**EX NO: 2 DOCUMENT THE SOFTWARE REQUIREMENTS SPECIFICATION**

**DATE: 31.01.2025 (SRS) FOR THE IDENTIFIED SYSTEM.**

**AIM :**

**What is Software Requirement Specification(SRS)?**

SRS (Software Requirements Specification) is a formal document that defines the functional and non-functional requirements of a software system. It serves as a blueprint for development, ensuring all stakeholders understand the project scope, design, and expectations.

**Key Components of SRS :**

1. **Introduction** – Purpose, scope, definitions, and references.
2. **Overall Description** – Product perspective, user characteristics, and constraints.
3. **Functional Requirement** - Defines what the system should do (e.g., user login, student record management).
4. **Non-Functional Requirement** - Defines system qualities (e.g., performance, security, reliability).
5. **External Interface Requirement** - Specifies interactions with hardware, software, and communication interfaces.

**1. INTRODUCTION**

**1.1 Purpose**

This document defines the complete software requirements for the Online Student Information Management System (OSIMS). It details the system’s design, architecture, and functional components, providing necessary information for software support. OSIMS aims to replace manual student management, improving efficiency in handling student records, academic details, and administrative processes.

**1.2 Scope**

OSIMS is developed as a web-based platform to automate student record management, replacing outdated manual systems like PUMS. The system improves efficiency by:

* Storing and updating student details.
* Managing academic and administrative records.
* Providing result access for parents and students.
* Supporting report generation for faculty and administration.
* Offering a flexible mechanism for modifying student information.

**1.3 Environmental Characteristics**

The Online Student Information Management System (OSIMS) operates in a multi-user networked environment, accessible via web browsers over TCP/IP. It supports both online (cloud-based) and offline (local server-based) operations, ensuring accessibility and data integrity. The system is designed to function efficiently under various environmental conditions, including:

* **Operating System** - Windows, Linux, or macOS
* **Hardware Requirements** - Minimum 512MB RAM, 500MB disk space
* **Network Requirements:** - Stable internet connection for online access
* **Browser Compatibility:** Supports Chrome, Firefox, Edge, and Safari
* **Power Requirements:** Can operate on standard computing devices with UPS backup support for uninterrupted usage

**2. OVERALL DESCRIPTION**

**2.1 Product Perspective**

The **Online Student Information Management System (OSIMS)** is a **client-server** application designed to replace manual student record management systems. It integrates with existing administrative systems and databases to provide an automated, efficient, and user-friendly environment for handling student-related information. The system supports:

1. **Multi-user access** with role-based permissions.
2. **Centralized database** for secure data storage and retrieval.
3. **Web-based interface** for accessibility from any location.
4. **Integration** with existing institutional infrastructure.

**2.2 Product Features**

**The key features of OSIMS include:**

1. **Secure Login & Authentication** – Role-based access control for administrators, faculty, and students.
2. **Student Management** – Maintain student personal, academic, and administrative records.
3. **Course & Program Management** – Manage courses, schemes, and faculty assignments.
4. **Result Processing** – Automated result generation and publication.
5. **User Profile Management** – Students and faculty can update personal details.
6. **Reporting & Analytics** – Generate performance reports and statistical analysis.

**2.3 User Classes and Characteristics**

OSIMS is designed for various user classes, each with specific roles and access levels:

1. **System Administrator** – Manages users, system settings, and database integrity.
2. **Faculty Members** – Manage student records, update grades, and generate reports.
3. **Students** – View academic progress, register for courses, and update personal details.
4. **Parents/Guardians** – Access student results and academic performance reports.
5. **Guests/Public Users** – View general institutional information like events and schedules.

**2.4 Operating Environment**

OSIMS is designed to run in **multi-user environments** with the following specifications:

1. **Operating Systems:** Windows, Linux, macOS
2. **Database Management System:** MS SQL Server 2000 / MySQL / PostgreSQL
3. **Programming Language:** Java (Backend), Visual Basic 6.0 (Frontend)
4. **Development Platform:** Eclipse IDE
5. **Web Server:** Apache Tomcat / IIS
6. **Network Requirements:** TCP/IP-based connectivity for online access

**2.5 Design and Implementation Constraints**

1. The system must support at least **100 concurrent users** without performance degradation.
2. Database interactions should be optimized for **high-speed transactions**.
3. The system should follow **industry-standard encryption** for sensitive data.
4. Only the **administrator** has the privilege to delete records.
5. The application should be **scalable** to accommodate future expansions.
6. The interface must be **user-friendly** and support **multi-language options** if needed.

**2.6 User Documentation**

OSIMS will provide the following user documentation:

1. **User Manual** – Instructions for students, faculty, and administrators.
2. **Administrator Guide** – System setup, maintenance, and troubleshooting steps.
3. **Installation Guide** – Detailed steps for software installation and configuration.
4. **FAQs & Help Section** – Common troubleshooting solutions.
5. **Online Support** – Web-based customer support portal for issue resolution.

