Software Requirements Specification

UNISYNC

Version 0.1

Prepared by

Rishika Gaja	SE22UCSE308	se22ucse308@mahindrauniversity.edu.in
Samyukta Gade	SE22UCSE096	se22ucse096@mahindrauniversity.edu.in
Kushala K	SE22UCSE148	se22ucse148@mahindrauniversity.edu.in
Pavan Koushik	SE22UCSE326	se22ucse326@mahindrauniversity.edu.in
Rishi Tez Reddy Dhava	SE22UCSE323	se22ucse323@mahindrauniversity.edu.in
Abhinav Pulluri	SE22UCSE006	se22ucse006@mahindrauniversity.edu.in
N Vaibhav Naidu	SE22UCSE282	se22ucse282@mahindrauniversity.edu.in
G Sai Manasa	SE22UCSE104	se22ucse104@mahindrauniversity.edu.in

Instructor:	Vijay Rao Duddu
Course:	Software Engineering
Lab Section:	IT Lab – 2 FF (Computer Science Engineering)
Teaching Assistant:	Murali Krishna Bukkasamudram
Date:	10.03.2025(Date of Submission)

Contents

1 Introduction 3
1.1 Document Purpose 3
1.2 Product Scope 3
1.3 Intended Audience and Document Overview 4
1.4 Definitions, Acronyms and Abbreviations 4
1.5 Document Conventions 4
1.6 References and Acknowledgments 5
2 Overall Description 5
2.1 Product Overview 5
2.2 Product Functionality 6
2.3 Design and Implementation Constraints 6
2.4 Assumptions and Dependencies 7
3 Specific Requirements 7
3.1 External Interface Requirements 7
3.2 Functional Requirements 8
3.3 Use Case Model 10
4 Other Non-functional Requirements 12
4.1 Performance Requirements 12
4.2 Safety and Security Requirements 12
4.3 Software Quality Attributes 13
5 Other Requirements 14
Appendix A – Data Dictionary 15
Appendix B - Group Log 16

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	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.	00/00/00

1 Introduction

The project is a web-hosted platform designed for the staff of Mahindra University (MU) to share and synchronize their calendars seamlessly. The platform aims to facilitate efficient scheduling, reduce conflicts in meetings, and enhance coordination among faculty and staff members. The staff or the students can only login to the web portal with their individual university ID. Staff can update their calendar and the busy days or slots will be color coded accordingly that automates task management and optimizing meeting schedules. Two or more individuals or groups can merge and sync their calendars to find the empty slots. We are providing a simple one to one text messaging between the sync calendar groups to get notified about the meeting possibility. The students can view the faculty calendar, and the faculty with their colleagues.

1.1 Document Purpose

This document serves as the Software Requirements Specification (SRS) for the Mahindra University Staff Calendar Management System. It defines the functional and non-functional requirements for the development of a web-based platform that enables faculty and staff members to share, synchronize, and manage their calendars seamlessly.

The purpose of this document is to:

- Clearly outline the system's objectives, features, and constraints.
- Provide a reference for developers, testers, and stakeholders to ensure the system meets the intended requirements.
- Facilitate efficient scheduling by minimizing conflicts in meetings and enhancing coordination among university staff.

1.2 Product Scope

- The core purpose of the software is to synchronize the calendars of individuals or groups (that includes Faculty and organizational staff)
- Easy for the faculty to conduct meetings without any inconvenience with their schedule.

- The students can view the calendar and meet the faculty in the available slots.
- The one-to-one messaging helps with a short and clear way of expressing staff interest with the meeting.

1.3 Intended Audience and Document Overview

This document is intended for the following stakeholders involved in the development, deployment, and usage of the Mahindra University Staff Calendar Management System:

- Developers: To understand the functional and non-functional requirements necessary for implementing the system effectively.
- Project Managers: To track project progress, ensure alignment with requirements, and oversee the development timeline.
- University Faculty and Staff: As primary users, they will benefit from the platform's features, including calendar synchronization and meeting coordination.
- Students: To view faculty availability and schedule meetings accordingly.
- Testers: To verify and validate that the system meets the specified requirements and performs as expected.
- Instructors and Teaching Assistants: To evaluate the project, provide feedback, and ensure that the system aligns with software engineering principles.
- University IT Administrators: To deploy, maintain, and manage system configurations.

