School Management System

Requirements and specification

Vraj Patel - 249545250
Sandeep Kaur - 239685880
Asmi Patel - 239543330
Yash Desai - 239541990

October 2nd, 2024

COSC/ITEC 3506 001X
Amandeep S. Patti

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this **Software Requirements Specification (SRS)** is to define the functional and non-functional requirements for the **School Management System (SMS)**. The system aims to provide efficient management of school operations, including student records, attendance, teacher information, fee tracking, and academic performance.

This document serves as a guide for developers, project managers, testers, and other stakeholders to ensure the system is designed and implemented according to the requirements.

- **Audience**: Developers, Testers, Project Managers, End-users (School Staff, Administrators, Teachers).
- This document is intended for a **new system** for managing school activities and processes.

1.2 Scope

The School Management System (SMS) is designed to automate administrative tasks related to school management, such as student registration, attendance tracking, fee management, and reporting. The system will be used by school administrators, teachers, and staff for daily activities and decision-making.

- **Example Scope**: "The system will allow users to manage student registrations and attendance but will not include online payment integration in the initial release."

1.3 Definitions, Acronyms, and Abbreviations

This section defines key terms used in the document:

- **CRUD**: Create, Read, Update, Delete.
- **API**: Application Programming Interface.
- **SMS**: School Management System.
- **Admin**: Administrator of the system who has full access to all modules.

1.4 Overview

This document is structured as follows:

- Section 2: Overall description of the system's environment and interactions.
- Section 3: Detailed functional requirements.
- Section 4: Interface descriptions (user, hardware, software).
- Section 5: Use case models showing user interactions.
- Section 6: Non-functional requirements including performance, security, and scalability.
- Section 7: Additional legal and data retention requirements.
- Section 8: Appendices for supporting materials.

2. Overall Description

2.1 System Perspective

The School Management System (SMS) will operate within a school's administrative environment and interact with external systems like grading tools and communication software. The system fits within the school's infrastructure, supporting teachers, students, and administrators in managing day-to-day operations.

- **System Context**: The system will interact with the following external systems:
  - **Grading Software**: For managing student academic performance.
  - **Communication Tools**: For sending notifications to parents and students.

2.2 System Functions

At a high level, the system will provide functionalities such as student registration, attendance tracking, fee management, and report generation.

- **Example**: The system will allow administrators to view and update student records, teachers to mark attendance, and parents to view their child’s performance.

2.3 User Characteristics

The primary users of the system are:

- **Administrator**: Responsible for managing the system configuration and overseeing school operations.
- **Teacher**: Manages student attendance, academic performance, and communication with parents.
- **Staff**: Handles administrative tasks like student registration and fee collection.
- **Parents/Students**: Can view academic records and attendance online.

2.4 Constraints

The system must comply with local educational regulations and support a high volume of concurrent users, especially during exam periods.

- **Example**: The system must handle 1,000 concurrent users and comply with data privacy laws.

2.5 Assumptions and Dependencies

The system assumes stable network connectivity for real-time updates and depends on third-party software for report generation and communication with parents.

3. System Features (Functional Requirements)

3.1 Feature Description

Feature 1: User Authentication

- **Description**: The system will allow users (administrators, teachers, parents) to create accounts, log in, and reset passwords.
- **Inputs**: Username, password.
- **Processes**: Authenticate users with encrypted credentials.
- **Outputs**: Authentication token, success or failure messages.

Feature 2: Student Registration

- **Description**: The system will enable administrators to register new students by entering personal information, class assignments, and parent/guardian contact details.
- **Inputs**: Student details (name, age, grade, address, parent/guardian details).
- **Processes**: Validate and store student information in the database.
- **Outputs**: Confirmation of successful registration or error messages.

Feature 3: Attendance Management

- **Description**: The system will allow teachers to take and track student attendance on a daily basis, view attendance history, and generate attendance reports.
- **Inputs**: Student ID, date, and attendance status (present/absent).
- **Processes**: Record attendance data and generate summary reports.
- **Outputs**: Daily attendance records, attendance reports, warnings for absenteeism.

Feature 4: Fee Management

- **Description**: The system will allow school staff to manage student fee payments, generate invoices, and track payment statuses.
- **Inputs**: Student ID, fee amount, payment details (method, date).
- **Processes**: Record fee payments and generate financial reports.
- **Outputs**: Payment confirmation, outstanding fee reports, invoice receipts.

Feature 5: Report Generation

- **Description**: The system will generate various reports, including student academic performance, attendance records, and financial summaries.
- **Inputs**: Report criteria (e.g., date range, class, student ID).
- **Processes**: Query the database based on criteria and generate reports in PDF format.
- **Outputs**: PDF reports for printing or sharing.

3.2 Use Case Diagram Integration

For each system feature, use case diagrams are provided to show interactions between actors (users) and the system.

- **Example**: In the "Attendance Management" feature, a use case diagram shows a teacher (actor) interacting with the system to mark student attendance.
- **Relationships**: Use cases such as “Mark Attendance” and “Generate Attendance Report” are linked with the teacher role through association.

