**Software Requirements Specification (SRS) for Student Information System (SIS)**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to define the software requirements for a web-based Student Information System (SIS). The system will manage student data, including personal details, academic records, course enrollment, and administrative functionalities.

**1.2 Scope**

The SIS is a web-based application designed for educational institutions to manage student information efficiently. The system will allow administrators, faculty, and students to access and update information in real-time. Key functionalities include student registration, grade management, attendance tracking, and reporting.

**1.3 Definitions, Acronyms, and Abbreviations**

* SIS: Student Information System
* UI: User Interface
* DBMS: Database Management System
* HTTP: HyperText Transfer Protocol
* CRUD: Create, Read, Update, Delete

**1.4 References**

* IEEE Standard for Software Requirements Specifications (IEEE 830-1998)
* Institutional guidelines on student record management

**1.5 Overview**

This document provides an overview of the system requirements, including functional and non-functional requirements, system architecture, and constraints.

**2. General Description**

**2.1 Product Perspective**

The SIS is a standalone web application that integrates with existing institutional databases and services. It will be accessible via web browsers and support multiple user roles.

**2.2 Product Functions**

* Student Registration and Profile Management
* Course Enrollment and Management
* Academic Performance and Grade Tracking
* Attendance Management
* Fee and Payment Management
* Reporting and Analytics

**2.3 User Characteristics**

* **Administrators**: Manage system settings, user roles, and reports.
* **Faculty**: Manage courses, update grades, and track attendance.
* **Students**: View personal information, enroll in courses, and check academic performance.

**2.4 Constraints**

* Web-based application accessible via modern browsers.
* Secure authentication and role-based access control.
* Compliance with institutional and legal data protection policies.

**2.5 Assumptions and Dependencies**

* The system will use MySQL or PostgreSQL for database management.
* Users must have an active internet connection.
* Integration with institutional email services for notifications.

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 User Authentication and Authorization**

* The system shall allow users to log in using unique credentials.
* The system shall enforce role-based access control.

**3.1.2 Student Management**

* The system shall allow administrators to add, edit, and delete student records.
* Students shall be able to update their personal details (except critical data like Student ID).

**3.1.3 Course Management**

* Faculty members shall be able to create and manage course details.
* Students shall be able to enroll in courses through the system.

**3.1.4 Grade and Attendance Tracking**

* Faculty members shall be able to input grades for students.
* The system shall generate automated attendance reports.

**3.1.5 Reporting and Analytics**

* The system shall generate performance reports based on academic data.
* Administrators shall be able to generate and export reports in PDF format.

**3.2 Non-Functional Requirements**

* **Security**: The system shall implement encryption for sensitive data.
* **Performance**: The system shall support concurrent access by at least 1000 users.
* **Usability**: The system shall provide an intuitive and user-friendly interface.
* **Availability**: The system shall have an uptime of 99.9%.
* **Scalability**: The system shall be scalable to accommodate growing student data.

**3.3 External Interface Requirements**

**3.3.1 User Interface**

* The system shall have a responsive web UI.
* The UI shall provide easy navigation with minimal learning curve.

**3.3.2 Hardware Interfaces**

* The system shall support deployment on cloud-based or on-premise servers.

**3.3.3 Software Interfaces**

* The system shall integrate with institutional databases.
* The system shall support RESTful APIs for third-party integration.

**4. System Models**

**4.1 Use Case Diagram**

* **Actors**: Student, Faculty, Administrator
* **Use Cases**: Register, Login, Enroll in Course, View Grades, Manage Students, Generate Reports

**4.2 Data Flow Diagram (DFD)**

* **Level 0**: User Inputs → Processing → Database → Output
* **Level 1**: Student Enrollment, Course Management, Grade Tracking

**5. Appendices**

* Institutional policies on student data privacy
* Compliance guidelines for web application security

**End of Document**