1.4 Definitions, Acronyms and Abbreviations

- MU - Mahindra University
- RBAC - Role-Based Access Control
- CRUD - Create, Read, Update, Delete
- API - Application Programming Interface
- FCM - Firebase Cloud Messaging
- OAuth 2.0 - Open Authorization 2.0
- SSL/TLS - Secure Sockets Layer / Transport Layer Security
- FERPA - Family Educational Rights and Privacy Act
- UML - Unified Modeling Language
- HTML - HyperText Markup Language
- CSS - Cascading Style Sheets
- UI - User Interface
- IT - Information Technology
- DB - Database
- CPU - Central Processing Unit
- RAM - Random Access Memory

1.5 Document Conventions

This document follows standard formatting and structuring guidelines to ensure consistency and readability. The conventions used in this Software Requirements Specification (SRS) are as follows:

Formatting Conventions

- The document adheres to IEEE SRS formatting standards.
- Font Style & Size: The primary text is written in Arial, size 11 or 12.
- Text Spacing & Margins: The document is single-spaced with 1-inch margins on all sides.
- Section and Subsection Titles: Follow the numbering and hierarchical structure used in this template (e.g., 1, 1.1, 1.1.1).
- Italics: Used for comments or placeholder text that needs to be updated.
- Bold: Used to highlight important terms, section headers, and key concepts.

Naming Conventions

- All system-related entities (e.g., "User", "Staff", "Calendar Event") are written in Title Case.
- Variable and database attributes (if referenced) follow camelCase or snake_case notation, as per coding standards.
- User roles such as Faculty, Student, and Administrator are capitalized for clarity.

1.6 References and Acknowledgments

References

The following documents and resources were referenced during the preparation of this SRS:

- IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications
- Mahindra University IT Policies and Guidelines
- Software Engineering: A Practitioner's Approach – Roger S. Pressman
- Web development best practices and calendar synchronization techniques
- User Interface Design Principles and Guidelines

Acknowledgments

We would like to express our gratitude to the following individuals for their valuable guidance and support throughout the development of this project:

- Vijay Rao Duddu – Instructor, for providing direction and feedback on project development.
- Murali Krishna Bukkasamudram – Teaching Assistant, for assisting with technical queries and project evaluation.
- Mahindra University Faculty and IT Team – For their input on user requirements and system feasibility.
- Our Peers and Group Members – For their collaboration, dedication, and contributions to the successful documentation of this project.

2 Overall Description

2.1 Product Overview

The UNISYNC Calendar Management System is a new web-based tool designed for Mahindra University faculty, staff, and students to efficiently schedule meetings and avoid conflicts. It allows faculty to sync calendars, students to check availability, and everyone to coordinate easily. This system saves time, reduces scheduling hassles, and improves coordination across the university.

How It Works

- Faculty & staff sync, share and manage their calendars.
- Students check faculty availability before requesting meetings.
- Built-in messaging helps coordinate schedules.
- University authentication ensures secure access.

2.2 Product Functionality

The UNISYNC Calendar Management System provides the following core functionalities, ensuring seamless scheduling and coordination among faculty, staff, and students:

- User Authentication & Authorization – Secure login and role-based access control (RBAC) using university credentials.
- Calendar Management – CRUD (Create, Read, Update, Delete) operations for calendar events, ensuring dynamic scheduling.
- Calendar Synchronization – Multi-user event synchronization to identify overlapping free time slots.
- Student-Faculty Interaction – Read-only access for students to view faculty availability and request appointments.
- Inter-User Messaging – Real-time, one-on-one text messaging within synchronized calendar groups.
- Event Status Indication – Color-coded visualization for availability, pending meetings, and confirmed schedules.
- Notification System – Automated alerts for meeting requests, confirmations, and schedule updates.
- Role-Based Feature Access – Faculty, staff, and students have different privileges based on their user roles.

