

---

# Software Requirements Specification

for

## Student Activity Points Management

Prepared by:  
TEAM NO.6

<b>Ravipati Nikitha Chowdary</b>	<b>B220492CS</b>	<b>ravipati_b220492cs@nitc.ac.in</b>
<b>Shifana S Shafeek</b>	<b>B221204CS</b>	<b>shifana_b221204cs@nitc.ac.in</b>
<b>Sreedevi K</b>	<b>B220545CS</b>	<b>sreedevi_b220545cs@nitc.ac.in</b>

# Contents

- Revisions ii
- 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 1
  - 1.5 Document Conventions 1
  - 1.6 References and Acknowledgments 2
- 2 Overall Description 2
  - 2.1 Product Overview 2
  - 2.2 Product Functionality 2
  - 2.3 Design and Implementation Constraints 2
  - 2.4 Assumptions and Dependencies 2
- 3 Specific Requirements 3
  - 3.1 External Interface Requirements 3
  - 3.2 Functional Requirements 3
  - 3.3 Use Case Model 4
- 4 Other Non-functional Requirements 9
  - 4.1 Performance Requirements 9
  - 4.2 Safety and Security Requirements 9
  - 4.3 Software Quality Attributes 9
- Appendix A – Data Dictionary 10
- Appendix B - Group Log 10

# Introduction

## 1. Introduction

### 1.1 Document Purpose

This document provides an overview of the **Student Activity Points Management System**, detailing its scope, audience, and conventions. It serves as a reference for stakeholders involved in the development, implementation, and usage of the system.

### 1.2 Product Scope

The **Student Activity Points Management System (SAP)** will streamline and manage the process of tracking, verifying, and approving student activity points. The system enables students to submit activity requests, faculty members to review and approve them, and faculty advisors to oversee compliance with institutional requirements. SAP aims to enhance efficiency, transparency, and fairness in managing student activities while reducing manual workload.

### 1.3 Intended Audience and Document Overview

This SRS is intended for software developers, project managers, faculty members, faculty advisors, and system administrators involved in developing and deploying the **Student Activity Points Management System**.

### 1.4 Definitions, Acronyms and Abbreviations

- **SAP:** Student Activity Points
- **FA:** Faculty Advisor

### 1.5 Document Conventions

This document uses bold headers for section titles and employs standard numbering for easy reference.

## **1.6 References and Acknowledgments**

- Reference GitHub Repository:

[https://github.com/ShifanaSShfk/S6\\_SE\\_LAB](https://github.com/ShifanaSShfk/S6_SE_LAB)

# **Overall Description**

## **2.1 Product Overview**

This system is an independent web-based application integrated with institutional authentication mechanisms and role-based access control. It automates the activity point approval workflow and enhances transparency in student achievements.

## **2.2 Product Functionality**

The primary functionalities of SAP include:

- **Students:** Submit requests, track status, view points, receive notifications, access the leaderboard and calendar, and contact the admin.
- **Faculty:** Process requests, approve/reject submissions, upload certificates, manage events, view the calendar, and contact the admin.
- **Faculty Advisors:** Finalize approvals, track student progress, view the calendar, and contact the admin.

## **2.3 Design and Implementation Constraints**

- Compliance with institutional data policies.
- Predefined approval workflows.
- Support for high concurrent usage.

## **2.4 Assumptions and Dependencies**

- Integration with institutional login services.
- Timely faculty and FA reviews.

# Specific Requirements

## 3.1 External Interface Requirements

- **User Interface:** The system will provide role-based dashboards for students, faculty, and faculty advisors.
- **Hardware Interface:** The system will run on standard web browsers and require internet access.
- **Software Interface:** The system will integrate with authentication services for secure login and institutional databases for student and faculty records.
- **Communication Interface:** Email notifications will be sent for approvals, rejections, and deadline reminders.

## 3.2 Functional Requirements

### **3.2.1 Student Request Management**

- **FR1:** Submit new activity point requests with proof.
- **FR2:** Track the status of submitted requests.
- **FR3:** View request history, including approvals and rejections.

### **3.2.2 Approval Workflow**

- **FR1:** Faculty can review, approve, or reject student requests.
- **FR2:** Faculty advisors can oversee and manage approvals.
- **FR3:** Approvers can add comments or request additional information.

### **3.2.3 Role-Based Access Control**

- **FR1:** Provide separate dashboards for students, faculty, and faculty advisors.
- **FR2:** Restrict actions based on user roles (e.g., students cannot approve requests).

### **3.2.4 Category-Wise Point Tracking**

- **FR1:** Categorize activity points based on predefined criteria.
- **FR2:** Ensure students meet category-wise cut-off requirements.

### 3.2.5 Dashboard Management

- **FR1:** Students can view total points and pending requests.
- **FR2:** Faculty can see pending approvals and request details.
- **FR3:** Faculty advisors can monitor overall compliance and student progress.

### 3.2.6 Automated Notifications

- **FR1:** Alert students about approval status and missing requirements.

