Software Requirements Specification

for

MU-UniConnect

Prepared by
Infinity 8

Group Name: Infinity 8

LAKSHMI REDDY	**SE22UCSE277**	**se22ucse277@mahindrauniversity.edu.in**
JUGAL KISHORE REDDY	**SE22UCSE272**	**se22ucse272@mahindrauniversity.edu.in**
PARTHIV REDDY	**SE22UCSE298**	**se22ucse298@mahindrauniversity.edu.in**
ADITYA VARDHAN	**SE22UCSE015**	**se22ucse015@mahindrauniversity.edu.in**
HARSHITH ANCHURI	**SE22UCSE026**	**se22ucse026@mahindrauniversity.edu.in**
ADITYA BHONAGIRI	**SE22UCSE012**	**se22ucse012@mahindrauniversity.edu.in**
GURU CHARAN REDDY	**SE22UCSE275**	**se22ucse275@mahindrauniversity.edu.in**
JAITHRA SATHWIK	**SE22UCSE116**	**se22ucse116@mahindrauniversity.edu.in**

Instructor:	***Vijay Rao***
Course:	**Software Engineering**
Lab Section:	***CSE***
Teaching Assistant:	**Vijay Rao**
Date:	**03 / March / 2025**

Contents

Contents ii
Revisions iii
1. Introduction 1
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 3
2. Overall Description 4
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
3.1. External Interface Requirements 6
3.2. Functional Requirements 6
3.3. Use Case Model 7
4. Other Non-functional Requirements 12
4.1. Performance Requirements 12
4.2. Safety and Security Requirements 12
4.3. Software Quality Attributes 12
5. Other Requirements 14
Appendix A – Data Dictionary 15
Appendix B - Group Log........17

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Team - Infinity8	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.	09/03/25

1. Introduction

The document outlines the software requirements for the College Website Integration and Engagement Platform. It defines the system's interface details and constraints.

1.1. Document Purpose

Uniconnect is a robust solution engineered to streamline communication and collaboration within educational institutions. The document defines Uniconnect as a unified digital ecosystem where students, faculty, clubs, and college management can share information and coordinate activities seamlessly. It covers the entire scope of the system, including key functionalities such as the promotion and management of club events, a dynamic calendar feature to track and coordinate events, and mechanisms to prevent scheduling conflicts and miscommunication.
By clearly outlining the operational parameters, user interactions, and system interfaces, this SRS ensures that all stakeholders from developers and system architects to educators and administrative staff are aligned on the functional and performance objectives of Uniconnect, thereby facilitating a structured, efficient, and user-centric system development process.

1.2. Product Scope

The system to be proposed is a web-based application that will act as a centralised platform for college students, faculty, clubs, and administrative personnel. The main aim of the platform is to promote student engagement and facilitate greater seamless communication within the college community.

The platform centralises club activities, event management, and recruitment while preventing scheduling conflicts. It provides real-time access to college policy, faculty details, campus maps. By improving communication and fostering community engagement, it enhances student participation and connectivity.

1.3. Intended Audience and Document Overview

This document is intended for:

- Developers: To understand system functionalities and design requirements.
- Project Managers: To oversee implementation and ensure alignment with institutional needs.
- Testers and Documentation Writers: To validate system performance and compliance with requirements, and maintaining accurate records for future reference.
- Users (Students, Faculty, Clubs, Administration): To discover platform features.

The SRS document is structured as follows:

- Introduction: Purpose, scope, and definitions.
- Overall Description: An overview of the system and its functionality.
- Specific Requirements: Detailed functional requirements, use cases, and interface specifications.
- Other Non-functional Requirements: Performance,Accessibility, Safety, Security,Reliability and Quality attributes.
- Other Requirements: Additional requirements such as database, legal and other needs.
- Appendices: Supporting documents including a data dictionary and group log.

1.4. Definitions, Acronyms and Abbreviations

API - Application Programming Interface
CSS - Cascading Style Sheets
CMS-Content Management System
COMET- Collaborative Object Modelling and Enterprise Transformation methodology
DBMS - Database Management System
GDPR-General Data Protection Regulation
HTML - Hyper Text Markup Language
IEEE-Institute of Electrical and Electronics Engineers
MFA-Multi-factor authentication
QA - Quality Assurance
RBAC-Role-based access control
SRS - Software Requirement Specifications
UAT - User Acceptance Testing
UI/UX - User Interface/User Experience
UML-Unified Modelling Language

1.5. Document Conventions

This document follows IEEE formatting standards. The conventions used are:

Font: Arial, size 11 or 12 for standard text.

