**Software Requirements Specification (SRS) for School Management System**

**Module Name: Attendance Tracking and Fee Management**

## Table of Contents

**1. Introduction**

1.1 Purpose

1.2 Scope

1.3 Objectives

1.4 Definitions and Abbreviations

**2. Overall Description**

2.1 Product Perspective

2.2 Product Features

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

**3. System Features**

3.1 Attendance Tracking

3.2 Fee Management

3.3 Reporting and Dashboards

3.4 User Authentication

**4. Functional Requirements**

4.1 Attendance Module

4.2 Fee Management Module

4.3 Reporting Module

4.4 User Authentication

**5. Use Case Scenarios**

5.1 Use Case: Attendance Tracking

5.2 Use Case: Fee Payment Processing

5.3 Use Case Diagram for visualization

**6. Non-Functional Requirements**

6.1 Performance Requirements

6.2 Usability Requirements

6.3 Reliability Requirements

6.4 Security Requirements

6.5 Maintainability Requirements

6.6 Portability Requirements

**7. System Design and Architecture**

7.1 System Architecture Diagram

**8. Workflow Diagrams and Models**

8.1 Attendance Workflow Diagram

8.2 Fee Management Workflow Diagram

8.3 ER-Model (Entity Relational Model)

**9. Technical Requirements**

9.1 Software Requirements

9.2 Hardware Requirements

**10. Sample Screens and Reports**

10.1 Attendance Module Interface

10.2 Fee Management Module Interface

10.3 Reporting Module

**11. Legal and Regulatory Compliance**

11.1 Data Privacy

11.2 Financial Compliance

11.3 Accessibility Standards

11.4 Notifications and Consent

**12. Future Enhancements**

12.1 Mobile App Development

12.2 Offline Mode

12.3 Advanced Analytics

12.4 Multi-Language Support

12.5 Multi-School Management

12.6 Integration with Learning Management Systems (LMS)

12.7 Event and Activity Management

**13. Assumptions and Constraints**

13.1 Assumptions

13.2 Constraints

**14. Appendices**

14.1 Glossary of Terms

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the School Management System (SMS). The primary focus is on automating attendance tracking and fee management processes for schools, providing efficiency, accuracy, and transparency.

## 1.2 Scope

The system will facilitate:  
• Real-time attendance tracking with detailed summaries.  
• Fee management, including payment tracking, overdue reminders, and financial reporting.  
• Role-based access control for secure and appropriate access.  
• Analytics dashboards to visualize attendance and fee data.

## 1.3 Objectives

• Automate attendance and fee processes.  
• Enhance transparency in school administration.

## 1.4 Definitions and Abbreviations

• SMS: School Management System.  
• CRUD: Create, Read, Update, and Delete.  
• Admin: User with full system privileges.

**2. Overall Description**

## 2.1 Product Perspective

The SMS is a web-based platform designed for schools to replace manual attendance and fee processes. It will integrate with existing school systems and provide a seamless experience for teachers, administrators, and parents.

## 2.2 Product Features

• Attendance tracking with real-time updates.  
• Automated fee management and reminders.  
• Secure role-based user authentication.  
• Reporting tools with export options (PDF/Excel).

## 2.3 User Characteristics

• Teachers: Moderate tech skills; need a simple interface for attendance marking.  
• Staff: Basic familiarity with data entry for fee management.  
• Administrators: Advanced skills for managing system configurations.

## 2.4 Constraints

• Internet connectivity required for real-time data synchronization.  
• Limited to web browsers during the initial release.

## 2.5 Assumptions and Dependencies

• Schools will provide accurate data for system initialization.  
• Users will undergo brief training for system adoption.

# 3. System Features

**3.1 Attendance Tracking**

* Daily attendance marking for classes.
* Generate class and school-level attendance summaries.
* Notify parents of absenteeism automatically.

**3.2 Fee Management**

* Record payments (full/partial) and outstanding balances.
* Notify parents of overdue payments.
* Generate detailed financial reports.

**3.3 Reporting and Dashboards**

* Export data for specific timeframes.
* Provide charts for attendance trends and fee collections.

**3.4 User Authentication**

* Role-based access for teachers, staff, and administrators.
* Secure login using encryption protocols.

**4. Functional Requirements**

**4.1 Attendance Module**

* Teachers can mark attendance by class and date.
* Administrators can view monthly and yearly attendance reports.

