
Software Requirements Specification

for

MU INTRANET PORTAL

Version 1.0

Prepared by

Group Name: Ideal Bits

**Sai Sarvani.P
Vivek Reddy.P
Varshika.P
Vignesh Anoop Naidu.N
Shivani.N**

**SE22UARI127
SE22UARI122
SE22UARI124
SE22UARI108
SE22UARI109**

Instructor: Prof. Avinash Arun Chauhan
Course: Software Engineering
Lab Section: A/2
Teaching Assistant: Sri Venkata Surya Phani Teja
Date: 04-03-2025

Contents

CONTENTS	II
REVISIONS.....	II
1 INTRODUCTION.....	1 - 3
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	2
1.5 DOCUMENT CONVENTIONS	2
1.6 REFERENCES AND ACKNOWLEDGMENTS	2
2 OVERALL DESCRIPTION	4 – 5
2.1 PRODUCT OVERVIEW	4
2.2 PRODUCT FUNCTIONALITY	4
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	4
2.4 ASSUMPTIONS AND DEPENDENCIES	5
3 SPECIFIC REQUIREMENTS	6 - 15
3.1 EXTERNAL INTERFACE REQUIREMENTS	6
3.2 FUNCTIONAL REQUIREMENTS	7
3.3 USE CASE MODEL	9
4 OTHER NON-FUNCTIONAL REQUIREMENTS	16
4.1 PERFORMANCE REQUIREMENTS	16
4.2 SAFETY AND SECURITY REQUIREMENTS	16
4.3 SOFTWARE QUALITY ATTRIBUTES	16
APPENDIX A – DATA DICTIONARY	17
APPENDIX B – GROUP LOG	18

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Sarvani, Vivek, Vignesh, Varshika, Shivani	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	04/03/2025

1. Introduction

Developing a user-friendly intranet portal to enhance student and faculty interactions with improved navigation and interactive features.

1.1. Document Purpose

This document outlines the functional and non-functional requirements for the MU Intranet Portal, a centralized web-based platform designed to enhance university operations. The system aims to provide secure access for students and faculty to manage academic and administrative tasks, including attendance tracking, course management, and event notifications. By integrating authentication protocols, real-time notifications, and role-based access, the portal ensures seamless communication and efficient task management.

The MU Intranet Portal will act as a unified hub, digitizing key university functions and reducing manual workload. It will offer a responsive interface accessible across devices, supporting features such as chat systems, notice boards, and resource sharing. This document serves as a technical blueprint, detailing system capabilities, constraints, and integration with existing university infrastructure.

1.2. Product Scope

The MU Intranet Portal is a web-based platform designed to streamline academic and administrative processes for students and faculty. It provides role-based access, ensuring that users can securely log in and access features tailored to their needs. The portal integrates a comprehensive attendance management system, allowing faculty to mark attendance and generate reports while enabling students to track their attendance records. Additionally, the course and resource management feature ensures that students can access lecture notes, assignments, and study materials, while faculty can efficiently upload and organize content.

To enhance communication, the real-time chat and notification system facilitates seamless interaction between students and faculty. The platform also includes an event and notice board system, where users can stay updated on important announcements and upcoming academic events. Administrative users benefit from advanced controls and reporting features, providing insights into student performance, course engagement, and overall portal usage. With its intuitive interface and responsive design, the MU Intranet Portal aims to improve efficiency and accessibility across the university ecosystem.

1.3. Intended Audience and Document Overview

This document is intended for developers, project managers, faculty, and students. It provides a structured overview of the system's architecture, functionality, constraints, and key assumptions, ensuring a shared understanding of the MU Intranet Portal's design and implementation.