Spacing: Single-spaced with 1” margins.

Titles and Subtitles: Section headings follow a hierarchical numbering system.

Italics: Used for comments and notes.

Code and Commands: Displayed in a monospaced font.

1.6. References and Acknowledgments

This report draws on several sources, including:

College administrative policies and guidelines.

IEEE Software Engineering Standards.

Past system design reports for comparable platforms.

Web design best practice and CMS documentation.

Acknowledgments:

We appreciate the inputs of our professors, project advisors, and stakeholders who gave valuable perspectives in the construction of this platform.

2. Overall Description

2.1. Product Overview

MU-UniConnect is a comprehensive Web Platform that enables students, faculty, clubs and administrators to efficiently manage campus activities and information. This system ensures real-time event updates, a dynamic user interface, and secure data management, making it an essential tool for modern college operations.

2.2. Product Functionality

The Web Platform provides two primary functional modules: the Clubs and Events Module and the College Information Module. These modules work together to ensure a seamless experience for students, faculty, and administrators, offering efficient event management, real-time communication, and easy access to essential campus information.

The Clubs and Events Module is designed to streamline the management of student clubs and activities. It allows student organisations to create and manage events, track participation, and recruit new members. A real-time notification system ensures that students remain informed about upcoming activities, competitions, and social gatherings. The module also features a conflict-detection event scheduling system, which prevents overlapping event bookings, ensuring a more organised campus experience. Additionally, clubs can post recruitment opportunities and collaborate on projects using the built-in communication tools, fostering a dynamic and interactive student community.

The College Information Module serves as a centralised knowledge hub, making it easier for students to navigate the campus and access essential resources. An interactive campus map provides a visual representation of college buildings, lecture halls, and facilities, helping students and visitors find their way efficiently. The module also includes lecturer profiles, displaying office hours, ongoing research projects, and office locations, allowing students to connect with faculty members for academic guidance.

2.3. Design and Implementation Constraints

Development of this platform shall be constrained by the following:

Software Design Methodology: The COMET methodology shall have to be applied to formal software design.
Modelling Language: System modelling shall be done using UML diagrams.
Technology Stack : The system shall be web-based and shall have to accommodate responsive design.
Security: Data encryption and authentication shall be provided and also must adhere to college data privacy policies.

Integration: Integration with current college systems shall be possible wherever needed

2.4. Assumptions and Dependencies

To ensure the success of the college website, we're making the following key assumptions:

Stable Internet Access – Users will have a reliable internet connection to access the platform smoothly.
Active Participation – Clubs, event organisers, and the administration will regularly update the platform with fresh content.
Accurate Information – The details provided about events, policies, and faculty will be correct and up to date.
Security & Privacy Compliance – Users will follow authentication rules and respect data privacy guidelines.
Cross-Device Compatibility – The website will work well on modern web browsers, laptops, and smartphones.
Scalability – The platform can handle a growing number of users without performance issues.
Regular Updates – Club leaders and event organisers will continue adding and managing event listings.
Stable College Policies – No major changes will occur in how clubs, events, and administrative updates are managed.

The project relies on the following external factors to function effectively:

Tech Stack – The platform will be built using **React.js (frontend), MongoDB (database), and Node.js (backend)**. Any issues with these technologies could impact development.
Hosting & Domain – The website will need a hosting provider and domain registration to be publicly accessible.
College Databases & APIs – we are integrating faculty profiles, hostel services and campus maps, we depend on those systems being available and updated.
User Authentication – we are us college mail login system, their availability is crucial for smooth user access.
Calendar & Scheduling Integration – The system may sync with other calendar's
Ongoing Maintenance – Regular updates and security patches will be needed to keep the website running smoothly.
Compliance with College Policies – The website must adhere to the college's security, privacy, and accessibility regulations.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

The system will have a web-based user interface with an intuitive design, allowing users to access club events, announcements, faculty details, and maintenance services through a interactive dashboard. It will be responsive for both desktop and mobile devices, ensuring seamless navigation. Key features include a dynamic calendar for event scheduling, an interactive campus map, and profile pages for lecturers and clubs. A robust search function and clear menus enhance accessibility, while user authentication ensures secure access to features.

3.1.2. Hardware Interfaces

The platform will interact with:
College servers for data storage and processing.
User devices such as laptops, tablets, and mobile phones.