2.3 Design and Implementation Constraints

- Methodology & Modeling: The system must follow the COMET method for structured software design and use UML diagrams for modeling.
- Hardware Limitations: Runs on cloud servers with 8 CPU cores, 32GB RAM, and 500GB storage; client devices need 4GB RAM, HTML5 support.
- Software Constraints: Uses Firebase and Node.js for the backend, HTML, CSS, React.js and Next.js for the frontend, and PostgreSQL as the database.
- Integration Needs: Must support Google Calendar API for synchronization.
- Security Requirements: Implements OAuth 2.0 for secure communication.
- Scalability & Deployment: Supports load balancing with Verse, Firebase, and ensures compliance with FERPA privacy policies.
- Concurrency Handling: Enables parallel user operations without data conflicts

2.4 Assumptions and Dependencies

Assumptions

1. University Infrastructure Support – Mahindra University provides the necessary hosting, authentication mechanisms, and API access for seamless integration.
2. Consistent Internet Access – The system assumes reliable internet connectivity for real-time calendar updates and messaging features.
3. User Familiarity with Web Interfaces – Faculty, staff, and students are expected to have basic familiarity with online scheduling tools, reducing the need for extensive user training.
4. Limited Concurrent Users (Initial Phase) – In early deployment, we expect a manageable number of users. Scalability testing will be needed for future university-wide adoption.
5. Standardized University Email Authentication – The system assumes that every faculty, staff, and student has a university-provided email ID that follows a standard format for authentication.

Dependencies

1. Google Calendar API Integration – The system relies on third-party APIs for calendar synchronization. Any changes in Google's API policies may require adjustments.
2. Authentication via OAuth 2.0 – Secure login and user verification are dependent on OAuth 2.0 implementation. If an OAuth provider changes, security configurations need updates.
3. Backend and Database Compatibility – The platform is built on Node.js with Firebase and PostgreSQL. Any shifts in these technologies may impact performance and data consistency.
4. Frontend Frameworks – The UI depends on React.js and Next.js for dynamic interaction. Updates in these frameworks may require UI refactoring.
5. Cloud Hosting & Load Balancing – The system is hosted on cloud servers with Vercel and Firebase. Scalability depends on the cloud provider's infrastructure limits.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Responsive Web Design: The website is accessible across desktops, tablets, and mobile devices, adapting to different screen sizes.
- Intuitive Navigation: A clean and structured menu helps users find relevant information and features effortlessly.
- User Authentication: Users log in with secure credentials, ensuring access to personalized content.
- Dashboard Interface: A centralized dashboard provides an overview of key features, updates, and user-specific controls.
- Interactive Elements: Forms, buttons, and dropdown menus allow users to navigate, search, and interact with the system efficiently.
- Role-Based Access: Different users (e.g., students, faculty, admins) have tailored access to specific sections based on their roles.

3.1.2 Hardware Interfaces

- Client-Side Hardware:
    1. Desktops, Laptops, Tablets, and Smartphones: Users can access the Unisync platform through modern web browsers on any internet-enabled device.
    2. Input Devices: The system supports keyboard, mouse, and touchscreens for user interactions.
- Server-Side Hardware:
    1. Web Server: The website is hosted on a dedicated or cloud-based server, which handles user requests and data processing.
    2. Database Server: A relational database system stores user data, schedules, and other platform information.
    3. Storage Systems: Cloud or local storage solutions manage file uploads, documents, and media related to the platform.
- Network Interfaces:
    1. The platform requires a stable internet connection (Wi-Fi, Ethernet, or mobile data) for real-time access and updates.

The UNISYNC website does not rely on any specialized hardware beyond standard web infrastructure and compatible user devices.

3.1.3 Software Interfaces

- API Communication: Uses RESTful APIs (or WebSockets for real-time updates) to manage authentication, schedule syncing, notifications, and commands.
- Database Integration: Both website and app interact with a centralized database to ensure real-time data consistency for user preferences, schedules, and system updates.
- Authentication & Security: Secure access via OAuth 2.0 with encrypted data transmission using HTTPS and SSL/TLS.
- Push Notifications: Website triggers alerts via Firebase Cloud Messaging (FCM) to keep users informed.
- Seamless Experience: Ensures a smooth, secure, and synchronized user experience across the UNISYNC website and mobile app.

3.2 Functional Requirements

3.2.1 F1: User Authentication & Access Control

- The system shall allow users to register, log in, and log out securely.
- It shall support authentication via OAuth 2.0.
- Users shall have role-based access (e.g., Admin, Standard User).

3.2.2 F2: Data Synchronization

- The system shall sync user preferences, schedules, and system settings across the website and mobile app in real time.
- Any changes made on one platform shall be instantly reflected on the other.