1.4. Definitions, Acronyms and Abbreviations

- **MU** – Mahindra University
- **SRS** – Software Requirements Specification
- **API** – Application Programming Interface
- **JWT** – JSON Web Token
- **UI/UX** – User Interface / User Experience

1.5. Document Conventions

This document follows the **IEEE SRS standard**. The following conventions are used:

- **Font:** Times, size 11 or 12 for body text.
- **Headings:** Bold and numbered.
- **Tables:** Used for structured data representation.
- **Terminology:** Standard software engineering terms and definitions are followed.
References and Acknowledgments
- University Database for Student & Faculty Records
- REST API & GraphQL Standards

1.6. References and Acknowledgments

- [1] Node.js, Node.js Documentation, Available: <https://nodejs.org/>.
- [2] Express.js, Express.js Web Framework for Node.js, Available: <https://expressjs.com/>.
- [3] MySQL, MySQL Reference Manual, Available: <https://dev.mysql.com/doc/>.
- [4] JWT, JSON Web Tokens Introduction, Available: <https://jwt.io/introduction/>.
- [5] Tailwind CSS, Tailwind CSS Documentation, Available: <https://tailwindcss.com/>.
- [6] IEEE, IEEE Recommended Practice for Software Requirements Specifications (IEEE 830-1998), Available: <https://standards.ieee.org/>.

- [7] University Database, Student & Faculty Records, [Confidential/Internal Use].
- [8] H. Gomaa, Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Cambridge University Press, 2016.
-
- [9] G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 2005.

We express our gratitude to Avinash Sir for his invaluable guidance and insights for this project. Special thanks to faculty mentors for their support in defining the project structure and requirements.

We acknowledge Mahindra University for providing resources and datasets essential for implementation.

Thanks to open-source communities for their documentation, APIs, and technical support.

2. Overall Description

2.1. Product Overview

The MU Intranet Portal is designed to enhance university operations by digitizing key academic and administrative processes, including attendance tracking, course management, and faculty-student interactions. By integrating with the university's centralised database, the system ensures seamless authentication, secure role-based access, and efficient management of academic resources. Faculty can upload course materials, manage attendance, and post important announcements, while students gain real-time access to coursework, notifications, and university events.

In addition to its academic functionalities, the portal streamlines event coordination and administrative reporting. The event and notice board system keeps students and faculty updated with university events, deadlines, and policy changes. Built-in real-time chat and notification features facilitate instant communication, reducing delays in information exchange. By consolidating these essential functions into a single platform, the MU Intranet Portal fosters a more connected and efficient university ecosystem.

2.2. Product Functionality

- **User Authentication & Authorisation** – Secure login with JWT authentication.
- **Role-Based Access Control** – Custom views for students and faculty.
- **Attendance Management** – Faculty can manage attendance and generate reports.
- **Course Management** – Faculty uploads course materials, students access them.
- **Event & Notice Board** – Faculty posts announcements and upcoming events.
- **Chat & Notifications** – Real-time messaging for students and faculty.

2.3. Design and Implementation Constraints

Several constraints must be considered while developing the MU Intranet Portal to ensure compatibility, scalability, and security:

- **Software Design Methodology:** The system must adhere to the COMET (Collaborative Object Modeling and Enterprise Transformation) method, which emphasizes object-oriented software design, real-time system modeling, and enterprise-level transformations. This ensures that the portal is modular, scalable, and easily maintainable.
 - Reference: Gomaa, H. (2016). "Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures." Cambridge University Press.

- **Modeling Language:** All system architecture and process flows must be documented using Unified Modeling Language (UML). This includes use case diagrams, sequence diagrams, class diagrams, and state diagrams to provide a standardized representation of system behavior.
 - Reference: Booch, G., Rumbaugh, J., & Jacobson, I. (2005). "The Unified Modeling Language User Guide." Addison-Wesley.
- **Hardware Constraints:** The system must be optimized, ensuring it can handle 4,000+ concurrent users without performance degradation. The database must support high-speed query execution to prevent lag in student-faculty interactions.
- **Security Considerations:** The portal must implement OAuth 2.0 and JWT-based authentication to prevent unauthorized access and ensure secure communication between users and the server. Encryption must be applied to sensitive data, including student records, attendance logs, and course materials.
- **Communication Protocols:** All real-time chat and notification services must use WebSocket-based communication for efficient and low-latency message exchange. Email alerts should be sent via SMTP with TLS encryption.
- **Development Standards:** The system must comply with IEEE 830-1998 standards for Software Requirements Specifications (SRS). Code should follow the Google JavaScript Style Guide for readability and consistency. Backend development will adhere to RESTful API principles, ensuring modularity and scalability.