**4.2 Fee Management Module**

* Staff can record fee payments and update balances.
* The system sends automated reminders for unpaid fees.

**4.3 Reporting Module**

* Reports can be filtered by date, class, and student.
* Data export functionality is supported.

**4.4 User Authentication**

* Secure login for all user roles with password encryption.

# 5. Use Case Scenarios

## 5.1 Use Case: Attendance Tracking

Actors: Teacher, Administrator

Preconditions: Teacher is authenticated and has access to the assigned class.

Post conditions: Attendance is recorded, and relevant stakeholders are notified.

**Main Flow:**

1. Teacher logs into the system.
2. Navigates to the "Attendance" module.
3. Selects the class and date.
4. Marks each student's attendance status (Present, Absent, Late, Excused).
5. Submits the attendance record.
6. System saves the data and updates attendance summaries.
7. System sends notifications to parents of absent students.
8. Administrator can view and generate attendance reports.

## 5.2 Use Case: Fee Payment Processing

Actors: Accountant, Parent

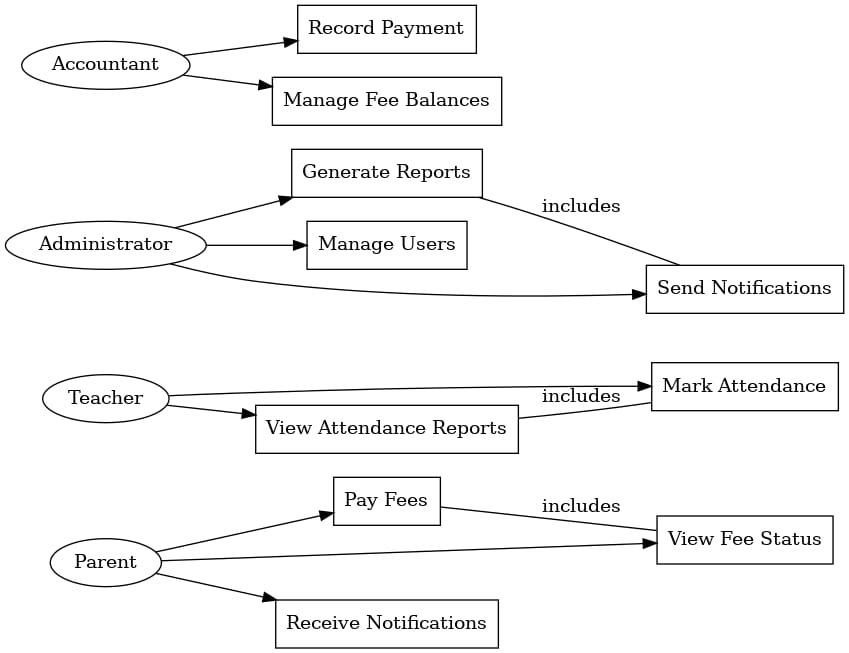
Preconditions: Student's fee structure is defined; Parent has access to the payment portal.

Post conditions: Payment is recorded, and receipt is generated.

**Main Flow:**

1. Parent logs into the system.
2. Navigates to the "Fee Management" module.
3. Views outstanding fees and due dates.
4. Selects a payment method (Credit Card, Bank Transfer, etc.).
5. Enters payment details and confirms the transaction.
6. System processes the payment and updates the student's fee balance.
7. System generates and sends a payment receipt to the parent.
8. Accountant can view and generate financial reports.

## 5.2 Use Case Diagram:



Here's a concise explanation of the **Use Case diagram**:

**Key Actors:**

1. **Parent**: Handles payments and views student information.
2. **Teacher**: Marks attendance and views class reports.
3. **Administrator**: Manages users, generates reports, and configures the system.
4. **Accountant**: Records payments and manages fee balances.

**Key Use Cases:**

1. **Mark Attendance**: Teachers can record daily attendance for their classes.
2. **View Attendance Reports**: Teachers and administrators can access attendance summaries.
3. **Pay Fees**: Parents can pay fees using online methods.
4. **View Fee Status**: Parents can check outstanding balances.
5. **Send Notifications**: The system notifies parents about overdue payments and absenteeism.
6. **Generate Reports**: Administrators can generate attendance and fee collection reports.
7. **Manage Users**: Administrators can create and manage users (e.g., teachers, parents).
8. **Receive Notifications**: Parents receive alerts for payments and attendance.

