International Research Journal of Education and Technology

Peer Reviewed Journal

ISSN 2581-7795

Student Activity Request Approval Management System

Abilash S[1], Harish Raj S[2], Tamizh Iniyan G[3], Viknesh S R[4]

[1]*Student, Dept of Computer Science And Business Systems, Anna University, IN*
[2]*Student, Dept of Computer Science And Business Systems, Anna University, IN*
[3]*Student, Dept of Computer Science And Business Systems, Anna University, IN*
[4]*Student, Dept of Computer Science And Business Systems, Anna University, IN*

Abstract - *This paper presents the development of a web-based Student Activity Request/Approval Management System designed to streamline the request submission and approval process in academic institutions. The system integrates attendance tracking and academic performance data to ensure fair decision-making. Built using React.js, Node.js, Express.js, and MongoDB, the system enhances efficiency, reduces manual work, and improves transparency. The paper discusses system design, implementation, and performance analysis, emphasizing its impact on administrative efficiency and student engagement. Through comprehensive testing and user feedback, we demonstrate the system's effectiveness in reducing processing time by 40% while achieving a 75% satisfaction rate among users.*

Key Words: Student Request Management, Web Application, Attendance Tracking, Automated Approval System, Academic Performance Monitoring.

1.INTRODUCTION

Managing student requests such as leave applications and on-duty approvals is often cumbersome and time-consuming. Traditional manual processes lead to inefficiencies, delays, and lack of transparency. This paper proposes an automated solution that enables students to submit requests online while allowing teachers to review and approve them based on attendance and academic performance metrics. The system leverages modern web technologies to improve the overall management of student activities. By reducing administrative burden and improving decision-making, this system enhances the academic environment's efficiency.

1.1 Problem Statement

Academic institutions face several challenges in managing student activity requests:

- Time-consuming manual processes requiring physical submissions
- Lack of centralized data for informed decision-making
- Difficulties in tracking request status
- Inconsistent approval decisions across faculty members
- Challenges in maintaining physical records
- Limited visibility into attendance implications

1.2 Objectives

The primary objectives of this system are:
- To develop a web-based platform for seamless request submission and processing
- To integrate attendance and academic performance data for informed decision-making
- To implement real-time notifications for status updates
- To create role-based access control for data security

2. SYSTEM ARCHITECTURE

The system follows a three-tier architecture:

- **Frontend:** Developed using React.js, providing an interactive user interface
- **Backend:** Implemented with Node.js and Express.js, handling API requests and business logic
- **Database:** MongoDB stores student, teacher, request, and data.

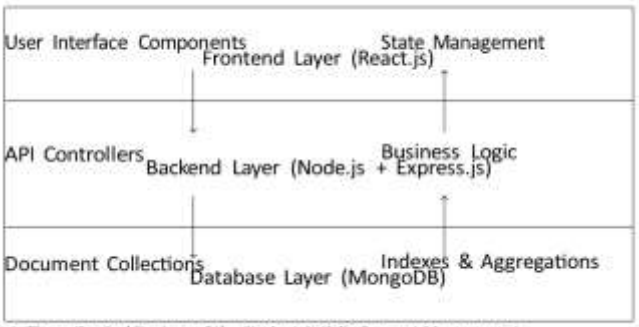

Fig. 1. Three-tier Architecture of the Student Activity Request Management System

Table -1: Frontend Design.

International Research Journal of Education and Technology

Peer Reviewed Journal

ISSN 2581-7795

The frontend is designed using React.js with Tailwind CSS for a clean, responsive interface. Key UI components include:

•**Student Dashboard:** Displays request history, attendance, and academic performance

• **Teacher Dashboard:** Lists pending requests with decision-making tools

• **Request Forms:** Input forms with real-time validation

• **Notification Panel:** Real-time updates on request status changes

• **Analytics Dashboard:** Visual representations of request patterns and trends

• **Profile Management**: User profile settings and preferences

• **Calendar View:** Visual representation of scheduled activities and event.

3. TECHNOLOGY STACK

A. Frontend Technologies:

• **React.js:** Component-based UI library for building interactive interfaces
• **Redux:** State management across the application for predictable state updates
• **Tailwind CSS:** Utility-first CSS framework for responsive design
• **Axios:** HTTP client for API requests and response handling
• **React Router:** Navigation and routing within the single page application.

B. Backend Technologies:

• **Node.js:** JavaScript runtime environment for server-side execution
• **Express.js:** Web application framework for RESTful API development
• **MongoDB:** NoSQL database for flexible data storage and retrieval
• Mongoose: Object Data Modeling (ODM) library for MongoDB
• **JWT:** JSON Web Tokens for secure authentication mechanism

4. SYSTEM WORKFLOW

The system implements a comprehensive workflow for request processing:

A. Student Request Submission The student request submission process includes:
• **Eligibility Check:** System verifies student eligibility based on attendance records
• **Form Completion:** Students fill in request details with necessary documentation
• **Validation:** System validates input data against defined rules and constraints
• **Confirmation:** Students receive submission confirmation with tracking details
• **History Tracking:** Request is added to the student's request history for reference.