**3. FUNCTIONAL REQUIREMENTS**

**Functional Requirements by User Class**

**fn R1: User Account Management**

* **R1.1:** Create, update, and delete user accounts.
  + **R1.1.1:** Create new user accounts (admin, faculty, students).
  + **R1.1.2:** Update user account credentials and roles.
  + **R1.1.3:** Delete user accounts when necessary.

**fn R2: Student, Faculty, and Course Management**

* **R2.1:** Add, update, and remove student, faculty, and course details.
  + **R2.1.1:** Add new students, faculty members, and courses.
  + **R2.1.2:** Update existing student, faculty, and course records.
  + **R2.1.3:** Remove outdated or irrelevant records.

**fn R3: System Configuration and Maintenance**

* **R3.1:** Configure system settings and user access control.
  + **R3.1.1:** Set permissions for different roles (admin, faculty, student).
  + **R3.1.2:** Configure system settings (email, notifications, etc.).
* **R3.2:** Backup and restore the database.
  + **R3.2.1:** Schedule and perform database backups.
  + **R3.2.2:** Restore database from backups when necessary.

**fn R4: Report Generation**

* **R4.1:** Generate student performance, attendance, and administrative reports.
  + **R4.1.1:** Generate academic performance reports for students.
  + **R4.1.2:** Generate attendance records for students.
  + **R4.1.3:** Generate administrative records such as user activity.

**fn R5: Security Management**

* **R5.1:** Implement user authentication policies.
  + **R5.1.1:** Require strong password policies for all users.
  + **R5.1.2:** Enforce session timeouts after inactivity.
* **R5.2:** Enforce role-based access control.
  + **R5.2.1:** Assign different levels of access to admin, faculty, and students.

**fn R6: Course Registration**

* **R6.1:** Register for courses each semester.
  + **R6.1.1:** Select and register courses for the upcoming semester.
* **R6.2:** View course schedules.
  + **R6.2.1:** Check schedules for registered courses.

**4. NON - FUNCTIONAL REQUIREMENT**

**Non-FN N1: Database**

**N1.1:** The database must support efficient storage and retrieval of data.

* **N1.1.1:** The system must support **relational databases** such as MS SQL Server or MySQL.
* **N1.1.2:** Database should allow for real-time querying and data manipulation.
* **N1.1.3:** The system should handle at least **100 concurrent users** without performance degradation.

**N1.2:** Database backup and restore should be supported.

* **N1.2.1:** Regular backups must be taken daily.
* **N1.2.2:** Backup data should be stored securely in a separate location.

**Non-FN N2: Performance**

**N2.1:** The system should be capable of processing 95% of transactions in under 1 second.

* **N2.1.1:** System should handle **concurrent logins** and actions without delay.

**N2.2:** The system should maintain **99.9% uptime** for availability.

* **N2.2.1:** The system should be available 24/7, except during scheduled maintenance.

**Non-FN N3: Security**

**N3.1:** Strong encryption should be used for all sensitive data.

* **N3.1.1:** User passwords should be stored in **hashed** format.
* **N3.1.2:** Secure **SSL/TLS encryption** should be used for all data transmission.

**N3.2:** Role-based access control should be implemented.

* **N3.2.1:** Administrators, faculty, and students should have clearly defined access levels.

**Non-FN N4: Usability**

**N4.1:** The system must be **user-friendly** and easy to navigate.

* **N4.1.1:** Provide clear instructions and tooltips for all users.
* **N4.1.2:** Support multiple languages for diverse user demographics.

**5.EXTERNAL INTERFACE REQUIREMENT**

**5.1 User Interfaces**

The OSIMS system will provide the following user interfaces:

* **Login Page:** A secure authentication portal with username and password fields.
* **Dashboard:** Role-based dashboard with access to student records, course details, and reports.
* **Student Management Interface:** Forms for adding, updating, and deleting student details.
* **Course Management Interface:** Interface for faculty to assign courses and update student performance.
* **Reports Section:** Interactive charts and downloadable reports for student performance analysis.
* **Settings Page**: System configuration settings including user roles, access permissions, and preferences.

**5.2 Hardware Interfaces**

1. Minimum **640x480 screen resolution**
2. Support for **dot matrix, deskjet, and laserjet printers**
3. Network-enabled systems for **multi-user access**

**5.3 Software Interfaces**

1. **Operating System:** Windows, Linux, macOS
2. **Frontend:** Microsoft Visual Basic 6.0
3. **Backend:** MS SQL Server 2000 / MySQL / PostgreSQL
4. **Platform:** Java (Backend)
5. **IDE:** Eclipse

**5.4 Communication Interfaces**

1. All data transmissions will use **TCP/IP protocols** for secure communication.
2. System must support **HTTPS for encrypted transactions**.

**RESULT:**