# 6. Non-Functional Requirements

## 6.1 Performance Requirements

* The system shall support up to 1,000 concurrent users without performance degradation.
* Attendance data entry should be processed within 2 seconds.
* Fee transaction processing should be completed within 5 seconds.
* Reports should be generated within 5 seconds for standard queries.

## 6.2 Usability Requirements

* The user interface shall be intuitive and consistent across all modules.
* The system shall provide tooltips and help documentation for user assistance.
* Users shall be able to perform primary tasks (e.g., marking attendance, processing payments) within three clicks from the dashboard.

## 6.3 Reliability Requirements

* The system shall have an uptime of 99.9% during school hours (8 AM to 4 PM, Monday to Friday).
* Data backups shall be performed daily to prevent data loss.
* The system shall provide mechanisms for error detection and recovery.

## 6.4 Security Requirements

* All user data shall be encrypted in transit and at rest using industry-standard encryption protocols.
* The system shall enforce strong password policies and support multi-factor authentication.
* Access to sensitive data shall be restricted based on user roles and permissions.
* The system shall log all user activities for audit purposes.

## 6.5 Maintainability Requirements

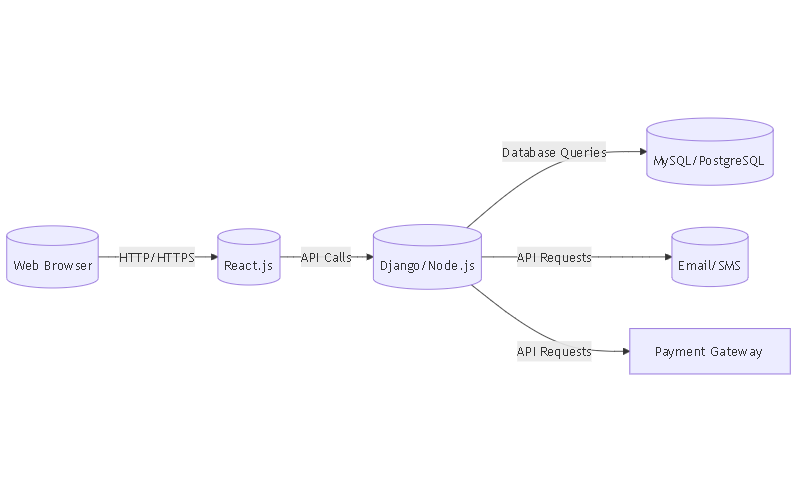
* The system shall be modular to facilitate easy updates and maintenance.
* Code documentation shall be provided to support future development.
* The system shall support configuration changes without requiring downtime.

## 6.6 Portability Requirements

* The system shall be accessible via major web browsers (Chrome, Firefox, Safari, Edge).
* The system shall be responsive and usable on various devices, including desktops, tablets, and smartphones.

# 7. System Design and Architecture

## 7.1 System Architecture Diagram

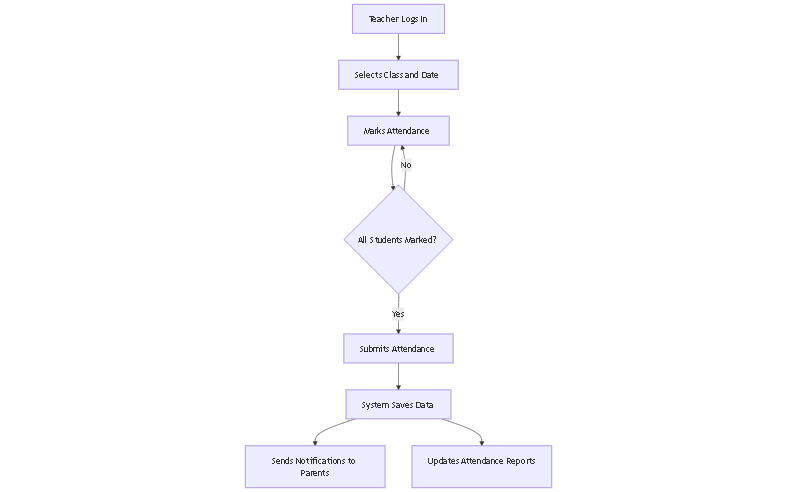


**Explanation:**

* Client Layer: Users access the system through web browsers on various devices.
* Frontend Layer: Developed using React.js, it handles the presentation logic and user interactions.
* Backend Layer: Implemented with Django or Node.js, it manages business logic, processes requests, and interacts with the database.
* Database: MySQL or PostgreSQL stores all persistent data, including user information, attendance records, and financial transactions.
* Notification Service: Handles sending emails and SMS notifications to parents and staff.
* Payment Gateway: Integrates with external payment processors to handle fee transactions securely.

