**SOFTWARE REQUIREMENTS SPECIFICATION**or

**COLLEGE CLUB MANAGEMENT SYSTEM**

**Prepared by,**

**Aakaash Sanjeevi K S(23CSR002)**

**Dharshini S(23CSR047)**

**Divya A(23CSR056)**

**KONGU ENGINEERING COLLEGE, ERODE**

**DATE: 27.02.2025**

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 System Feature 1 4

3.2 System Feature 2 (and so on) 5

4. External Interface Requirements 5

4.1 User Interfaces 5

4.2 Hardware Interfaces 5

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

6. Other Requirements 6

Appendix A: Glossary 7

Appendix B: Analysis Models 7

Appendix C: Issues List 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **1.Introduction**

**1.1** **Purpose**

The purpose of this project is to develop a College Club Management System that simplifies club administration by providing a centralized platform with role-based access. It streamlines membership management, event registration, RSVP handling, and discussions, enhancing collaboration and efficiency for students and administrators.

## **1.2 Document Conventions**

This document follows a clear and structured format for easy readability. Headings are in bold, while *important terms* are italicized for emphasis. Code snippets and system names are written in monospace font. Each requirement is listed separately, with priorities clearly mentioned. Numbering is used for hierarchical organization, and figures include captions.

## **1.3 Intended Audience and Reading Suggestions**

This document is intended for college administrators, club coordinators, developers, and students involved in managing college clubs and events. Administrators and club coordinators should focus on system features and role-based access. Developers should refer to technical sections for implementation details. Students can review membership and event management features.

## **1.4** **Project Scope**

The College Club Management System streamlines club administration with a centralized platform for memberships, events, discussions, and RSVPs. It includes role-based access, event scheduling, notifications, and discussion forums. A single login ensures seamless access, improving collaboration, reducing manual work, and enhancing club management within the college.

**1.5 References**

1. User Interface Design Guidelines – Standard UI principles for web-based management systems.
2. Role-Based Access Control (RBAC) Standards – Ensuring secure and efficient user access management.
3. Software Requirement Specification (SRS) Template – IEEE Standard 830-1998 for documenting requirements.
4. Official Documentation for Web Technologies – References for HTML, CSS, JavaScript, and Django frameworks.

**2. Overall Description**

The College Club Management System simplifies managing clubs, events, and discussions through a centralized digital platform. It addresses manual processes, tracking issues, and system inefficiencies. Role-based access allows administrators, coordinators, and students to manage memberships, events, RSVPs, and discussions. A single login enhances collaboration, reduces workload, and ensures secure, efficient operations.

## **2.1** **Product Perspective**

The College Club Management System is a web-based application that centralizes club activities. It works independently but can integrate with college portals for authentication. A modular architecture ensures scalability and flexibility. Role-based access allows administrators, coordinators, and students to manage tasks efficiently. It enhances accessibility, automates event management, and improves communication. Modern web technologies provide a user-friendly interface and secure data management.

## **2.2 Product Features**

• **User Authentication:** Secure login system for administrators, coordinators, and students.  
• **Role-Based Access:** Permissions based on user roles for efficient management.  
• **Membership Management:** Tracks club registrations and member details.  
• **Event Scheduling:** Organizes and manages club events with RSVP tracking.  
• **Discussion Forums:** Facilitates communication and collaboration among members.  
• **Notifications & Alerts:** Sends updates on events, announcements, and discussions.

## **2.3** **User Classes and Characteristics**

• **Students:** Join clubs, register for events, participate in discussions, and receive notifications.  
• **Club Coordinators:** Manage club memberships, schedule events, moderate discussions, and track participation.  
• **Administrators:** Oversee all clubs, manage user roles, approve event and ensure security.  
• **Faculty Advisors**: Monitor club activities, provide guidance, and review event proposals.

