

---

# **Software Requirements Specification**

**for**

# **Private School Management Website**

**Prepared by Kavishka**

**12/5/2024**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Project Scope .....	1
1.5 References.....	1
<b>2. Overall Description .....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Features .....	2
2.3 User Classes and Characteristics .....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints .....	3
2.6 User Documentation .....	3
2.7 Assumptions and Dependencies .....	3
<b>3. External Interface Requirements .....</b>	<b>3</b>
3.1 User Interfaces .....	3
3.2 Hardware Interfaces .....	3
3.3 Software Interfaces .....	4
3.4 Communications Interfaces .....	4
<b>4. Other Nonfunctional Requirements.....</b>	<b>4</b>
4.1 Performance Requirements .....	4
4.2 Safety Requirements .....	4
4.3 Security Requirements .....	4
4.4 Software Quality Attributes .....	4

# 1. Introduction

## 1.1 Purpose

The purpose of this system is to create a private school management web application that allows administrators to manage courses and teacher details while enabling users to view class details and apply for classes. The admin panel is restricted to authorized personnel only, ensuring data integrity and secure access.

## 1.2 Document Conventions

- **Font Style:** Times New Roman, 12pt
- **Headings:** Bold, 14pt
- **References:** IEEE citation style
- **Figures/Tables:** Numbered sequentially (e.g., Figure 1.1, Table 2.1)

## 1.3 Intended Audience and Reading Suggestions

- **Audience:**
  - Developers: To understand the system's requirements and constraints.
  - Testers: To design and execute test cases.
  - Administrators: For understanding system capabilities and limitations.
  - End-Users: To ensure usability.
- **Suggestions:**
  - Developers: Focus on sections 3, 4, and 5.
  - Testers: Focus on sections 3.1 and 5.

## 1.4 Project Scope

This project covers a private school management system with two main components:

1. A **website** where users can view course and teacher details and apply for classes.
2. An **admin panel** for managing courses and teacher information, accessible only to authorized administrators.

## 1.5 References

1. IEEE Standards for Software Requirements Specifications.

2. W3C Standards for Web Development (HTML, CSS, JavaScript).
3. PHP Documentation (<https://www.php.net/docs.php>).
4. MySQL Documentation (<https://dev.mysql.com/doc/>).

## 2. Overall Description

### 2.1 Product Perspective

This application serves as a comprehensive school management platform that integrates course and teacher management with a user-friendly web interface for students to apply for classes. It replaces traditional manual systems, improving efficiency and accessibility.

### 2.2 Product Features

1. Admin Panel:
  - Add, edit, and delete course details.
  - Add, edit, and delete teacher information.
  - View user-submitted applications.
2. Website:
  - Display course and teacher details.
  - Allow students to submit applications for classes.

### 2.3 User Classes and Characteristics

1. **Administrators:**
  - Technical knowledge required.
  - Responsible for managing the backend operations.
2. **Students/Users:**
  - Basic knowledge of web browsing.
  - Access the website to view and apply for courses.

### 2.4 Operating Environment

1. **Client-Side:**
  - Devices: PC, Laptop, Mobile, and Tablets.
  - Browser: Chrome, Firefox, Edge.

## 2. Server-Side:

- PHP-based server with a MySQL database.

## 2.5 Design and Implementation Constraints

- Must be built using **HTML, CSS, JavaScript, and PHP**.
- MySQL is required for database management.
- Responsive design for compatibility with all screen sizes.

## 2.6 User Documentation

- **User Manual:** Guides for users to apply for classes.
- **Admin Manual:** Instructions for administrators to manage courses and teacher information.

## 2.7 Assumptions and Dependencies

- Assumes users have internet access.
- Depends on hosting infrastructure for availability.
- Relies on accurate inputs from administrators for displaying course and teacher details.

# 3. External Interface Requirements

## 3.1 User Interfaces

### 1. Website:

- Home Page for navigation.
- Detailed pages for courses and teachers.
- Application form for students.

### 2. Admin Panel:

- Dashboard with course and teacher management options.
- Login authentication.

## 3.2 Hardware Interfaces

- **Client Devices:**
  - Minimum Requirements: 4 GB RAM, 1 GHz processor, internet-enabled device.
- **Server Requirements:**

- PHP-enabled web server (e.g., Apache).
- MySQL-compatible database server.

### **3.3 Software Interfaces**

- **Frontend:** HTML, CSS, JavaScript.
- **Backend:** PHP.
- **Database:** MySQL.

### **3.4 Communications Interfaces**

- **HTTPS Protocol:** To secure data exchange.
- **SMTP:** For email notifications.

## **4. Other Nonfunctional Requirements**

### **4.1 Performance Requirements**

- Pages should load within 3 seconds.
- The system should support 50 concurrent users.

### **4.2 Safety Requirements**

- Daily backups of database data.
- Graceful error handling with descriptive messages.

### **4.3 Security Requirements**

- Admin panel secured by authentication.
- Data encrypted during transmission using HTTPS.

### **4.4 Software Quality Attributes**

- **Usability:** Simple and intuitive UI.
- **Maintainability:** Clean, modular code for easier updates.
- **Scalability:** Designed to support additional features and higher user loads.