### 3.2.7 Reporting and Analytics

- **FR1:** Generate reports on student activity points and compliance.
- **FR2:** Provide insights into event participation trends.

## 3.3 Use Case Model

The **Student Activity Points Management System (SAP)** use case model outlines interactions between user roles (**Student, Faculty, Faculty Advisor, Administrator**) and system functionalities such as submitting requests, approving/rejecting requests, tracking activity points, and generating reports.

## Student Use Cases

### 3.1.1 View History - U1

- **Purpose:** Allow students to view their past activity submissions and approvals.
- **Actors:** Student
- **Preconditions:** The student must be logged into the system.
- **Postconditions:** The system displays the student's past activity records.

### 3.1.2 View Leaderboard - U2

- **Purpose:** Enable students to check their rank and compare activity points with peers.
- **Actors:** Student
- **Preconditions:** The student must be logged into the system.
- **Postconditions:** The system displays the leaderboard with rankings based on activity points.

### 3.1.3 Submit Request - U3

- **Purpose:** Allow students to submit activity requests along with proof.
- **Actors:** Student
- **Preconditions:** The student must be logged in and provide necessary details.
- **Postconditions:** The request is saved in the system and forwarded for approval.

### 3.1.4 Upload Proof - U4

- **Purpose:** Enable students to attach necessary proof while submitting activity requests.
- **Actors:** Student
- **Preconditions:** The student must be submitting or updating a request.
- **Postconditions:** The proof is uploaded and linked to the request.

### 3.1.5 Track Request Status - U5

- **Purpose:** Allow students to monitor the status of their submitted requests.
- **Actors:** Student
- **Preconditions:** The student must have submitted at least one request.
- **Postconditions:** The system displays the status (Pending, Approved, or Rejected).

### 3.1.6 View Calendar - U6

- **Purpose:** Enable students to view activity events and deadlines.
  - **Actors:** Student
  - **Preconditions:** The student must be logged into the system.
  - **Postconditions:** The system displays upcoming events and deadlines.
- 

## Faculty Use Cases

### 3.2.1 Overview - U7

- **Purpose:** Allow faculty to get a summary of student requests and approvals.
- **Actors:** Faculty
- **Preconditions:** The user must be logged in as faculty.
- **Postconditions:** The system presents an overview of pending and processed requests.

### 3.2.2 Approve/Reject Request - U8

- **Purpose:** Allow faculty to approve or reject student activity requests.
- **Actors:** Faculty
- **Preconditions:** There must be pending student requests for approval.
- **Postconditions:** The request is marked as approved or rejected, and feedback is provided.

### 3.2.3 Manage Categories - U9

- **Purpose:** Allow faculty to add, edit, or remove activity categories.
  - **Actors:** Faculty
  - **Preconditions:** The user must be logged in as faculty.
  - **Postconditions:** The changes to categories are saved and reflected in the system.
- 

## Faculty Advisor Use Cases

### 3.3.1 Generate Reports - U10

- **Purpose:** Enable faculty advisors to generate reports on student activity points.
- **Actors:** Faculty Advisor
- **Preconditions:** The system must have student activity data.
- **Postconditions:** A report is generated and available for download or viewing.

### 3.3.2 Overview - U11

- **Purpose:** Allow faculty advisors to get a summary of student requests and approvals.
- **Actors:** Faculty Advisor
- **Preconditions:** The user must be logged in as a faculty advisor.
- **Postconditions:** The system presents an overview of pending and processed requests.

### 3.3.3 Approve/Reject Request - U12

- **Purpose:** Allow faculty advisors to approve or reject student activity requests.
- **Actors:** Faculty Advisor
- **Preconditions:** There must be pending student requests for approval.
- **Postconditions:** The request is marked as approved or rejected, and feedback is provided.



### 3.3.4 Add/Waive Faculty - U13

- **Purpose:** Allow faculty advisors to add or waive faculty members responsible for approvals.
  - **Actors:** Faculty Advisor
  - **Preconditions:** The user must be logged in as a faculty advisor.
  - **Postconditions:** Faculty roles are updated in the system.
- 