3.1.3. Software Interfaces

The system will connect with:
Cloud storage services for file management.
Authentication APIs for secure user login.

3.2. Functional Requirements

Users will be able to register, login, and profile manage.
Clubs will be able to manage and create events with conflict schedule detection.
Lecturers can manage and edit their respective pages and post their projects.
Administrators will be enabled to create announcements and updates.
Students will be enabled to see faculty profiles, office hours, and project information.
There will be a hostel maintenance system that allows the users to request services and repairs.
Management can use the calendar to schedule the events without overlapping any other events, holidays, exams and other constraints according to the student counsel chatter.
Users can use an interactive map to navigate through the campus and find unknown Areas of Campus.

3.3. Use Case Model

3.3.1. Use Case 1: View College Information

Identify Team Member: S. Jaithra Sathwik
Purpose: Provide students, faculty, and admins access to essential college-related details.
Requirements Traceability: College website should provide up-to-date information on events, faculty, courses, and announcements.
Priority: High
Preconditions: User must be logged into the system.
Postconditions: User successfully retrieves college-related information.
Actors: Student, Faculty, Admin
Extends: None
Flow of Events

- Basic Flow:
    1. User logs into the system.
    2. User selects "View College Information."
    3. System retrieves and displays the required details.
- Alternative Flow:
    1. If data is not available, system displays an error message.
- Exceptions:
    1. User session timeout before retrieving information.

Use Case 2: Update College Information
Identify Team Member: S. Jaithra Sathwik
Purpose: Allow the admin to update college-related details.
Requirements Traceability: Admin should be able to modify and manage college information.
Priority: High
Preconditions: Admin must be logged in.
Postconditions: The updated college information is successfully saved in the system.
Actors: Admin
Extends: View College Information
Flow of Events

- Basic Flow:
    1. Admin logs in.
    2. Admin selects "Update College Information."
    3. Admin modifies the necessary details.
    4. System saves the updated information.
- Alternative Flow:
    1. If invalid data is entered, system prompts for correction.
- Exceptions:
    1. System fails to save the information due to a server issue.

Use Case 3: Schedule Events
Identify Team Member: S. Jaithra Sathwik
Purpose: Facilitate scheduling of college events by authorized users.
Requirements Traceability: Admin, faculty, and students should be able to schedule events.
Priority: High
Preconditions: The user must be logged in with the required permissions.
Postconditions: The scheduled event is successfully saved in the system.
Actors: Student, Faculty, Admin
Extends: None
Flow of Events

- Basic Flow:
    1. User logs in.
    2. User selects "Schedule Events."
    3. User enters event details.
    4. System checks for scheduling conflicts.
    5. If no conflicts, system saves the event.
- Alternative Flow:
    1. If a conflict is detected, system prompts for rescheduling.
- Exceptions:

1. System failure while saving the event.

Use Case 4: Club Management
Identify Team Member: S. Jaithra Sathwik
Purpose: Allow the admin and club head to manage clubs.
Requirements Traceability: Clubs should be properly managed within the system.
Priority: Medium
Preconditions: Admin or club head must be logged in.
Postconditions: Club details are successfully managed.
Actors: Admin, Club Head
Extends: None
Flow of Events

- Basic Flow:
  1. Admin/Club Head logs in.
  2. Selects "Club Management."
  3. Modifies club details (e.g., name, members).
  4. System updates the database.
- Alternative Flow:
  1. If invalid data is entered, system prompts for correction.
- Exceptions:
  1. System error prevents updating club details.

Use Case 5: Organize Club Events
Identify Team Member: S. Jaithra Sathwik
Purpose: Allow club heads and faculty to organize events.
Requirements Traceability: The system should allow organizing and managing club events.
Priority: Medium
Preconditions: Faculty or club head must be logged in.
Postconditions: The event is successfully organized and displayed in the system.
Actors: Faculty, Club Head
Extends: None
Flow of Events

- Basic Flow:
  1. Faculty/Club Head logs in.
  2. Selects "Organize Club Events."

3. Enters event details.
4. System validates and saves the event.

- Alternative Flow:
  1. If the event conflicts with another, system prompts for rescheduling.
- Exceptions:
  1. System failure while saving event details.

Use Case 6: Register for Event
Identify Team Member: S. Jaithra Sathwik
Purpose: Allow students to register for scheduled events.
Requirements Traceability: The system should support event registration.
Priority: High
Preconditions: Student must be logged in.
Postconditions: Registration is successfully recorded.
Actors: Student
Extends: View Event Details
Flow of Events

- Basic Flow:
  1. Student logs in.
  2. Views available events.
  3. Selects an event and clicks "Register."
  4. System saves the registration.
- Alternative Flow:
  1. If event is full, system denies registration.
- Exceptions:
  1. System failure during registration.

