Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

FACULTY KIT

Objective

The faculty kit is designed to outline the roles, responsibilities, and tools required for the successful design, development, and implementation of a School Admin System. The system involves interaction among students, teachers, administrators, and parents through various modules. These modules include attendance tracking, grade management, timetable scheduling, communication, and fee payment management. This kit ensures that faculty members can efficiently guide and monitor students in developing the system, focusing on key deliverables, timelines, and testing protocols.

Requirements Specification

The School Admin System will include:

1. User Authentication

- Secure login with role-based access for Admins, Teachers, and Students.
- Password recovery and two-factor authentication for enhanced security.

2. Attendance Management

- **Teachers**: Mark and update daily attendance for students.
- **Students**: View attendance records.
- Automated notifications for low attendance thresholds to students.

3. Grade Management

- **Teachers**: Input and manage grades for assessments and examinations.
- **Students**: View grades for progress tracking.
- Statistical analysis and graphical representation of performance trends.

4. Timetable Scheduler

- Automated timetable creation with conflict resolution for classes and resources.
- Customization options for special schedules (exams, holidays, or events).

5. Fee Payment and Management

- **Admins**: Manage fee structures, payment schedules, and generate invoices.
- **Students**: Pay fees through an integrated payment gateway.
- Automated reminders for due or overdue payments.

6. Communication Module

- **Teachers to Students**: Announcements, messages, and updates.
- **Admins to All Users**: Broadcast important notices.
- Messaging system for quick communication within the platform.

7. Reports and Analytics

- Generate reports on:
    - Attendance.
    - Academic performance.
    - Fee payment status.
- **Admins**: Access system-wide analytics for decision-making.

8. Library Management

- Access to eBooks, articles, and other educational materials.
- Advanced search functionality for students and teachers.
- Admin control for adding or managing resources.

9. Notifications and Alerts

- Automated alerts for events like fee due dates, exam schedules, or attendance issues.
- Real-time push notifications for updates via email or SMS.

10. Admin Control Panel

- Manage users and assign roles (Admin, Teacher, Student).
- Approve or reject timetable changes, fee adjustments, or grade edits.
- System monitoring and user activity tracking.

11. Scalability and Security

- Ensure the system is scalable to accommodate growing numbers of users and data.
- Implement industry-standard encryption for data protection.
- Regular system backups and disaster recovery mechanisms.

Technology Familiarization –

The School Admin System will be developed using Node.js for the backend and Express.js for handling server-side logic and API development, ensuring a scalable and high-performance environment. React.js will serve as the frontend framework, enabling the creation of a responsive, dynamic, and user-friendly interface. MongoDB, a NoSQL database, will be used to manage structured and unstructured data such as student records, attendance, grades, and fees with high flexibility and scalability. This MERN (MongoDB, Express, React, Node) stack offers a modern, full-stack JavaScript solution that supports real-time updates, robust performance, and ease of future integration with mobile applications or third-party systems. The architecture also emphasizes security, maintainability, and smooth user experience, making it ideal for comprehensive school administration.

Database Creation –

The School Admin System will utilize a structured relational database to manage essential data:

- **MongoDB Database** for structured information, including student records, teacher details, attendance, and fee transactions.
- **User**: Stores user details, roles (Admin, Teacher, Student), and authentication credentials.
- **Class Management**: Data on classes, subjects, schedules, and teacher assignments.
- **Fees and Payments**: Information regarding fee structures, payments, and due dates.
- **Attendance**: Tracks daily attendance for students and teachers.