# 8. Workflow Diagrams and Models

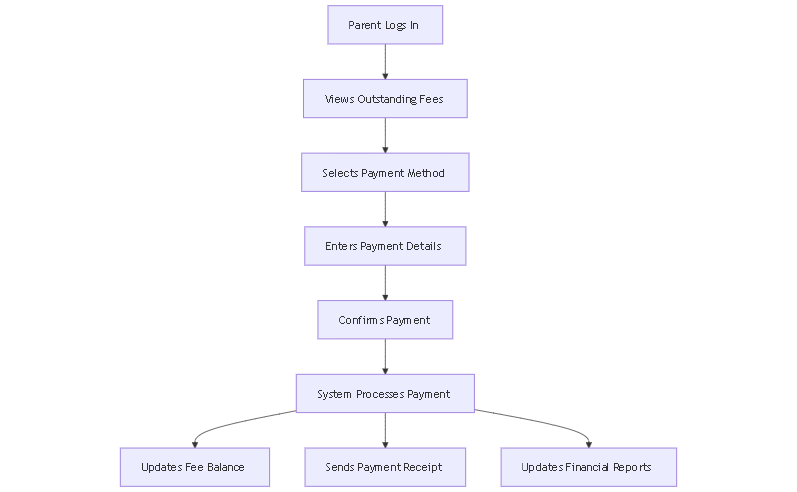
## 8.1 Attendance Workflow Diagram



**Explanation:**

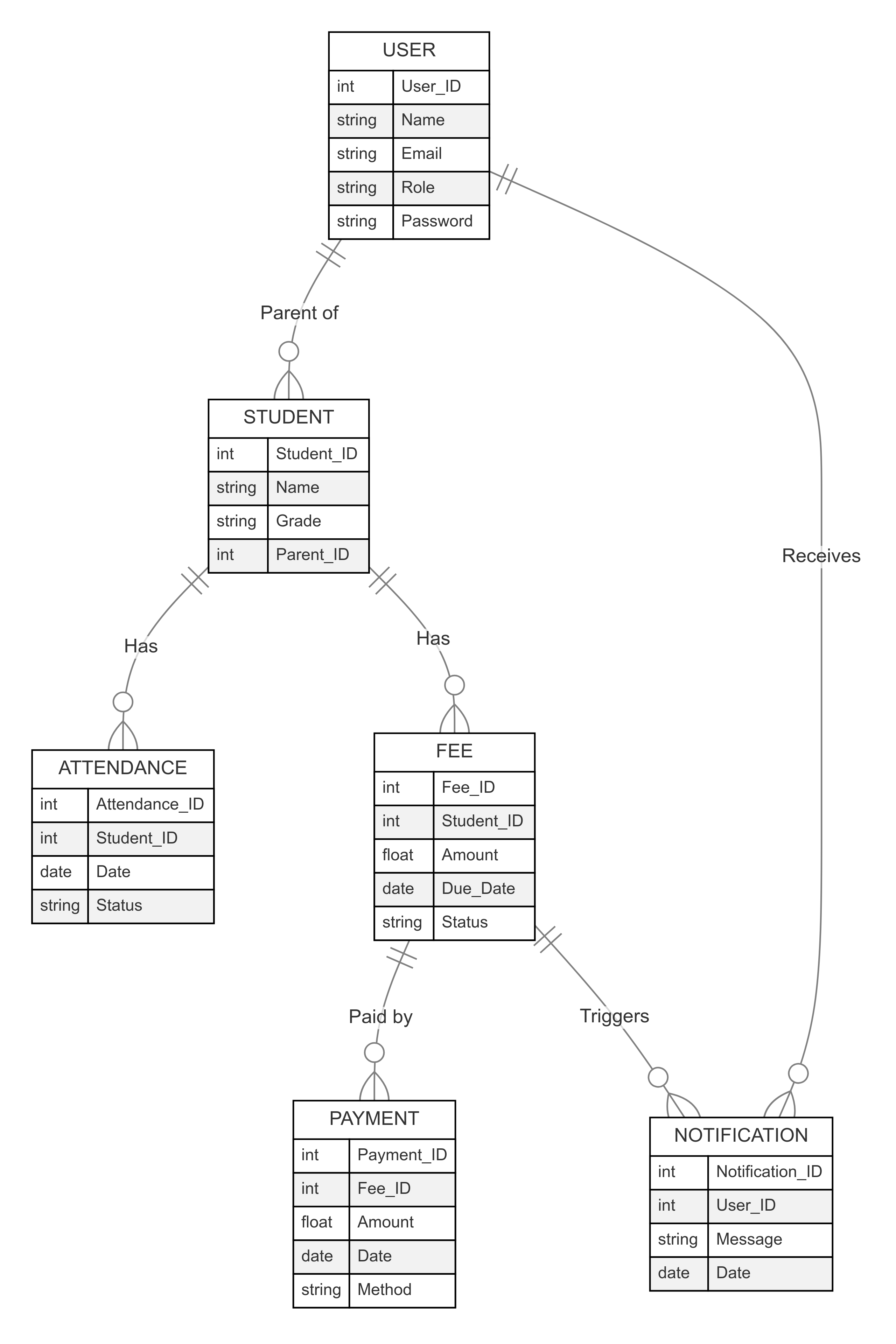
* The teacher logs into the system and selects the relevant class and date.
* Marks each student's attendance status.
* Once all students are marked, the teacher submits the attendance.
* The system saves the data, sends notifications to parents of absent students, and updates attendance reports.