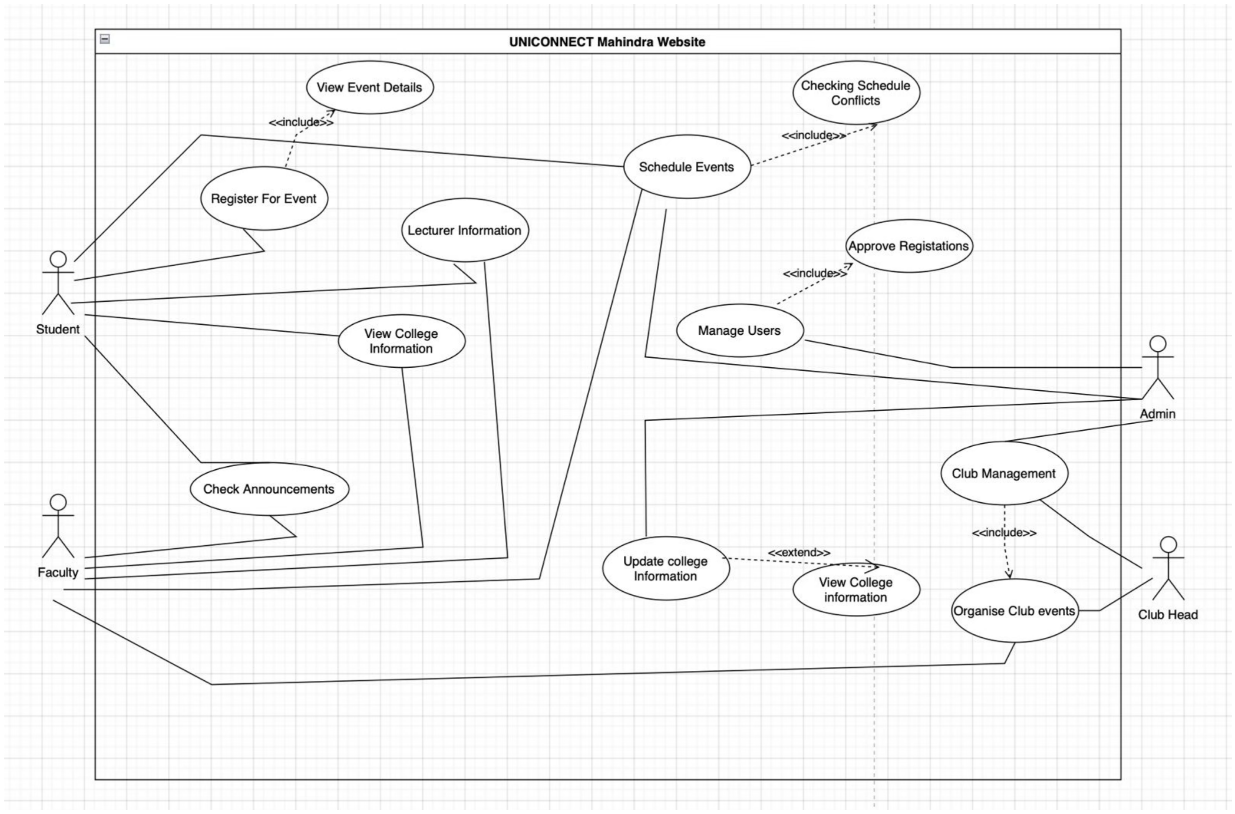
Use Case 7: Approve Registrations
Identify Team Member: S. Jaithra Sathwik
Purpose: Allow admin to approve or reject student event registrations.
Requirements Traceability: Admin should have control over event registrations.
Priority: Medium
Preconditions: Admin must be logged in.

Postconditions: The registration is approved or rejected.
Actors: Admin
Extends: None
Flow of Events

- Basic Flow:
    1. Admin logs in.
    2. Views pending registrations.
    3. Approves or rejects registrations.
    4. System updates status.

- Alternative Flow:
    1. If the event is full, system denies approval.

- Exceptions:
    1. System failure during the approval process.



4. Other Non-functional Requirements

4.1. Performance Requirements

P1: The system shall load the main dashboard within 3 seconds under normal network conditions.

