

**1. Introduction**

**1.1 Purpose**

This document outlines the requirements for a Student Information System (SIS). The goal of the SIS is to provide an integrated platform that supports the academic lifecycle—from student registration and course enrollment to grade management and attendance tracking. The system is designed to simplify administrative tasks, improve communication among students, teachers, and administrators, and offer timely access to academic information.

**1.2 Scope**

The SIS will be a web-based and mobile-accessible solution that covers:

* Secure student registration and profile management.
* Course enrollment and schedule management.
* Recording and reporting of exam results and academic performance.
* Attendance tracking.
* Communication channels for notifications and messages.
* Integration with other campus systems (e.g., library and finance).

**1.3 Abbreviations**

* **SIS:** Student Information System
* **GPA:** Grade Point Average
* **UI:** User Interface
* **API:** Application Programming Interface

**1.4 References**

* IEEE Standards for Software Requirements Specifications.
* Best practices in academic management systems.
* Existing documentation on campus integration standards.

**1.5 Overview**

This SRS provides a comprehensive look at both the functional and non-functional requirements for the SIS. It covers how users will interact with the system, describes major system features, and sets the stage for subsequent design and implementation efforts.

**2. General Description**

**2.1 Product Perspective**

The SIS is a standalone application that will integrate smoothly with existing campus systems (such as library and financial management systems). It is designed as a central hub for managing academic and administrative processes, ensuring data consistency and ease of access across the institution.

**2.2 Product Functions**

* **User Registration & Management:** Enabling new users to register and existing users to update their profiles.
* **Course Management:** Creating and maintaining course records, enrollment processes, and schedules.
* **Academic Performance Management:** Inputting and processing exam scores, calculating GPA, and generating academic transcripts.
* **Attendance Tracking:** Recording attendance for courses and generating reports.
* **Communication & Notifications:** Sending out announcements, deadlines, exam results, and other critical information.
* **Reporting:** Providing detailed reports on student performance, enrollment statistics, and attendance trends.

**2.3 User Characteristics**

* **Students:** Will use the system to view academic records, enroll in courses, check exam results, and manage personal details.
* **Teachers:** Responsible for entering grades, marking attendance, and managing course-related information.
* **Administrators:** Oversee system operations, manage course and user data, and ensure compliance with institutional policies.

**2.4 Constraints**

* **Platform Compatibility:** Must work on modern web browsers and mobile devices.
* **Data Protection:** Compliance with data privacy laws and institutional data protection policies.
* **Performance:** Capable of handling peak usage without significant delays.
* **Scalability:** Should easily accommodate growing numbers of users as the institution expands.

**3. Specific Requirements**

**3.1 Functional Requirements**

**User Management**

* **Registration:** Users can sign up by providing required details.
* **Authentication:** Secure login mechanism with role-based access (Student, Teacher, Admin).
* **Profile Management:** Users can update their personal and academic information.

**Student Records Management**

* **Record Maintenance:** Create, update, and delete student profiles.
* **Search:** Enable search by student ID, name, or other identifiers.

**Course Management**

* **Course Creation:** Admins can add new courses and update existing ones.
* **Enrollment:** Students can enroll in available courses based on prerequisites and schedule.
* **Schedule Management:** Maintain and display course timings and venues.

**Academic Performance**

* **Exam Score Entry:** Teachers can record exam scores.
* **GPA Calculation:** System calculates GPA automatically from input scores.
* **Transcript Generation:** Provide downloadable academic transcripts for students.

**Attendance Tracking**

* **Recording:** Teachers mark attendance for each class session.
* **Reporting:** Generate attendance summaries for students and courses.

**Communication and Notifications**

* **Announcements:** Broadcast important messages and alerts.
* **Reminders:** Automated notifications for assignment deadlines, exam dates, etc.
* **Messaging:** Option for students and teachers to communicate directly via the system.

**Reporting**

* **Statistical Reports:** Provide insights on enrollment, academic performance, and attendance.
* **Custom Reports:** Allow administrators to generate reports based on specific criteria.

**3.2 Non-Functional Requirements**

* **Performance:** The system should return responses within 2 seconds for typical queries.
* **Security:** Must use encrypted communications (e.g., HTTPS) and robust authentication/authorization methods.
* **Usability:** The user interface should be intuitive, with minimal training required.
* **Scalability:** Should handle an increasing load gracefully as the number of users grows.
* **Reliability:** System downtime must be minimized; regular backups and a disaster recovery plan are essential.
* **Maintainability:** The codebase should be well-documented to facilitate future updates and maintenance.

**3.3 System Features**

* **Dashboard:** A central hub providing snapshots of academic progress, upcoming deadlines, and alerts.
* **Calendar Integration:** Sync course schedules and exam dates with personal calendars.
* **Multi-Platform Access:** Accessible via web browsers and mobile devices with a consistent user experience.
* **Inter-System Communication:** API endpoints to integrate with other campus management systems.
* **Customization:** Ability for institutions to tailor features and reporting formats to their specific needs.

**4. Appendices**

* **A. Data Flow Diagrams (DFD):** Visual representations of the system processes and data exchanges.
* **B. Use Case Diagrams:** Illustrations of user interactions with the system.
* **C. Database Schematics:** Proposed database design and entity relationships.
* **D. UI Wireframes:** Preliminary sketches of the user interface.
* **E. Glossary of Terms:** Definitions of key terms used within the document.