4. External Interface Requirements

4.1 User Interfaces

The system must have a user-friendly interface, with different screens designed for administrators, teachers, and parents.

- **Example**: The login page will have fields for username and password, along with "Forgot Password" and "Create Account" links.

4.2 Hardware Interfaces

The system will interact with hardware such as barcode scanners for managing student IDs and biometric systems for attendance tracking.

- **Example**: "The system will use biometric attendance data to update the attendance records."

4.3 Software Interfaces

The School Management System will integrate with other systems such as:

- **Grading Software**: For managing and importing student grades.
- **API Integration**: Will include API communication with third-party systems for reporting and communication tools.

4.4 Communication Interfaces

The system will use secure communication protocols, such as HTTPS, for transmitting sensitive data between clients (teachers, parents) and the server.

- **Example**: "Data will be exchanged in JSON format for web-based APIs, and email notifications will be sent using SMTP."

5. System Models (Use Case Diagrams)

5.1 Use Case Diagram Guidelines

Use case diagrams illustrate how actors (users) interact with the system. Each actor will interact with various functionalities within the system, as detailed in the previous sections.

- **Actors**: Admin, Teacher, Parent.
- **Use Cases**: Examples include "Login," "Mark Attendance," "Generate Report," "Manage Fees."

5.2 Example of a Use Case Diagram

- **Actors**: Admin, Teacher, Parent.
- **Use Cases**: The "Admin" can "Manage Students," "Generate Reports," and "Track Fees." The "Teacher" can "Mark Attendance" and "View Reports." Parents can "View Student Performance."

6. Non-Functional Requirements

6.1 Performance Requirements

- The system must handle up to **1,000 concurrent users** without performance degradation.
- Response times must be less than **2 seconds** for all user actions.

6.2 Security Requirements

- All data transfers must be encrypted using **SSL/TLS**.
- Passwords must be stored using a **secure hashing algorithm**.
- Only authorized users can access sensitive student and financial data.

6.3 Usability Requirements

- The system must be easy to navigate for users with basic technical skills (teachers, parents).
- The user interface must comply with **WCAG 2.1 accessibility standards**.

6.4 Reliability and Availability

- The system should maintain **99.9% uptime**, ensuring availability during school hours.
- Any critical failures must be resolved within a **few hours**.

6.5 Maintainability

- The system will be modular, allowing individual components to be updated without downtime for the entire system.
- All source code will be documented to facilitate future updates.

6.6 Scalability

- The system must be scalable to accommodate a growing student population and increased administrative workloads.
- **Cloud hosting** will be used to ensure that the system can handle future growth, scaling up to thousand students and staff members.

7. Other Requirements

7.1 Legal and Regulatory Requirements

- The system must comply with **local educational laws** and **data privacy regulations** such as the **GDPR**.
- Student data must be stored in compliance with **FERPA** regulations.

7.2 Scalability and Future-Proofing

- The system's architecture must support **horizontal scaling** to allow future system expansions.
- The system will support multi-language interfaces to cater to a broader user base.

7.3 Data Retention and Archiving

- Student records and academic performance data will be stored for a minimum of **7 years**.

- Old data must be archived securely, with provisions for retrieval when necessary.

8. Appendices

Appendix A: Glossary

- **CRUD**: Create, Read, Update, Delete operations.
- **API**: Application Programming Interface.
- **FERPA**: Family Educational Rights and Privacy Act.
- **GDPR**: General Data Protection Regulation.

Appendix B: System Context Diagram

- A detailed system context diagram showing the interaction between the School Management System and external entities, such as grading software and parent communication tools.

Appendix C: Use Case Diagrams

- Visual representations of use cases showing interactions between users (actors) and system functionalities like "Student Registration" and "Attendance Management."

9. Conclusion

This Software Requirements Specification (SRS) outlines the detailed functional and nonfunctional requirements for the **School Management System**. It is intended to serve as a guide for developers, testers, and other stakeholders involved in the system's design, development, and implementation. The system is designed to be scalable, secure, and user-friendly, ensuring it meets the needs of modern educational institutions.

The School Management System is designed to address key challenges faced by educational institutions, such as streamlining administrative tasks, automating processes, and providing a centralized platform for managing student data, attendance, and academic records. The system will feature robust functionalities like user authentication, role-based access control, student and staff management, timetable scheduling, attendance tracking, fee management, and reporting. It ensures ease of use

for all user types, from school administrators to teachers, students, and parents.

Key Benefits:

- **Scalability**: The system architecture allows for scalability, ensuring that it can support a growing number of users, whether it's a small institution or a large educational network.
- **Security**: With strong emphasis on security, including encryption of sensitive data and compliance with modern data protection standards, the system will protect personal information and student records.
- **User-Friendly Interface**: Designed with usability in mind, the SMS will provide an intuitive interface that enhances the experience for all users, reducing training needs and improving overall satisfaction.
- **Efficiency**: By automating routine administrative tasks and enabling real-time access to data, the system will improve the overall efficiency of school operations, freeing up resources for more critical educational activities.
- **Compliance**: The system will adhere to relevant legal and regulatory requirements, such as data privacy laws, ensuring compliance with modern standards for data handling and storage.