# Software Requirements Specification (SRS) of ISMT Hub

Table of Contents

[1. Introduction 1](#_Toc172636788)

[1.1 Purpose 1](#_Toc172636789)

[1.2 Scope 1](#_Toc172636790)

[1.3 Wish list Scope 1](#_Toc172636791)

[1.4 Definitions, Acronyms, and Abbreviations 1](#_Toc172636792)

[1.5 References 1](#_Toc172636793)

[1.6 Overview 2](#_Toc172636794)

[2. Overall Description 2](#_Toc172636795)

[2.1 Product Perspective 2](#_Toc172636796)

[2.2 Product Features 2](#_Toc172636797)

[2.3 Product Wishlist Feature 2](#_Toc172636798)

[2.3 User Classes and Characteristics 2](#_Toc172636799)

[2.4 Operating Environment 2](#_Toc172636800)

[2.5 Design and Implementation Constraints 3](#_Toc172636801)

[2.6 Assumptions and Dependencies 3](#_Toc172636802)

[3. Specific Requirements 3](#_Toc172636803)

[3.1 Functional Requirements 3](#_Toc172636804)

[3.1.1 WebApp Dashboard 3](#_Toc172636805)

[3.1.2 ISMT Notifications Center 3](#_Toc172636806)

[3.1.3 Event Calendar 3](#_Toc172636807)

[3.1.4 Custom Form Creation 3](#_Toc172636808)

[3.1.5 Role-Based Authentication 3](#_Toc172636809)

[3.2 Performance Requirements 4](#_Toc172636810)

[3.3 Design Constraints 4](#_Toc172636811)

[3.4 Software System Attributes 4](#_Toc172636812)

[3.5 Database Requirements 4](#_Toc172636813)

[3.6. Wish list requirement 4](#_Toc172636814)

[3.6.1 Job Vacancies 4](#_Toc172636815)

[3.6.2 Academic Calendar 4](#_Toc172636816)

[3.6.3 Assignment Calendar 4](#_Toc172636817)

# 1. Introduction

## 1.1 Purpose

The purpose of this SRS document is to define the requirements for the ISMT Hub. This system aims to enhance communication between the college administration and students, streamline administrative tasks, and provide a centralized platform for various student-related information and activities.

## 1.2 Scope

The ISMT Hub will include the following features:

* WebApp Dashboard
* Event Calendar
* Custom Form Creation with Sharable Links
* Role-Based Authentication (Admin and User)
* ISMT Notifications Center

## 1.3 Wish list Scope

* Job Vacancies (Placement automation)
* Academic Calendar
* Assignment Calendar

## 1.4 Definitions, Acronyms, and Abbreviations

* **SRS:** Software Requirements Specification
* **ISMT:** International School of Management and Technology
* **Admin:** Administrator with higher privileges
* **User:** General user (students and staff)

## 1.5 References

ISMT official website: https://ismt.edu.np/

## 1.6 Overview

This document provides a detailed description of the functional and non-functional requirements for the ISMT Hub. It serves as a guide for the development team to design, implement, and test the system.

# 2. Overall Description

## 2.1 Product Perspective

The ISMT Hub is a standalone web application designed to enhance communication and administrative efficiency within the college. It integrates seamlessly with existing systems like Google Classroom and supports over 1000 users, including staff and students.

## 2.2 Product Features

* **WebApp Dashboard:** Centralized interface for all functionalities.
* **Event Calendar:** Comprehensive calendar for college events and important dates.
* **Custom Form Creation:** Tool for creating and distributing custom forms.
* **Role-Based Authentication:** Secure login system with different roles for Admin and User.
* **ISMT Notifications Center:** Centralized notifications hub for all college announcements using email.

## 2.3 Product Wishlist Feature

* **Job Vacancies:** Section for job and internship opportunities.
* **Academic Calendar:** Comprehensive calendar for academic details and important dates.
* **Assignment Calendar:** Comprehensive calendar for Assignment details and important dates.

## 2.3 User Classes and Characteristics

**Administrators:** Manage content, create forms, and oversee job postings.

**Students and Staff (Users):** View notifications, events, job vacancies, and participate in surveys.

## 2.4 Operating Environment

**Client Side:** Web browsers (Chrome, Firefox, Safari, Edge)

**Server Side:** Node.js environment for running NestJS applications

## 2.5 Design and Implementation Constraints

**Technology Stack:** Must use NestJS for backend and Next.js for frontend.

**Authentication:** Use JWT for role-based authentication.

## 2.6 Assumptions and Dependencies

* Reliable internet connection for accessing the web application.
* Users have basic knowledge of web applications and internet usage.

# 3. Specific Requirements

## 3.1 Functional Requirements

### 3.1.1 WebApp Dashboard

**FR1:** The system shall provide a central dashboard for users after login.

**FR2:** The dashboard shall display a summary of notifications, events, and job vacancies.

### 3.1.2 ISMT Notifications Center

**FR3:** The system shall allow admins to post new notifications.

**FR4:** The system shall allow users to view and filter notifications by category and date.

### 3.1.3 Event Calendar

**FR5:** The system shall provide a calendar view of all events.

**FR6:** The system shall allow users to RSVP to events and receive reminders.

### 3.1.4 Custom Form Creation

**FR7:** The system shall provide a UI for admins to create custom forms.

**FR8:** The system shall generate sharable links for created forms.

**FR9:** The system shall collect and store form responses in a database.

### 3.1.5 Role-Based Authentication

**FR10:** The system shall provide a secure login mechanism using JWT.

**FR11:** The system shall restrict access to admin functionalities based on user roles.

## 3.2 Performance Requirements

**PR1:** The system shall support concurrent access by at least 500 users.

**PR2:** The system shall respond to user actions within 2 seconds under normal load.

## 3.3 Design Constraints

**DC1:** The system must be developed using NestJS and Next.js.

**DC2:** The system must adhere to responsive design principles for accessibility on different devices.

## 3.4 Software System Attributes

* **Scalability:** The system should be scalable to accommodate growing user numbers.
* **Reliability:** The system should have an uptime of 99.5%.
* **Usability:** The system should have an intuitive and user-friendly interface.
* **Security:** The system should implement role-based access control and data encryption.

## 3.5 Database Requirements

**DB1:** The system shall use MongoDB or PostgreSQL for data storage.

**DB2:** The database shall store user information, notifications, events and form responses securely.

## 3.6. Wish list requirement

### 3.6.1 Job Vacancies

**WR1:** The system shall allow admins to post job and internship opportunities.

**WR2:** The system shall allow users to view and apply for job vacancies.

### 3.6.2 Academic Calendar

**WR3:** The system shall provide a calendar view of all academic details.

**WR4:** The system shall allow users to RSVP to events and receive reminders.

### 3.6.3 Assignment Calendar

**WR5:** The system shall provide a calendar view of all assignment details.

**WR6:** The system shall allow users to RSVP to events and receive reminders.