**Software Design Specifications**

**for**

**<** **Website for staff of MU to share and synch their calendars for conducting meetings, push academic calendar>**

**PREPARED BY:**

**TEAM 4**

**Document Information:**

|  |  |
| --- | --- |
| **Title:** | **Document Version Date:** |
| **Staff calanders&meetings** | **31-03-2025** |
| **Prepared By:** | **Preperation Date:** |
| **Team 4** | **02-04-2025** |

**Version History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Date** | **Revised By** | **Description** | **File Name** |
| **0.1** | **03.03.25** | **Ghanashyam, Nihal** | **Initial draft created with outline** | **SDS\_V0.1\_Draft.docx** |
| **0.5** | **10.03.25** | **Abhinav,**  **Rohan** | **Added use case view & portal design overview** | **SDS\_V0.5\_UseCase\_Design.docx** |
| **1.0** | **24.03.25** | **Vaishnavi,**  **Hanum** | **Completed first full version with all sections** | **SDS\_V1.0\_Complete.doc** |
| **1.1** | **31.03.25** | **Sreehitha, Bhanu** | **Minor updates in Exception Handling & Parameters** | **SDS\_V1.1\_MinorUpdate.docx** |

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **1 INTRODUCTION ............................................................................................................................................................** | | **4** |
| 1.1 | PURPOSE ...................................................................................................................................................................... | 4 |
| 1.2 | SCOPE .......................................................................................................................................................................... | 4 |
| 1.3 | DEFINITIONS, ACRONYMS, AND ABBREVIATIONS ........................................................................................................ | 4 |
| 1.4 | REFERENCES ................................................................................................................................................................ | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| **2** | **USE CASE VIEW ............................................................................................................................................................** | | **4** |
|  | 2.1 | USE CASE ..................................................................................................................................................................... | 4 |
| **3** | **DESIGN OVERVIEW .....................................................................................................................................................** | | **4** |
|  | 3.1 | DESIGN GOALS AND CONSTRAINTS ............................................................................................................................. | 5 |
|  | 3.2 | DESIGN ASSUMPTIONS ................................................................................................................................................. | 5 |
|  | 3.3 | SIGNIFICANT DESIGN PACKAGES ................................................................................................................................. | 5 |
|  | 3.4 | DEPENDENT EXTERNAL INTERFACES ........................................................................................................................... | 5 |
|  | 3.5 | IMPLEMENTED APPLICATION EXTERNAL INTERFACES ................................................................................................. | 5 |
| **4** | **LOGICAL VIEW .............................................................................................................................................................** | | **5** |
|  | 4.1 | DESIGN MODEL ............................................................................................................................................................ | 6 |
|  | 4.2 | USE CASE REALIZATION .............................................................................................................................................. | 6 |
| **5** | **DATA VIEW ....................................................................................................................................................................** | | **6** |
|  | 5.1 | DOMAIN MODEL .......................................................................................................................................................... | 6 |
|  | 5.2 | D ATA MODEL (PERSISTENT DATA VIEW)...................................................................................................................... | 6 |
|  | *5.2.1 Data Dictionary....................................................................................................................................................* | | *6* |
| **6** | **EXCEPTION HANDLING .............................................................................................................................................** | | **6** |
| **7** | **CONFIGURABLE PARAMETERS ..............................................................................................................................** | | **6** |
| **8** | **QUALITY OF SERVICE ................................................................................................................................................** | | **7** |
|  | 8.1 | AVAILABILITY.............................................................................................................................................................. | 7 |
|  | 8.2 | SECURITY AND AUTHORIZATION ................................................................................................................................. | 7 |
|  | 8.3 | LOAD AND PERFORMANCE IMPLICATIONS ................................................................................................................... | 7 |
|  | 8.4 | MONITORING AND CONTROL ....................................................................................................................................... | 7 |

**1. Introduction**

1.1 Purpose

The purpose of this document is to define the software design of the MU Staff Calendar system. This document is intended for developers, testers, and project stakeholders. Developers will use it as a reference for implementing the system, while testers and stakeholders will understand how the system components work together.