P2: Event scheduling conflict detection shall be processed within 2 seconds after event creation.

P3: The system shall support at least 500 concurrent users without performance degradation.

P4: Real-time updates, such as new announcements, shall be reflected in the user interface within 5 seconds.

4.2. Safety and Security Requirements

S1: User authentication shall be required for accessing personal data and administrative functions.

S2: All sensitive data, including login credentials and user profiles, shall be encrypted using AES-256 encryption.

S3: Role-based access control (RBAC) shall be implemented to prevent unauthorised access to administrative features.

S4: The platform shall follow institutional data privacy policies to ensure compliance with security standards.

S5: Regular security audits shall be conducted to identify vulnerabilities.

4.3. Software Quality Attributes

4.3.1 Reliability

R1: The system shall have a 99.9% uptime to ensure accessibility at all times.

R2: Automatic backups shall be performed daily to prevent data loss in case of system failure.

4.3.2 Usability

U1: The platform shall have a user-friendly interface with intuitive navigation.

U2: A responsive design shall be implemented to ensure usability across different devices, including desktops, tablets, and mobile phones.

4.3.3 Maintainability

M1: The system architecture shall be modular to allow easy updates and feature enhancements.

M2: Detailed documentation shall be provided for future developers to maintain and extend the platform.

4.3.4 Security

SEC1: Multi-factor authentication (MFA) shall be available for enhanced login security.

SEC2: User sessions shall be automatically logged out after 15 minutes of inactivity.

These non-functional requirements ensure that the platform meets performance expectations, remains secure, and provides a high-quality user experience.

5. Other Requirements

Database Requirements

The application shall employ a relational database (e.g., MySQL or PostgreSQL) to maintain user information, event information, and announcements.

Backup processes shall be automated in order to avoid data loss.

Internationalization Requirements

The system must be multi-language capable to suit various student populations.

Date and time formats shall be configurable depending on user options.

Legal and Compliance Requirements

The platform will be in line with institutional data privacy regulations and laws like GDPR for protection of user data.

User data will not be transferred to third parties without express permission.

Reusability and Scalability

The system will be modularly designed so that new features can be added in the future without redesigning it.

It will be scalable to accommodate more colleges and universities in the future.

These extra requirements guarantee the platform's compliance, scalability, and usability while security and accessibility standards are upheld.

Appendix A – Data Dictionary

1.Constants

NAME	Description	Value/Example
MAX_USERS	Maximum number of users allowed	Integer (e.g., 5000)

2.State Variables

Variable Name	Description	Possible States
USER_ROLE	Defines user role	"Student", "Faculty", "Admin"
ACCOUNT_STATUS	Status of user account	"Active", "Inactive"
NOTIFICATION_TYPE	Defines type of notifications	"Event Alert", "Club Update"

3.Inputs

Input Name	Description	Related Operations
CLUB_ID	Unique identifier for each club	Used in club management system
EVENTID	Unique identifier for each event	Used for event creation & updates
USERNAME	Username for login	Used in authentication

PASSWORD	Password for authentication	Used in login process
COURSE_ID	Unique identifier for each course	Used in course registration system
MAP_LOCATION	Location for campus navigation	Used in navigation feature
FEEDBACK_ID	Unique identifier for feedback	Used for feedback tracking

4.Outputs

Output Name	**Description**	**Related Operations**
REPORT_ID	Unique identifier for reports	Used in admin reporting system
USER_COUNT	Number of active users	Displayed on dashboard

Appendix B - Group Log

1. Meeting Logs

Date	Time	Platform	Agenda	Key Discussion
01/02/2025	6:00 PM	Offline	Initial Planning	Discusses Specific Roles
02/02/2025	9:00 PM	Offline	Statement Of Work	Finished SOW
10/02/2025	6:30 PM	Offline	Planning Layouts	Discussed of Different Layouts
16/02/2025	6:00 PM	Offilne	Software Requirements	About Requirements
03/03/2025	6:00 PM	Offline	SRS Document	Finished SRS Draft

2. Group Activities & Contributions

Date	Task/Activity	Members Involved	Status	Notes
19/02/2025	Research on Project Requirements	All Members	Completed	Research on entire data
28/02/2025	Functional and non Functional Requirements	All Members	Completed	Have reveiw Form everyone
01/03/2025	System Design Documentation	All Members	Completed	Defined system architecture

05/03/2025	Security & Performance analysis	All Members	Completed	Identified key Security requirements
09/03/2025	Final Review	All Members	Completed	Final Take on SRS