Software Requirements Specification (SRS)

School Management System

1. Introduction

1.1 Purpose

The purpose of the School Management System (SMS) is to provide an efficient and centralized platform for managing academic and administrative activities in a school. The system allows administrators, teachers, students, and parents to interact with the platform to manage attendance, grades, fees, schedules, and communication. This document outlines the requirements and specifications for the development of the system.

1.2 Document Conventions

- **Abbreviations:**
  - SMS: School Management System
  - UI: User Interface
  - DBMS: Database Management System
- **Standards:**
  - The document follows IEEE standards for SRS documentation.

1.3 Intended Audience and Reading Suggestions

- **Developers & Testers:** Focus on Sections 3 & 4 for functional and non-functional requirements.
- **Project Managers:** Review Sections 1 & 2 for scope and objectives.

- **School Administrators & Stakeholders:** Read Sections 2 & 3 for features and user roles.

1.4 Project Scope

The School Management System is a **web-based application** designed to automate daily school operations. It provides modules for managing students, teachers, attendance, grades, fees, and reports. The system aims to:

- Improve efficiency in managing school operations.
- Reduce paperwork and manual errors.
- Provide easy access to academic data for students and parents.
- Enhance communication between teachers, students, and parents.

1.5 References

- PHP & MySQL documentation
- XAMPP documentation
- Bootstrap / Tailwind CSS documentation

2. Overall Description

2.1 Product Perspective

The School Management System is a standalone web application developed using PHP, MySQL, HTML, CSS, and JavaScript. It will be deployed on a local server (XAMPP/WAMP) or a web hosting platform.

2.2 Product Features

- **Login System:** Role-based access (Admin, Teacher, Student, Parent).
- **Student Management:** Add, update, and view student details.

- **Teacher Management:** Manage teacher profiles and schedules.
- **Attendance System:** Record and track daily attendance.
- **Examination & Results:** Upload, update, and view marksheets.
- **Fee Management:** Track fee payments and generate receipts.
- **Reports:** Generate reports for attendance, performance, and fees.

2.3 User Classes and Characteristics

- **Admin:** Manages the whole system, users, and reports.
- **Teachers:** Manage student attendance, grades, and class schedules.
- **Students:** View personal details, grades, attendance, and fee status.
- **Parents:** Monitor student progress, fees, and attendance.

2.4 Operating Environment

- **Frontend:** HTML, CSS, JavaScript, Bootstrap/Tailwind
- **Backend:** PHP
- **Database:** MySQL
- **Server:** Apache (XAMPP/WAMP/LAMP)
- **Devices:** Desktop & mobile-friendly browsers

2.5 Design and Implementation Constraints

- Must run on a local server or cloud hosting with PHP & MySQL support.
- Responsive UI for mobile devices.

- Secure login and database protection.

2.6 User Documentation

- Online help (FAQs).
- User manual for administrators, teachers, and students.

2.7 Assumptions and Dependencies

- Users have internet access.
- The school provides correct student/teacher data.
- The system depends on PHP and MySQL compatibility.

3. System Features

3.1 User Authentication

- **Priority:** High
- **Description:** Secure role-based login for all users.
- **Functional Requirements:**
  - Users log in with username and password.
  - Different dashboards for Admin, Teacher, Student, Parent.

3.2 Student Management

- **Priority:** High
- **Description:** Manage student profiles.

- **Functional Requirements:**
    - Add, edit, delete student details.
    - View student list.

3.3 Attendance Management

- **Priority:** High
- **Description:** Record and track attendance.
- **Functional Requirements:**
    - Teachers mark attendance daily.
    - Students/Parents can view attendance history.

3.4 Examination & Result Management

- **Priority:** High
- **Description:** Manage exam schedules and results.
- **Functional Requirements:**
    - Teachers upload marks.
    - Students and parents view results.

3.5 Fee Management

- **Priority:** Medium
- **Description:** Track and manage fees.
- **Functional Requirements:**

    - Admin records fee payments.
    - Generate receipts and pending fee reports.

3.6 Reports Generation

- **Priority:** Medium
- **Description:** Generate detailed reports.
- **Functional Requirements:**
    - Attendance reports.
    - Student performance reports.
    - Fee payment reports.

4. External Interface Requirements

4.1 User Interfaces

- Responsive UI with dashboards.
- Forms for data entry.
- Search and filter options.

4.2 Hardware Interfaces

- Supports standard desktops and laptops.
- Mobile responsive for students/parents.

4.3 Software Interfaces

- MySQL Database.
- PHP-based server-side logic.

4.4 Communication Interfaces

- HTTPS for secure connections (if deployed online).

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Fast response time (<2 seconds).
- Support for 100+ concurrent users.

5.2 Safety Requirements

- Daily backup of database.
- Data recovery in case of system failure.

5.3 Security Requirements

- Password encryption.
- Role-based access control.
- Protection against SQL Injection.

5.4 Software Quality Attributes

- Maintainable code (modular PHP).
- User-friendly design.
- Scalability for future modules.

6. Other Requirements

- Notifications for attendance/fees.
- Parent-teacher communication module (future enhancement).

Appendices

A. Glossary

- **Attendance Management:** Process of tracking student presence.
- **Responsive Design:** UI that adapts to desktop/mobile.

B. Analysis Models

- Use case diagrams (Login, Attendance, Results).
- Sequence diagrams for fee payment and report generation.

C. Issues List

- Pending: Finalize hosting option.
- Resolved: Database schema design.