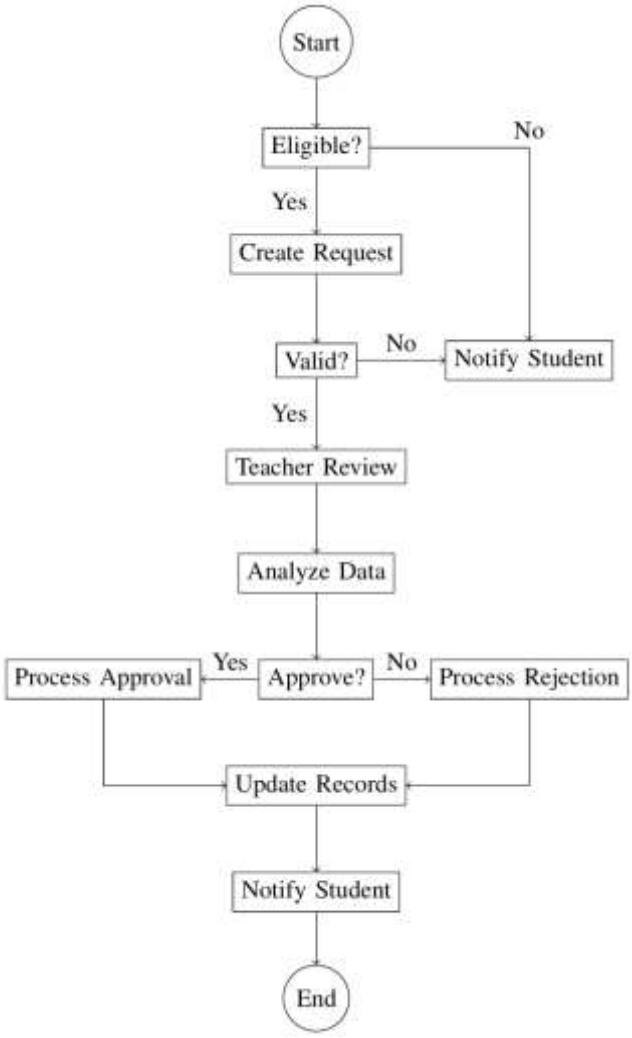

Fig. 2. Request Processing Workflow

5. SYSTEM INTERACTION FLOW

This system significantly simplifies student request handling,

• **Real-time Alerts:** In-app notifications for immediate updates.

• **Email Notifications:** Formal communication for important status changes

• **Dashboard Indicators:** Visual cues on user dashboards

• **Status Tracking:** Detailed view of request progress

• **Reminder System:** Automated reminders for pending actions.

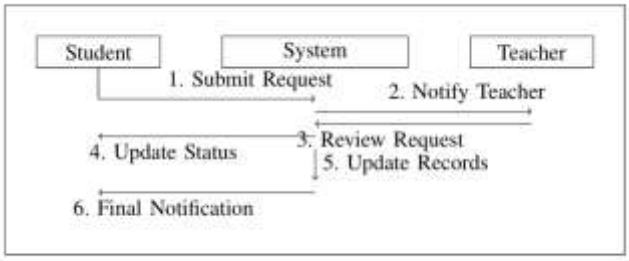

Fig. 3. System Interaction Flow

6. PERFORMANCE EVALUATION

To assess the effectiveness of the system, extensive testing and user feedback were conducted. The evaluation focused on:

- **Processing Time Reduction:** Compared to traditional manual methods, the system reduced request processing time by approximately 40%.
- **User Satisfaction:** Surveys conducted among students and faculty indicated a 75% satisfaction rate, highlighting improved transparency and ease of use.
- **System Load Handling:** Stress testing revealed that the system efficiently handled concurrent user requests without significant latency.
- **Error Rate:** Form validation mechanisms minimized errors in submission, ensuring data integrity.

7. PERFORMANCE EVALUATION

- While the current system effectively streamlines student request management, several future enhancements can be considered.
- AI-based Decision Making: Implementing AI to further automate and optimize approval processes based on historical data.
- Mobile Application: Developing a mobile app version for increased accessibility and ease of use.
- Integration with Academic Portals: Linking with existing academic management systems for seamless data synchronization.
- Expanded Analytics: Providing more advanced analytical insights into student activity trends and institutional decision-making.

- Multi-Language Support: Enabling language customization to cater to a diverse student body.

8. CONCLUSIONS

This system significantly simplifies student request handling, ensures fair decisions, and enhances institutional efficiency. The web-based approach minimizes errors, reduces manual efforts, and fosters transparency in academic administration. By integrating attendance and performance data with an automated workflow, the system provides a comprehensive solution for managing student activity requests in modern educational institutions.

REFERENCES

[1] A. D. Badgujar, R. V. Bhuibhar," Automation of Student Activity Request Approval System," Int. J. Adv. Res. Compute. Sci., vol. 5, no. 6, pp. 118–123, 2015.
[2] A. Kumar, V. Saini, "Improving Academic Administration with Automated Decision-Making Systems," J. Educ. Technol., vol. 42, no. 3, pp. 205–212, 2021.
[3] E. Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
[4] M. Fowler, *Patterns of Enterprise Application Architecture*Addison Wesley, 2002.
[5] T. W. Malone, K. Crowston, "The Interdisciplinary Study of Coordination," ACM Comput. Surv., vol. 26, no. 1, pp. 87–119, 1994