These constraints ensure that the MU Intranet Portal is scalable, secure, and maintainable while meeting the university's operational needs.

2.4. Assumptions and Dependencies

The design and functionality of the **MU Intranet Portal** are based on the following key assumptions:

- **University Database Access:** The university will provide secure and structured access to student and faculty records, ensuring seamless authentication and data retrieval.
- **User Device Compatibility:** Students and faculty will access the portal through modern web browsers and mobile devices, ensuring compatibility with the system's interface.
- **Authentication System Integration:** The university will support OAuth 2.0 or LDAP-based authentication for secure login access.
- **Regular Data Backups:** The system will rely on periodic database backups to prevent data loss and ensure system resilience in case of failures.
- **Faculty and Student Compliance:** Users are expected to regularly update their profiles and interact with the system for attendance, coursework, and event management.

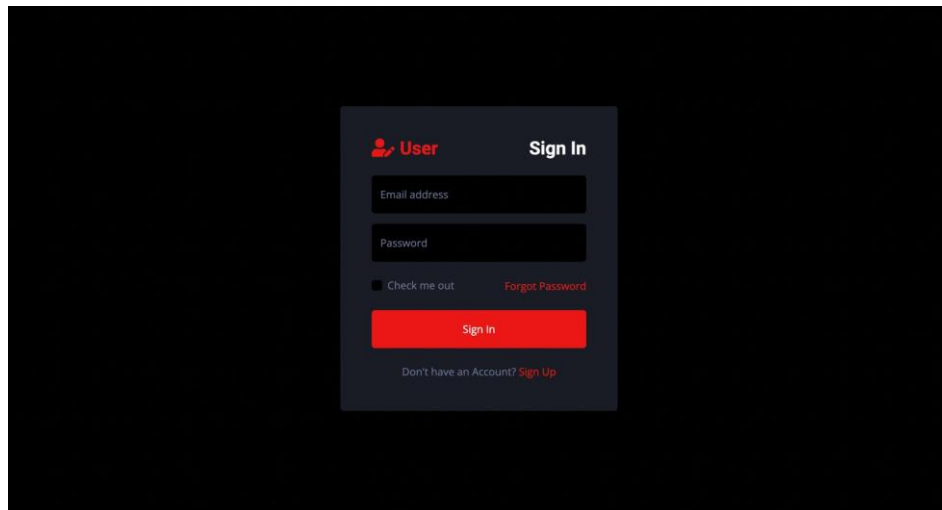
These assumptions, if altered, could impact system performance, security, and overall user experience.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

- The web-based interface will have a responsive design for desktops, tablets, and mobile devices.



- **Login Page:** Allows users to enter credentials and authenticate securely.
- **Dashboard:** Provides an overview of attendance, courses, and notifications.
- **Course Page:** Displays available resources, assignments, and lecture materials.
- **Chat & Notifications:** Interface for real-time messaging.

3.1.2. Hardware Interfaces

- **Server Hardware:** The portal will be hosted on on-premises servers located in the college's server room, ensuring secure data storage and efficient handling of database queries.
- **Mobile Devices:** Smartphones and tablets for accessing the portal.

- **Admin Workstations:** Computers used by faculty for attendance and course management.

3.1.3. Software Interfaces

- **Database Interface:** MySQL stores attendance, users, and course data.
- **Authentication API:** OAuth and JWT for login security.
- **Notification System:** Push notifications and email alerts for event updates.