## 8.2 Fee Management Workflow Diagram



**Explanation:**

* The parent logs into the system and views outstanding fees.
* Selects a preferred payment method and enters the necessary details.
* Confirms the payment, after which the system processes the transaction.
* The system updates the student's fee balance, sends a payment receipt to the parent, and updates financial reports.

**8.3 ER-Model (Entity Relational Model)**

**Here's a concise explanation of the ER diagram:**

* **User**: Represents parents, teachers, or administrators. A user can be a parent of one or more students.
* **Student**: Represents students. Each student has multiple attendance records and fee entries.
* **Attendance**: Tracks daily student attendance (e.g., Present, Absent).
* **Fee**: Manages fee details for students, including amount, due date, and status.
* **Payment**: Tracks payments made toward fees (e.g., partial or full payments).
* **Notification**: Handles alerts (e.g., fee reminders, attendance updates) sent to users.

**Key Relationships:**

* A **User** is a "Parent of" one or more **Students**.
* A **Student** "Has" multiple **Attendance** and **Fee** records.
* A **Fee** can be "Paid by" multiple **Payments**.
* A **User** "Receives" notifications, while a **Fee** "Triggers" notifications.

# 9. Technical Requirements

## 9.1 Software Requirements

### Server-Side:

* Operating System: Ubuntu 20.04 LTS or later
* Web Server: Nginx or Apache
* Programming Language: Python 3.8+ (for Django) or Node.js 14+ (for Node.js)
* Database: MySQL 8.0+ or PostgreSQL 12+
* Notification Service: Integration with SMTP server for emails and SMS gateway for text messages
* Payment Gateway: Integration with Stripe or PayPal APIs

### Client-Side:

* Web Browser: Latest versions of Chrome, Firefox, Safari, or Edge.
* JavaScript Enabled for responsive user interfaces.

**9.2 Hardware Requirements**

* Client Systems: Minimum of 4GB RAM and a 2GHz processor for smooth operation.
* Hosting: Cloud-based hosting (AWS, Azure, or Google Cloud) to ensure 99.9% uptime.
* Why this is important: These requirements provide a scalable and secure platform that is compatible with modern devices and services.

**10. Sample Screens and Reports**

**10.1 Attendance Module Interface**

* Dropdown menus for selecting class and date.
* Checkbox-based options for marking each student as Present, Absent, Late, or Excused.

**10.2 Fee Management Module Interface**

* Fields for entering payment details (amount, method).
* Real-time display of balances and overdue fee reminders.

**10.3 Reporting Module**

* Charts and graphs for attendance trends and fee collection statistics.
* Export options for reports in PDF or Excel formats.

**Why this is important:** These samples make the system easy to use, ensuring stakeholders can visualize and interact with key features efficiently.