1.2 Scope

This document applies to the MU Staff Calendar web application designed to facilitate scheduling, calendar syncing, and academic event planning among university staff. It defines the design for modules like user management, event syncing, and notifications.

1.3 Definitions, Acronyms, and Abbreviations

MU – Mahindra University

UI – User Interface

DB – Database

API – Application Programming Interface

CRUD – Create, Read, Update, Delete

1.4 References

Software Requirements Specification (SRS) Document

UML Diagrams – Project Folder

GitHub Repo – Source code

**2. Use Case View**

Use Case 1: Schedule Meeting

Description: Enables a staff member to schedule a new meeting with selected participants and check for time conflicts.

Actors: Staff Member

Steps:

Staff logs in.

Navigates to “Schedule Meeting”.

Enters meeting title, date, time, and participants.

System checks availability and confirms the schedule.

Meeting is added to calendars of all participants.

Use Case 2: Sync with Personal Calendar

Description: Allows staff to sync their MU calendar with personal calendars (e.g., Google Calendar).

Actors: Staff Member

Steps:

Staff logs in and goes to Settings.

Selects “Sync Calendar”.

Grants access to external calendar.

System syncs events.

**3. Design Overview**

3.1 Design Goals and Constraints

Should allow multi-user access with calendar conflict detection

Should integrate with Google Calendar API

Must be deployable on university servers

Tech Stack: React, Node.js, MongoDB

3.2 Design Assumptions

All users have institutional email accounts

Google Calendar API is available

The application will be accessed over the university intranet

3.3 Significant Design Packages

auth: Handles user authentication and roles

calendar: Event creation, editing, and syncing

notification: Sends email/calendar invites

admin: Admin interface for managing users and calendars

3.4 Dependent External Interfaces

External Application Module Using Interface Functionality Description

Google Calendar API calendar module Used to sync events and retrieve availability

3.5 Implemented Application External Interfaces

Interface Name Module Implementing Functionality Description

REST API /events calendar Allows CRUD operations for calendar events

REST API /auth/login auth Authenticates users

**4. Logical View**

4.1 Design Model

Key classes and modules:

User class – stores user info

Event class – stores meeting data

NotificationService – manages event notifications

CalendarService – handles Google Calendar integration

4.2 Use Case Realization: Schedule Meeting

Sequence Diagram:

User → UI → Auth → CalendarService → NotificationService

Details include validation, availability checks, and API calls.

**5. Data View**

5.1 Domain Model

Entities:

User (userId, name, email)

Event (eventId, title, startTime, endTime, participants[])

Notification (notificationId, message, sentTime)

5.2 Data Model

5.2.1 Data Dictionary

Attribute Description Type

userId Unique user identifier String

email User email String

eventId Unique event identifier String

startTime Start time of the event DateTime

**6. Exception Handling**

AuthException – for login failures

CalendarSyncException – for API errors

DatabaseException – for DB failures

Exceptions are logged to a log file and displayed with user-friendly messages

**7. Configurable Parameters**

Configuration Parameter Name Definition and Usage Dynamic?

MAX\_EVENT\_PARTICIPANTS Maximum number of attendees Yes

CALENDAR\_API\_KEY Key to access Google Calendar API No

**8. Quality of Service**

8.1 Availability

Scheduled daily backups

Graceful error handling

Admin panel for maintenance tasks

8.2 Security and Authorization

Role-based access (admin, faculty)

OAuth2-based login using institutional emails

Data access is restricted via JWT tokens

8.3 Load and Performance Implications

Expected max concurrent users: 200

Average events/day: 500

MongoDB indexes used for efficient queries

8.4 Monitoring and Control

Log files for error and user actions

Admin dashboard to monitor system health

Uptime monitoring tools configured

|  |
| --- |
| Abhinav (SE22UCSE271) |
| Bhanu (SE22UCSE060) |
| Ghanashyam (SE22UCSE127) |
| Sreehitha (SE22UCSE297) |
| Naga Vaishnavi (SE22UCSE094) |
| Hanum (SE22UCSE143) |
| Nihal (SE22UCSE181) |
| Rohan (SE22UCSE232) |