3.2. Functional Requirements

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform.

3.2.1 F1: User Authentication

The system shall allow users to log in using their university credentials and enforce secure authentication mechanisms such as JWT tokens.

3.2.2 F2: Role-Based Access

The system shall provide different access levels for students and faculty, ensuring restricted functionalities based on user roles.

3.2.3 F3: Attendance Management

The system shall allow faculty to view, edit, and generate attendance reports. Students can check their attendance records in real time.

3.2.4 F4: Course Registration & Elective Selection

The system shall allow students to register for courses and choose electives based on their academic program and seat availability.

3.2.5 F5: Timetable Management

The system shall display a structured weekly timetable for students and faculty, including class locations and times.

3.2.6 F6: Class Rating & Feedback

The system shall prompt students to rate and provide feedback on lectures immediately after the class ends.

3.2.7 F7: Fee Portal

The system shall allow students to view their fee details, make online payments, and download payment receipts.

3.2.8 F8: Course Content & Notes

The system shall allow faculty to upload and manage lecture notes, assignments, and other resources, which students can access and download.

3.2.9 F9: Events & Notice Board

The system shall allow faculty and admins to post announcements, and students shall be able to register for university events.

3.2.10 F10: Results & Grade Tracking

The system shall display students' grades, CGPA, and performance reports for each semester.

3.2.11 F11: Library Management

The system shall allow students to check book availability, view due dates, and request book borrowing or returning.

3.2.12 F12: Book a Slot to Meet Faculty

The system shall allow students to schedule meeting slots with faculty members through an online booking system.

3.2.13 F13: Faculty Slot Confirmation

The system shall allow faculty members to confirm or reschedule student meeting requests.

3.2.14 F14: MU Campus Map

The system shall provide a university map to help users navigate buildings and locate faculty cabins.

3.2.15 F15: Calendar with Events & Reminders

The system shall provide a calendar view of upcoming lectures, exams, assignment deadlines, and university events.

3.2.16 F16: Live Chat

The system shall provide real-time messaging for students and faculty to communicate instantly.