3.2.3 F3: User Dashboard & Control Panel

- The website shall provide an interactive dashboard displaying key system data.
- Users shall be able to modify schedules, update preferences, and manage settings through a user-friendly interface.

3.2.4 F4: Notifications & Alerts

- The system shall send real-time notifications via email, push notifications (FCM).
- Users shall receive alerts for system updates, schedule changes, or security warnings.

3.2.5 F5: Security & Data Protection

- The system shall encrypt sensitive user data using SSL/TLS protocols.
- It shall enforce secure password policies and multi-factor authentication (MFA) for added protection.

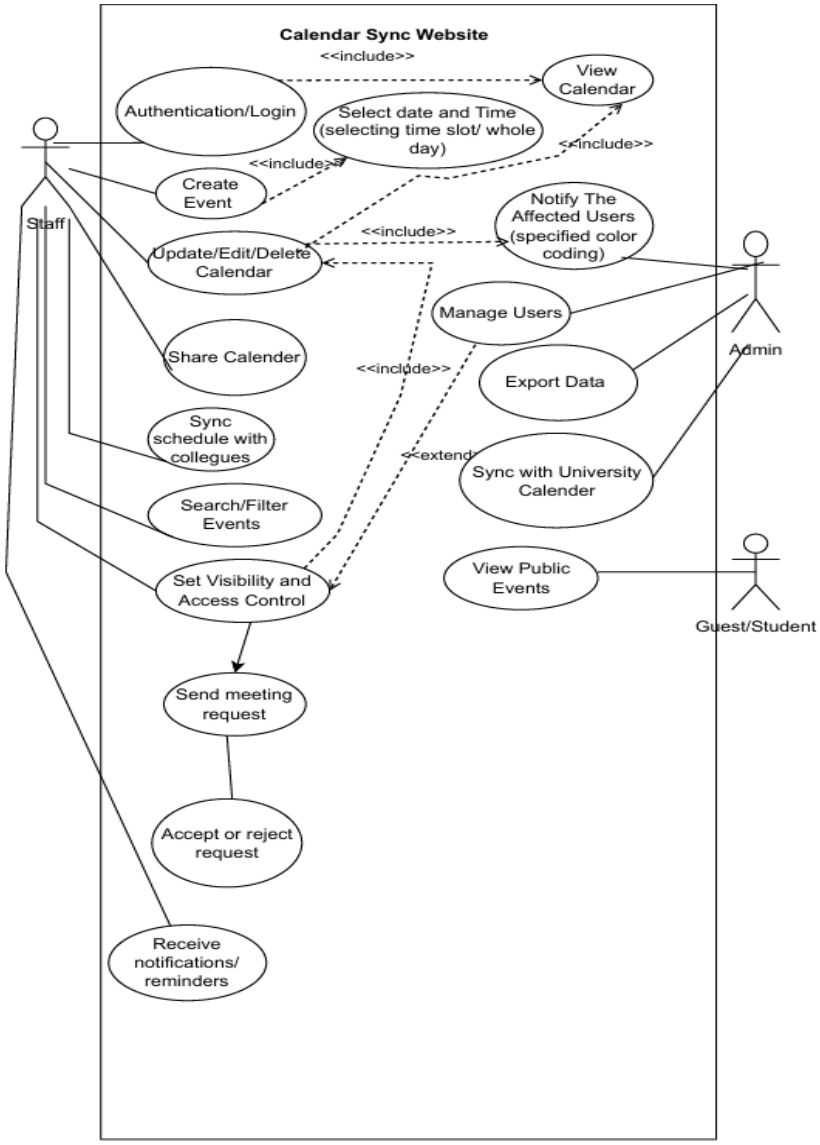
3.2.6 F6: API & Third-Party Integrations

- The system shall provide a RESTful API for seamless integration with third-party services.
- It shall support external connections such as cloud storage, IoT devices, or analytics tools.

3.2.7 F7: Logging & Error Handling

- The system shall maintain activity logs for tracking user actions.
- It shall provide error-handling mechanisms and system diagnostics for troubleshooting.

3.3 Use Case Model



3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

Author – The whole team contributed to make the UML complete.

Purpose - This use case allows a Staff user to create an event in the calendar system. The objective is to ensure users can schedule events efficiently while notifying affected users and ensuring visibility control.