**11. Legal and Regulatory Compliance**

The system shall comply with all relevant legal and regulatory requirements to ensure secure, ethical, and lawful operation. The following standards are considered critical:

**11.1 Data Privacy**

* General Data Protection Regulation (GDPR): If operating in the EU, the system shall ensure that personal data is collected, stored, and processed securely with user consent.
* Family Educational Rights and Privacy Act (FERPA): For U.S. schools, the system shall protect students’ educational records and restrict unauthorized access.
* Data Encryption: All sensitive data (e.g., student attendance and parent payment information) shall be encrypted at rest and in transit.

**11.2 Financial Compliance**

* The system shall integrate with PCI-DSS compliant payment gateways (e.g., Stripe, PayPal) to ensure secure financial transactions.
* Adhere to local financial reporting laws for managing school fee records, ensuring audit readiness.

**11.3 Accessibility Standards**

The system shall comply with WCAG 2.1 (Web Content Accessibility Guidelines) to ensure accessibility for users with disabilities.

**11.4 Notifications and Consent**

For regions requiring parental consent, the system shall notify and obtain authorization from parents before processing sensitive student information.

**12. Future Enhancements**

The following features are planned for future versions of the School Management System to ensure scalability, usability, and adaptability:

**12.1 Mobile App Development**

Create a dedicated mobile application for teachers, parents, and administrators to access attendance and fee management features.

**12.2 Offline Mode**

Develop functionality for teachers to record attendance offline and sync data with the system when an internet connection is restored.

**12.3 Advanced Analytics**

Implement predictive analytics for analyzing attendance trends, identifying at-risk students, and generating insights on fee collection patterns.

**12.4 Multi-Language Support**

Add support for multiple languages to accommodate schools with diverse linguistic requirements.

**12.5 Multi-School Management**

Extend the system to allow centralized management of multiple schools, providing administrators with access to consolidated data across institutions.

**12.6 Integration with Learning Management Systems (LMS)**

Enable seamless integration with platforms such as Moodle, Google Classroom, and others to provide a unified experience for students, teachers, and administrators.

**12.7 Event and Activity Management**

Add a feature for scheduling and managing school events, extracurricular activities, and parent-teacher meetings.

**13. Assumptions and Constraints**

This section outlines the assumptions considered during the development of the School Management System (SMS) and the constraints that may impact its implementation and operation.

**13.1 Assumptions**

1. **Accurate Data Input**

Schools will provide complete and accurate initial data for setting up attendance and fee records.

1. **User Training**

Teachers, staff, and administrators will undergo basic training to ensure smooth system adoption.

1. **Stable Internet Connection**

Users will have access to a reliable internet connection to utilize real-time data synchronization.

1. **Compliance with Regulations**

Schools will adhere to privacy and financial regulations while using the system.

1. **Device Accessibility**

Users will access the system via modern web browsers on desktops, tablets, or smartphones.

**13.2 Constraints**

1. **Internet Dependency**

The system requires an active internet connection for real-time updates and notifications.

1. **Browser Compatibility**

The initial release is optimized for web browsers (e.g., Chrome, Firefox, Edge) and may not support outdated versions or less common browsers.

1. **Limited Offline Functionality**

Attendance tracking and fee management features are unavailable without internet connectivity.

1. **Hardware Requirements**

Client devices must meet minimum specifications (e.g., 4GB RAM, modern processors) for optimal performance.

1. **Data Entry Responsibility**

Schools are responsible for entering accurate attendance and fee-related data into the system.

1. **Payment Gateway Integration**

The system's fee management relies on third-party payment gateways that may have regional limitations or fees.

**14. Appendices**

**Glossary of Terms (Enhanced Explanation)**

This section explains technical terms and abbreviations to improve clarity for non-technical users:

* CRUD (Create, Read, Update, Delete): Fundamental operations performed on data in a database or software system. For example, teachers can "create" an attendance record, "read" student data, "update" attendance details, or "delete" incorrect records.
* Admin: A user with the highest level of access and privileges in the system. The admin can manage users, configure settings, and perform advanced actions like data exports.
* SMS (School Management System): The core software system being developed to automate school processes like attendance tracking and fee management.
* Role-Based Access Control: A security feature that restricts access based on user roles (e.g., Teacher, Parent, Admin), ensuring appropriate access levels.
* PCI-DSS Compliance: Standards for secure handling of payment data to protect financial transactions in the Fee Management module.
* WCAG (Web Content Accessibility Guidelines): Guidelines to ensure the system is accessible to users with disabilities, enhancing usability and inclusivity.