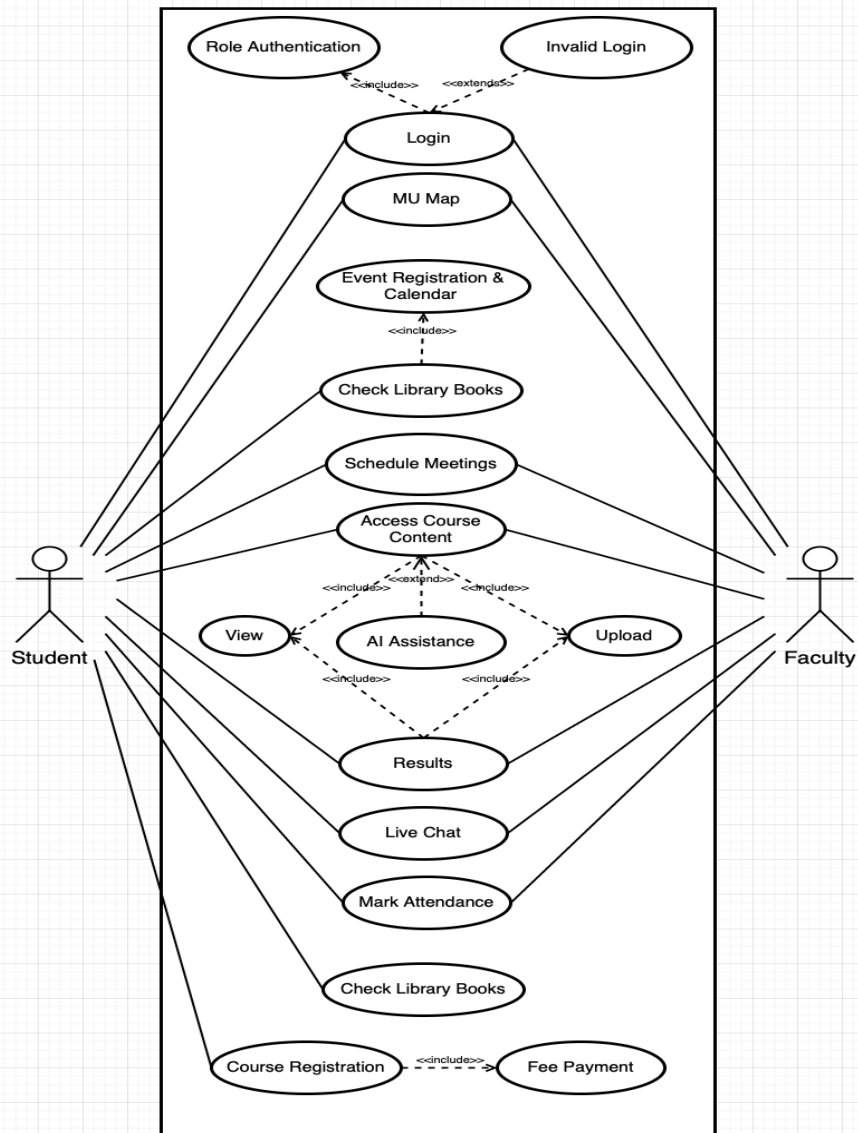
3.2.17 F17: Notification System

The system shall send push notifications and email alerts for assignments, deadlines, and announcements.

3.2.18 F18: Discussion Forums

The system shall provide a platform for students and faculty to discuss academic topics, ask questions, and share knowledge.

3.3. Use Case Model



3.3.1 Use Case 1: Login

Author: P.Sai Sarvani

Purpose: Allows students and faculty to log into the system using role-based authentication.

Requirements Traceability: System must support role-based authentication and prevent unauthorized access.

Priority: High

Preconditions: User must have valid login credentials.

Postconditions: User gains access to their respective dashboard.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. The user enters their username and password.
- b. The system verifies credentials.
- c. If valid, the user is redirected to their dashboard.

2. Alternative Flow:

- a. If the user enters incorrect credentials, the system displays an error message.

3. Exceptions:

- a. System failure or network issues prevent login.

Includes: Role Authentication, Invalid Login

Notes/Issues: Ensure secure password handling and implement CAPTCHA to prevent brute force attacks.

3.3.2 Use Case 2: Role Authentication

Author: P.Varshika

Purpose: Ensures that users are assigned correct permissions based on their role (Student or Faculty).

Requirements Traceability: System should distinguish between different user roles.

Priority: High

Preconditions: User must attempt login.

Postconditions: The user is granted access based on their role.

Actors: System (Automatic process)

Extends: None

Flow of Events:

1. Basic Flow:

- a. System checks the role assigned to the user.
- b. Redirects the user to the appropriate dashboard.

2. Exceptions:

- a. Role misclassification due to incorrect database records.

Includes: None

Notes/Issues: Ensure role-based security mechanisms are correctly implemented.

3.3.3 Use Case 3: Invalid Login

Author: P.Sai Sarvani

Purpose: Handles incorrect login attempts.

Requirements Traceability: System must display an error message when login fails.

Priority: Medium

Preconditions: User enters incorrect credentials.

Postconditions: The system prevents access and prompts for correct credentials.

Actors: Student, Faculty

Extends: Login

Flow of Events:

1. Basic Flow:

- a. User enters incorrect credentials.
- b. System displays an error message.

2. Exceptions:

- a. Multiple failed attempts may result in account lockout.

Includes: None

Notes/Issues: Implement account lockout after multiple failed attempts to enhance security.

3.3.4 Use Case 4: Event Registration & Calendar

Author: P.Vivek Reddy

Purpose: Allows students and faculty to view and register for events.