High-Level and Detailed Design –

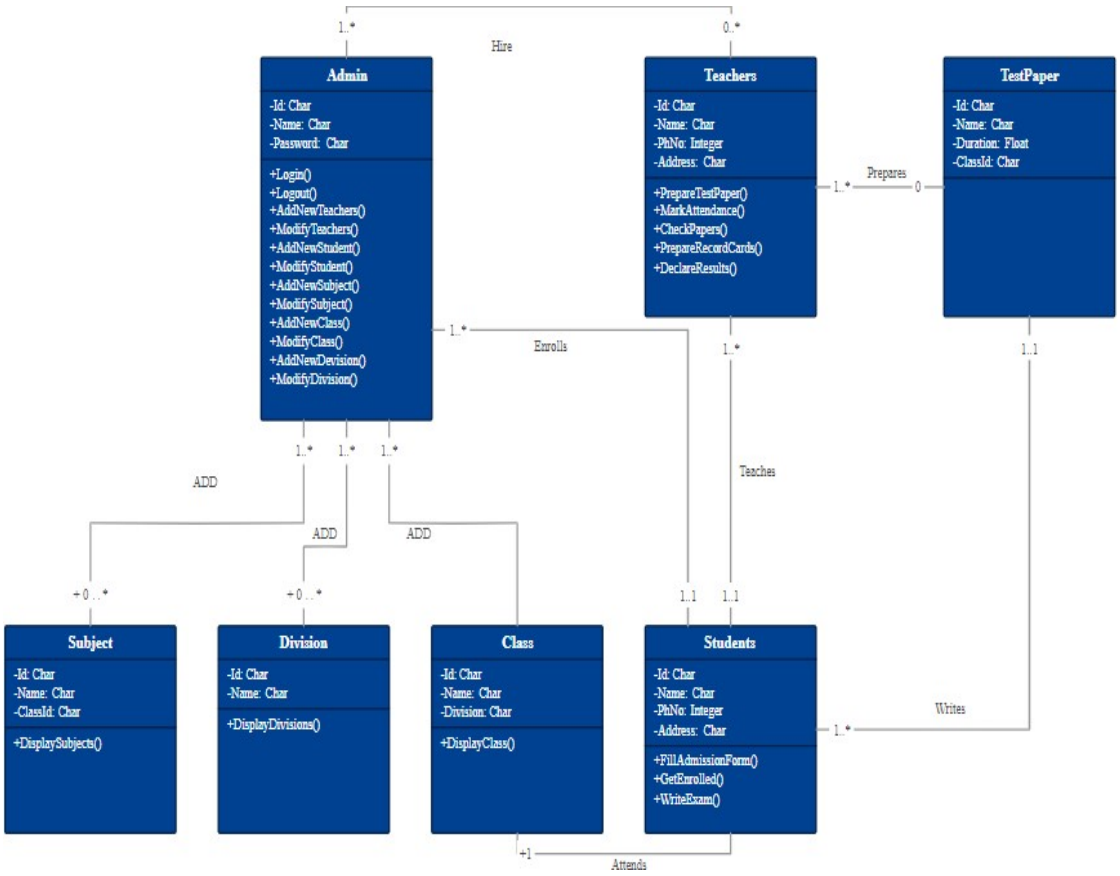
System Overview:

The system consists of three layers:

- **Frontend**: A React-based user interface allowing seamless interaction for material management, subscriptions, and payments.
- **Backend**: Node.js handles the server-side logic, database interactions, and communication with third-party services like payment gateways.
- **Database**: MongoDB (for user, materials, and transactions) ensure scalable, fast, and reliable data storage.

Detailed Design:

The backend of the School Admin System will expose RESTful APIs for user authentication, class management, fee processing, attendance tracking, and report generation. The frontend will interact with these APIs to display real-time data, providing a seamless and responsive user experience.



Frontend Implementation –

React.js will be used for developing the user interface of the School Admin System. The UI will include sections for login, registration, dashboard access, attendance, fee processing, and report management. Each user type (Admin, Teacher, Student) will have distinct permissions and views:

- **Students**: View schedules, track attendance, and access reports.
- **Teachers**: Manage attendance, assign grades, and upload class materials.
- **Admins**: Oversee user roles, manage school operations, and generate reports.

Integrating the Frontend with the Database –

To integrate the frontend with the database, the backend will expose REST APIs using NodeJS and MongoDB, which will handle HTTP requests. These requests will perform CRUD operations on the database, such as managing attendance, updating fee status, and fetching student or teacher data. React.js will consume these APIs via Axios, ensuring the user interface dynamically reflects the data stored in the MongoDB database.

Test Plan Review -

Testing will be carried out at multiple stages to verify the functionality and reliability of the system:

- **Unit Testing:** Validating the accuracy of backend logic and API functionality.
- **Integration Testing:** Testing the seamless interaction between the React.js frontend and the Node.js for backend.
- **UI/UX Testing:** Ensuring the user interface is intuitive, accessible, and fully functional.
- **Performance Testing:** Assessing the system's capability to manage high user loads and maintain performance.

Final Review -

At the end of the project, a comprehensive final review will be held to ensure that the system meets all outlined requirements. This review will focus on verifying the functionality of user authentication, material access, subscription management, and the payment module. Furthermore, feedback from users (Admin, Teacher, Parents, and Student) will be collected to identify areas for improvement and fine-tuning before the system goes live.

Documents/References that May Aid the Process of Evaluation-

- **NodeJS Documentation**: For backend implementation and API development.
- **React.js Documentation**: For frontend development and UI handling.
- **MongoDB Database Tutorials**: For database design and integration.
- **Agile Methodology Resources**: For project management and sprint planning.

Conclusion –

The Faculty Kit serves as a comprehensive guide for faculty members involved in the development and deployment of the School Admin System. By covering all stages of the project, from initial planning to final review, the kit ensures that faculty are well-equipped to manage and oversee the entire process. It provides insights into the technologies used, including Node, React, and MongoDB databases, as well as detailed implementation steps for both the front-end and back-end systems. With a focus on seamless integration and user-friendly design, the kit ensures that the system will meet educational objectives and provide a valuable learning experience for students. Additionally, the test and final review phases enable faculty to ensure the system's functionality, scalability, and security, ensuring a high-quality, user-centered School Admin System.