## **Operating Environment**

The College Club Management System is a web-based application accessible through Chrome, Firefox, Edge, and Safari. It supports Windows, macOS, Linux for desktops, and Android and iOS for mobile devices. The system uses MySQL or PostgreSQL for data storage, with Django or Node.js handling backend processing. The frontend, built with HTML, CSS, and React, ensures a responsive interface. It can be hosted on cloud platforms like AWS or Azure or self-hosted. Security measures include HTTPS, role-based access control, and encrypted storage.

## 

## **2.4 Design and Implementation Constraints**

• The system must be web-based and compatible with major browsers and devices to ensure accessibility.  
• User permissions must be strictly enforced through role-based access control for secure operations.  
• The database should be scalable to accommodate increasing users and data without performance issues.  
• Security measures, including encryption and compliance with data protection regulations, must be implemented.

## **2.5 User Documentation**

A user manual will provide step-by-step instructions for students, coordinators, and administrators. Online help sections will include FAQs, troubleshooting tips, and feature explanations to assist users. Video tutorials will guide users through key functions such as event registration and membership management. Additionally, a contact support option will be available for users requiring further assistance.

## **2.6 Assumptions and Dependencies**

• Users are assumed to have basic digital literacy to navigate the platform efficiently.  
• A stable internet connection is required for seamless access and functionality.  
• The system depends on third-party services for authentication, cloud hosting, and notifications.  
• Integration with existing college portals may be necessary for user authentication and data synchronization.

# **3. System Features**

• User Authentication: Secure login for students, coordinators, and administrators with role-based access control.  
• Membership Management: Allows students to join clubs, track membership status, and manage club details.  
• Event Scheduling: Enables club coordinators to create, update, and manage events with RSVP tracking.  
• Discussion Forums: Provides a platform for members to engage in club-related discussions and collaborations.  
• Notifications and Alerts: Sends automated updates on upcoming events, announcements, and important discussions.  
• Data Security: Ensures encrypted storage and secure access to protect user and club information.

## **3.1** **System Feature 1**

### **3.1.1 Description and Priority**

1. User Authentication (High): Secure login for students, coordinators, and administrators with role-based access control.
2. Membership Management (High): Allows students to join clubs, track membership, and manage club details.
3. Event Scheduling (High): Enables coordinators to create, update, and manage events with RSVP tracking.
4. Discussion Forums (Medium): Provides a platform for members to engage in club-related discussions.
5. Notifications and Alerts (Medium): Sends automated updates on events, announcements, and discussions.
6. Data Security (High): Ensures encrypted storage and secure access to protect user and club information.

### **3.1.2 Stimulus/Response Sequences•**

### **User Authentication**: User logs in → System verifies and grants/denies access. **• Membership Management:** Student joins a club → System updates membership records. **• Event Scheduling:** Coordinator creates an event → System notifies club members. **• Discussion Forums: User posts a message →** System updates and displays it. **• Notifications & Alerts: Event is scheduled →** System sends reminders to members. **• Data Security: Unauthorized access attempt →** System denies access and logs the attempt.

### **3.1.3 Functional Requirements REQ-1:** Users must log in with valid credentials; invalid attempts should show an error message**. REQ-2:** Students can join or leave clubs; system updates membership records accordingly. **REQ-3:** Coordinators can create, update, or cancel events; system notifies relevant users**. REQ-4:** Users can post and reply to discussions; system ensures real-time updates. **REQ-5**: System must send automated notifications for upcoming events and announcements.

## **3.2** **System Feature 2 (and so on)**

**Description and Priority:**

Students can join, leave, and track club memberships. (High Priority)

**Stimulus/Response Sequences:**  
• Student selects a club to join → System updates membership records.  
• Student leaves a club → System removes membership and updates records.

**Functional Requirements:**  
• Only registered users can join or leave clubs.  
• System prevents duplicate memberships and maintains accurate records.  
• Club coordinators can view and manage member lists.