Requirements Traceability: Calendar must be updated dynamically.

Priority: High

Preconditions: User must be logged in.

Postconditions: User is registered for an event.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. User views the event calendar.
- b. Selects an event and registers.
- c. System confirms the registration.

2. Alternative Flow:

- a. If registration is full, the user is put on a waitlist.

Includes: None

Notes/Issues: Ensure real-time updates of event availability.

3.3.5 Use Case 5: View Timetable

Author: N.Shivani

Purpose: Allows users to check their schedule.

Requirements Traceability: System should fetch the timetable dynamically.

Priority: High

Preconditions: User must be logged in.

Postconditions: User views the timetable.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. User selects timetable view.
- b. System fetches and displays the schedule.

Includes: None

Notes/Issues: Ensure up-to-date schedule synchronization.

3.3.6 Use Case 6: Schedule Meetings

Author: N.Vignesh Anoop

Purpose: Allows users to arrange meetings.

Requirements Traceability: System should handle meeting scheduling conflicts.

Priority: Medium

Preconditions: User must be logged in.

Postconditions: Meeting is scheduled.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. User selects meeting slot.
- b. System verifies availability.
- c. Confirmation is sent.

2. Alternative Flow:

- a. If a slot is unavailable, the system suggests alternate timings.

Includes: None

Notes/Issues: Ensure faculty approval process if required.

3.3.7 Use Case 7: Access Course Content

Author: P.Varshika

Purpose: Enables students to access course-related notes and materials.

Requirements Traceability: Course materials must be uploaded by faculty.

Priority: High

Preconditions: User must be enrolled in the course.

Postconditions: Course content is accessed.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. Student selects a course.
- b. System displays course materials.
Includes: View, Upload
Notes/Issues: Version control for uploaded content.

3.3.8 Use Case 8: View Results

Author: P.Vivek Reddy

Purpose: Allows students to check their academic results.

Requirements Traceability: Secure access to student records.

Priority: High

Preconditions: Results must be published.

Postconditions: User views results.

Actors: Student

Extends: None

Flow of Events:

1. Basic Flow:

- a. Student selects results section.
- b. System displays grades.
Includes: None
Notes/Issues: Ensure data privacy.

3.3.9 Use Case 9: Live Chat

Author: N.Vignesh Anoop

Purpose: Enables real-time communication.

Requirements Traceability: Secure chat implementation.

Priority: Medium

Preconditions: User must be logged in.

Postconditions: Message is sent/received.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. User initiates chat.
- b. System transmits message.
Includes: None
Notes/Issues: Ensure message encryption.

3.3.10 Use Case 10: Mark Attendance

Author: P.Varshika

Purpose: Faculty marks student attendance.

Requirements Traceability: Attendance records must be stored securely.

Priority: High

Preconditions: User must be faculty.

Postconditions: Attendance is recorded.

Actors: Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. Faculty selects class.
- b. Marks student attendance.

Includes: None

Notes/Issues: Support biometric integration if needed.

3.3.11 Use Case 11: Check Library Books

Author: P.Sai Sarvani

Purpose: Allows users to check book availability.

Requirements Traceability: System must be linked to the library database.

Priority: Medium

Preconditions: User must be logged in.

Postconditions: Book availability status is displayed.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. User searches for a book.
- b. System retrieves availability status.

Includes: None

Notes/Issues: Implement reservation feature in the future.

3.3.12 Use Case 12: Maps

Author: P.Vivek Reddy

Purpose: Helps students and faculty find locations within the campus.

Requirements Traceability: Search and navigation for campus locations.

Priority: High

Preconditions: User must be logged in.

Postconditions: User successfully finds the location.

Actors: Student, Faculty

Extends: None

Flow of Events:

1. Basic Flow:

- a. User selects the **Maps** feature.
- b. User searches for a location.

- c. System displays the location with directions.
Includes: None
Notes/Issues: Ensure map data is updated regularly.

