Software Requirements Specification

for

Attendance Tracking System for Students

Prepared by

Aadhi Pranesh S S(23CSR001)  
Abhinav Krishna B(23CSR005)  
Dheepisha G(23CSR050)

Kongu Engineering College  
  
26/02/2025

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience 1

1.3 Project Scope 1

1.4 Definitions,Acronyms and Abbreviations 1

1.5 References 1

2. Overall Description 2

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 2

3. System Features 3

3.1 User Authentication 3

3.2 Product Listing and Aproval Workflow 3

3.3 Shopping Cart and Checkout Flow 3

3.4 Payment Gateway and Order Tracking 3

3.5 Review and Feedback System 4

3.6 Seller Dashboard 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Security Requirements 5

5.3 Usability Requirements 5

5.4 Scalability Requirements 5

6. Other Requirements 5

# Introduction

## Purpose

The purpose of this **Software Requirements Specification (SRS)** is to define the functional and non-functional requirements for an **Attendance Tracking System for Students**. The system allows faculty members to record student attendance, students to track their attendance records, and to manage attendance-related reports efficiently.

## Intended Audience

1. **Faculty members**: To log student attendance.
2. **Students**: To view attendance records and track their progress.

## Project Scope

This **Attendance Tracking System** provides an efficient digital method for recording and monitoring student attendance. The key functionalities include:

1. Faculty marking attendance.
2. Students tracking their attendance records.
3. Generating reports on student attendance.
4. Viewing attendance trends using graphical representations.
5. Providing notifications or warnings for low attendance.

## Definitions, Acronyms, and Abbreviations

1. **Faculty**: Teaching staff responsible for marking attendance.
2. **Student**: A user who can view their attendance records.
3. **API**: Application Programming Interface.

**1.5 References**

* MongoDB Documentation: <https://www.mongodb.com/docs/>
* React Documentation: <https://reactjs.org/docs/>
* Node.js Documentation: <https://nodejs.org/en/docs/>

# Overall Description

## Product Perspective

This system is a **web application** that allows faculty to mark attendance digitally and students to track their attendance records. The frontend will be built using **ReactJS**, the backend will use **Node.js with Express.js**, and data will be stored in **MongoDB**.

## Product Features

1. **User Authentication** - Faculty and students can log in securely.
2. **Faculty Attendance Logging** - Faculty members can mark student attendance.
3. **Student Attendance Tracking** - Students can view their daily and monthly attendance.
4. **Attendance Reports** - Generate and download attendance reports.
5. **Graphical Attendance Overview** - Visual graphs for tracking attendance trends.
6. **Notifications** - Alerts for students with low attendance.

## User Classes and Characteristics

1. **Faculty**: Can log student attendance and view attendance history.
2. **Students**: Can track attendance and get alerts for low attendance.

## Operating Environment

* **Frontend**: React.js
* **Backend**: Node.js with Express.js
* **Database**: MongoDB
* **Browser Compatibility**: Chrome, Firefox, Edge

## Design and Implementation Constraints

1. The system must be secure and prevent unauthorized modifications.
2. The database should efficiently store attendance records for quick access.
3. The interface should be mobile-friendly for accessibility.

## 3. System Features

## 3.1 User Authentication

* **Description**: Secure login system for faculty and students using email and password.
* **Preconditions**: The user must be registered in the system.
* **Postconditions**: Successful authentication grants access to the system.

### ****3.2 Faculty Attendance Logging****

* **Description**: Faculty members can mark student attendance.
* **Preconditions**: Faculty must log in and select a class.
* **Postconditions**: Attendance records are saved in the database.

### ****3.3 Faculty Attendance Logging****

* **Description**: Students can view their attendance records.
* **Preconditions**: The student must be logged in.
* **Postconditions**: Attendance data is displayed with filters for date range.

### ****3.4 Attendance Reports****

* **Description**: Students can view their attendance records.
* **Preconditions**: The student must be logged in.
* **Postconditions**: Attendance data is displayed with filters for date range.

### ****3.5 Graphical Attendance Overview****

* **Description**: Attendance trends are shown using charts.
* **Preconditions**: Attendance data must be available.
* **Postconditions**: Graphs display attendance percentages over time.

## ****4. External Interface Requirements****

**4.1 User Interfaces**

* Login Page for faculty and students.
* Attendance Logging Page for faculty.
* Attendance Tracking Dashboard for students.
* Report Generation Page for administrators.

### ****4.2 Software Interfaces****

* Database: MongoDB stores attendance records.
* Backend: Node.js handles authentication and data retrieval.

**4.3 Communications Interfaces**

* **Frontend to Backend**: React.js communicates with Node.js to send and receive data.
* **Backend to Database**: Node.js interacts with MongoDB to store and retrieve attendance records.
* **Data Backup Service**: The system periodically backs up attendance records to prevent data loss.

## ****5. Other Nonfunctional Requirements****

**5.1 Performance Requirements**

* The system should handle at least 500 concurrent users without performance loss.

**5.2 Security Requirements**

* User authentication must be secure.
* Data encryption should be implemented for sensitive information.

**5.3 Usability Requirements**

* The system should be user-friendly and work on all devices.

**5.4 Scalability Requirements**

* The system should support additional students and faculty without slowing down.

# 6. Other Requirements

Error handling mechanisms should ensure smooth operations. The system should support role-based access control for different user types.

.