# **4.** **External Interface Requirements**

**4.1 User Interfaces**

The system will have a web-based interface with a responsive design, ensuring compatibility with desktops, tablets, and mobile devices. It will feature a dashboard for students, club coordinators, and administrators, with intuitive navigation menus, buttons, and forms for seamless interaction. Each screen will maintain a consistent layout, including a header, sidebar, and main content area. Error messages will be clear and user-friendly. A separate User Interface Specification Document will outline detailed design guidelines.  
  
**4.2 Hardware Interface**  
The system will run on desktops, laptops, tablets, and mobile devices with internet access. It requires servers for hosting the backend (Django/Node.js) and a database server (MySQL/PostgreSQL) for storing club-related data. The system will support standard input devices such as keyboards, mice, and touchscreens for seamless user interaction.

**4.3 Software Interface**  
The system will integrate with web browsers such as Chrome, Firefox, and Edge for accessibility. It will use APIs for third-party services like email notifications and authentication. The system will connect with a MySQL or PostgreSQL database for data storage and retrieval. Role-based access control will be implemented to ensure secure user authentication and data protection.

**4.4 Communications Interfaces**  
The system will use HTTPS to ensure secure communication between users and the server. It will support RESTful APIs for efficient data exchange between the frontend and backend. Email and SMS notifications will be sent using SMTP and third-party messaging APIs. The database server will communicate with the backend using SQL queries over a secure and encrypted connection.

# **5.** **Other Nonfunctional Requirements**

**5.1 Performance Requirements**  
• The system should support at least 1000 concurrent users without significant delays.  
• Club membership updates and event registrations should process within 2 seconds.  
• Notifications should be sent within 5 seconds of an event update.  
• The system should maintain 99.9% uptime for reliable access.  
• Database queries should be optimized to retrieve records in under 1 second.

**5.2 Safety Requirements**  
• Data encryption must be implemented to protect user and club information from unauthorized access.  
• The system should have automatic backups to prevent data loss.  
• Role-based access control (RBAC) must restrict sensitive data to authorized users.  
• The system must comply with standard data security protocols to ensure user privacy.  
• Error handling should prevent system crashes and unauthorized data modifications.

**5.3 Security Requirements**  
• User authentication must be implemented using secure login methods (e.g., OTP, password hashing).  
• Role-based access control (RBAC) should restrict access to club management features.  
• Data encryption (AES, SSL/TLS) must be used to protect user and club information.  
• The system must comply with standard data privacy and security regulations.  
• Audit logs should track all user actions for accountability and monitoring.

**5.4 Software Quality Attributes**  
• **Reliability:** The system should maintain 99.9% uptime for uninterrupted access.  
• **Security:** Data must be encrypted and protected with role-based access control.  
• **Usability:** The interface should be intuitive and easy to navigate for all users.  
• **Scalability:** The system should efficiently support an increasing number of users and events.  
• **Maintainability:** Code should be modular, well-documented, and easy to update or extend.

**6. Other Requirements**

• **Database Requirements:** The system will use MySQL or PostgreSQL for structured data storage with optimized queries.  
• **Internationalization:** Support for multiple languages may be added in future updates.  
• **Legal Compliance:** The system must adhere to data privacy regulations like GDPR.  
• **Reuse Objectives:** Modular design will allow reuse of authentication and event management components.

Appendix A: Glossary

• **RBAC:** Role-Based Access Control (security model for managing user permissions).  
• **API:** Application Programming Interface (enables communication between system components).  
• **SQL:** Structured Query Language (used for managing relational databases).  
• **UI:** User Interface (the visual and interactive elements of the system).  
• **GDPR:** General Data Protection Regulation (data privacy law for user information protection).

**Appendix B: Analysis Models**  
• **Data Flow Diagram (DFD):** Illustrates system interactions and data movement.  
• **Entity-Relationship Diagram (ERD):** Defines the database structure and relationships.  
• **Class Diagram:** Represents backend components and their relationships.

**Appendix C: Issues List**  
• **TBD:** Decision on third-party email/SMS notification service integration.  
• **Pending:** Finalizing user roles and access levels for different club members.  
• **Open Issue:** Selection of encryption algorithm for securing user and event data.