3.3.13 Use Case 13: AI Assistance

Author: N.Shivani

Purpose: Assists students with learning and faculty with grading/plagiarism checks.

Requirements Traceability: AI-powered summaries, translation, grading, and plagiarism detection.

Priority: High

Preconditions: User must have access to course content or assignments.

Postconditions: AI provides summaries, translations, grades, or plagiarism reports.

Actors: Student, Faculty

Extends: Access Course Content, Results

Flow of Events:

1. Basic Flow:

- a. **Student:** Uses AI for summaries, key phrases, synonyms, or translation.
- b. **Faculty:** Uses AI for grading and plagiarism checks.
Includes: View Course Content, Upload Assignments, Results Processing
Notes/Issues: Ensure accuracy of AI-generated content. Faculty should have the option to override AI grading.

4. Other Non-functional Requirements

4.1. Performance Requirements

The portal must handle at least 4000 concurrent users without lag to ensure seamless functionality during peak hours.

4.2. Safety and Security Requirements

- The system must implement JWT authentication and SSL encryption to protect user data and prevent unauthorized access.

4.3. Software Quality Attributes

4.3.1 Scalability

The system should be scalable to support future expansion and accommodate new features without compromising performance.

4.3.2 Usability

The user interface must be responsive and easy to navigate, ensuring users can access desired features without confusion.

4.3.3 Reliability

The attendance system should work 99.9% of the time, minimizing downtime to provide consistent service.

4.3.4 Data Backup

Automatic daily backups must be implemented to prevent data loss and maintain data integrity.

4.3.5 Maintainability

The system must follow modular coding practices, use proper documentation, and ensure easy debugging for future enhancements.

Appendix A – Data Dictionary

Name	Type	Description	Possible Values
UserID (PK)	Integer	Unique identifier for each user	Auto-generated
Username	String	Login username for authentication	User-defined
Password	String	Encrypted password for authentication	Encrypted value
Role	Enum (String)	User role	Student, Faculty, Admin
CourseID (PK, FK)	Integer	Unique identifier for each course	Auto-generated
AttendanceStatus	Enum (String)	Tracks student attendance status	Present, Absent
MessageID (PK)	Integer	Unique identifier for chat messages	Auto-generated
EventID (PK)	Integer	Unique identifier for events	Auto-generated
NotificationID (PK)	Integer	Unique identifier for notifications	Auto-generated
CreatedAt	Timestamp	Record creation time	Auto-generated
UpdatedAt	Timestamp	Last update time	Auto-updated

Appendix B - Group Log

6.1 Project Timeline (Feb 2025 - May 2025)

Week	Tasks
Week 1 (Feb 10 - Feb 16)	Project scope finalization, team role assignment, initial research
Week 2 (Feb 17 - Feb 23)	UI/UX wireframe creation, database schema design
Week 3 (Feb 24 - Mar 2)	No work due to exams
Week 4 (Mar 3 - Mar 9)	Frontend implementation, initial database connectivity
Week 5 (Mar 10 - Mar 16)	Course management feature development
Week 6 (Mar 17 - Mar 23)	Notification system and real-time chat integration
Week 7 (Mar 24 - Mar 30)	Event and notice board implementation
Week 8 (Mar 31 - Apr 6)	System testing and debugging
Week 9 (Apr 7 - Apr 13)	Performance optimization and UI refinements
Week 10 (Apr 14 - Apr 20)	Exams week
Week 11 (Apr 21 - Apr 27)	Security testing and data backup setup
Week 12 (Apr 28- May 4)	Final system testing and bug fixes
Week 13 (May 5 – May 11)	Documentation finalization, deployment, and project submission

6.2 Weekly Meeting Schedule

Meetings will be held twice a week for quick updates:

- **Monday Evening:** Progress review and task assignments for the week.
- **Friday Evening:** Status check on completed tasks and addressing challenges.