Requirements Traceability:

- F1: User Authentication & Access Control

- F2: Data Synchronization
- F3: User Dashboard & Control Panel
- F4: Notifications & Alerts

Requirements Traceability –

- F1: User Authentication & Access Control – Users must be logged in before they can create an event.
- F2: Data Synchronization – Once an event is created, it must be synchronized across all connected devices and users.
- F3: User Dashboard & Control Panel – The newly created event should be visible and manageable within the calendar interface.
- F4: Notifications & Alerts – Affected users must receive notifications about the event creation and updates.
- F5: Calendar Management – Users must be able to specify event details such as title, time, visibility, and description.
- F6: Conflict Resolution – The system should check for scheduling conflicts and notify users accordingly.

Priority - High

Preconditions -

- The user must be logged in and have the appropriate permissions to create an event.
- The calendar system must be accessible and functional.

Post conditions -

- A new event is successfully created and stored in the system.
- Affected users are notified (if applicable).
- The event appears in the calendar with proper visibility settings.

Actors –

- Staff (Primary Actor)
- System (Validates and saves the event, triggers notifications)
- Students (View the faculty calendar)

Extends – Update/Edit/Delete Calendar (U2) (if the user modifies the event later)

Flow of Events

1. Basic Flow -
    - Staff logs into the Calendar Sync Website.
    - Navigates to the Create Event section.
    - Selects the date and time slot for the event.
    - Enters event title, description, and visibility settings.
    - Adds affected users (if applicable).
    - Clicks Save to create the event.
    - The system stores the event in the database.

- The system notifies affected users with color-coded notifications.

2. Alternative Flow -
   - If the user selects "All Day Event," the system adjusts the time slot automatically.
   - If no affected users are added, the event is created without notifications.
3. Exceptions -
   - If the user session expires, they must re-authenticate before creating an event.
   - If the database connection fails, the system displays an error and prevents event creation.
   - If the event conflicts with an existing one (based on visibility rules), the system prompts the user with a warning.

Includes -

- Authentication/Login (U3) (User must be logged in before creating an event)
- Select Date and Time (U4) (User must specify event timing)
- Notify Affected Users (U5) (System sends notifications after event creation)

Notes/Issues -

- Consider adding event categories or tags for better organization.
- Ensure timezone support for remote users.

4 Other Non-functional Requirements

4.1 Performance Requirements

P1: The system should handle up to 500 concurrent users without a significant drop in response time (less than 2 seconds for event synchronization).

P2: Calendar synchronization should complete within 3 seconds of an update being made.

P3: The messaging feature should ensure message delivery within 1 second under normal network conditions.

P4: The platform should maintain 99.9% uptime except for scheduled maintenance.

P5: The system should support auto-scaling to accommodate peak usage periods, such as university enrollment weeks.

4.2 Safety and Security Requirements

S1: Authentication should be done using OAuth 2.0 with university credentials, preventing unauthorized access.

S2: The system must implement role-based access control (RBAC) to limit actions based on user roles (students, faculty, administrators).

S3: All sensitive data (calendar events, messages) must be encrypted at rest and in transit using AES-256 and TLS 1.2+.

S4: Users must be automatically logged out after 30 minutes of inactivity for security purposes.

S5: The system should have audit logs to track login attempts, failed authentications, and calendar modifications.

S6: FERPA compliance must be ensured for handling faculty and student data.

4.3 Software Quality Attributes

4.3.1 Reliability

- R1: The system should be fault-tolerant with automatic failover mechanisms in case of server failure.
- R2: Calendar data should be backed up daily to prevent data loss.
- R3: The platform should recover from failures within 5 minutes through automated recovery scripts.

4.3.2 Usability

- U1: The UI should follow responsive design principles, ensuring usability across desktop, tablet, and mobile devices.
- U2: The interface should provide tooltips and error messages for better user experience.
- U3: A simple, intuitive drag-and-drop calendar interface should be provided for ease of scheduling.

4.3.3 Maintainability

- M1: The system should be built using modular microservices architecture, enabling easier debugging and updates.
- M2: Code should follow standard naming conventions and documentation guidelines to aid future maintenance.
- M3: Updates should be backward compatible, ensuring users do not experience disruptions during upgrades.

4.3.4 Scalability

- SC1: The platform should handle a 10x increase in users with horizontal scaling.
- SC2: The system should support multi-tenant architecture, allowing additional institutions to onboard in the future.

5 Other Requirements

<This section is Optional. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>