## Admin Use Cases

### 3.4.1 Managing Records - U14

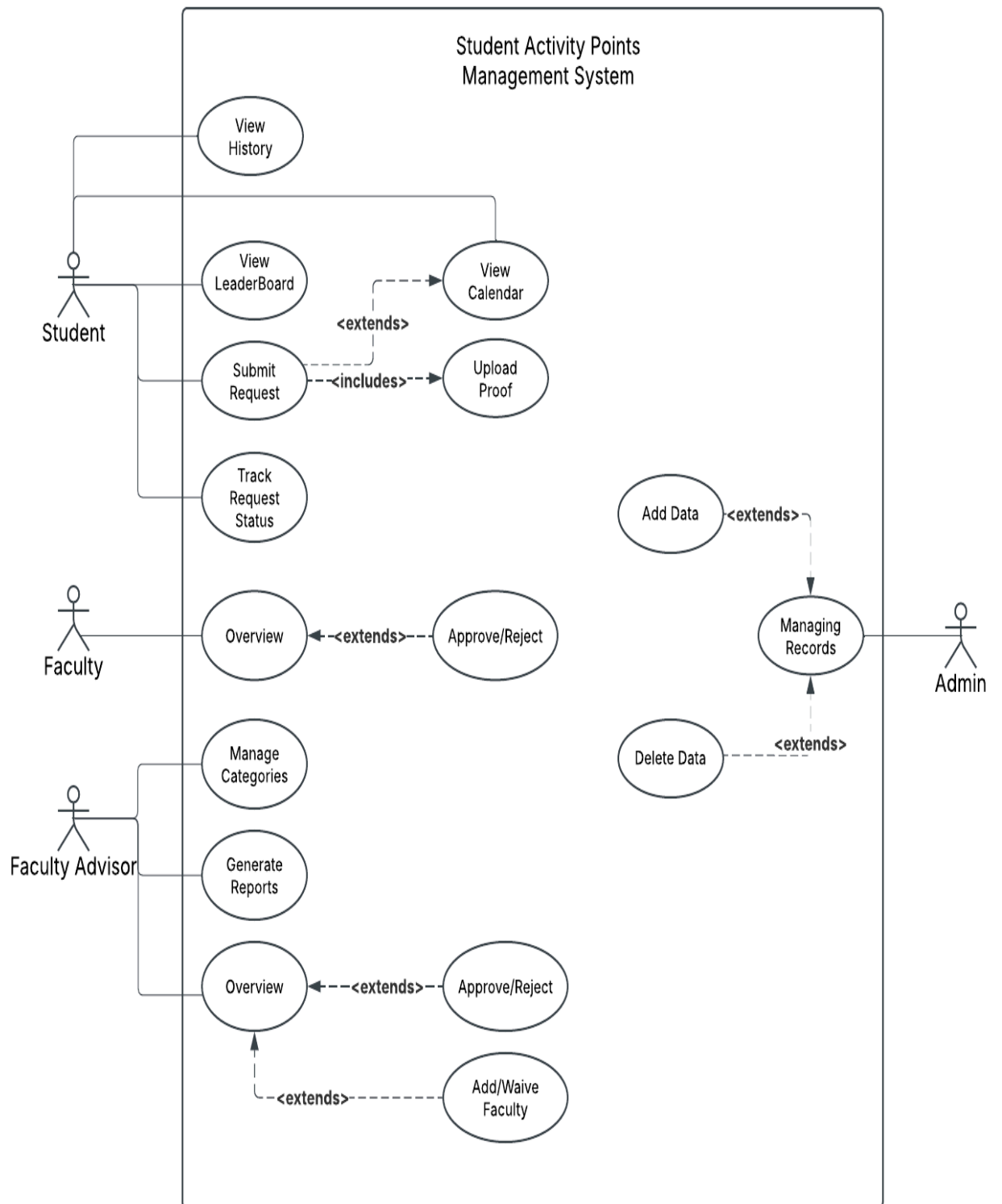
- **Purpose:** Allow the admin to manage system records, including faculty and student data.
- **Actors:** Admin
- **Preconditions:** The user must be logged in as an admin.
- **Postconditions:** The records are updated and reflected in the system.

### 3.4.2 Add Data - U15

- **Purpose:** Allow the admin to add new records related to users, events, or activity points.
- **Actors:** Admin
- **Preconditions:** The user must be logged in as an admin.
- **Postconditions:** The new data is added successfully.

### 3.4.3 Delete Data - U16

- **Purpose:** Enable the admin to remove outdated or incorrect data.
- **Actors:** Admin
- **Preconditions:** The user must be logged in as an admin.
- **Postconditions:** The specified data is removed from the system.



## Other Non-functional Requirements

### 4.1 Performance

The system should efficiently handle over 500 concurrent users without significant degradation in response time.

### 4.2 Safety and Security Requirement

- Data backups must be implemented to prevent data loss.
- User authentication and authorization mechanisms must be enforced.
- All data must be encrypted both in transit and at rest, ensuring compliance with institutional security standards.

### 4.3 Software Quality Attributes

- **Reliability:** Significant uptime.
- **Usability:** User-friendly, accessible interface.
- **Scalability:** Capable of handling increasing data load and additional users.

## Appendix A – Data Dictionary

Terms	Description
User	Any individual who interacts with the system. Users can be students, faculty members, faculty advisors, or administrators. Each user has specific roles and access permissions within the system.
Activity Points	A numerical value assigned to students for participating in extracurricular activities. These points contribute to meeting institutional requirements for student engagement.
Faculty Advisor (FA)	A faculty member responsible for overseeing and managing the approval of activity requests for a group of students.
Category	An activity which can be divided into Department level and Institute level.
Department Level Activity	Activities that are conducted within a specific department, such as department workshops, technical club events, and department-specific competitions.
Institute Level Activity	Activities that are organized at the college/university level, such as inter-departmental competitions, national-level hackathons, cultural fests, and sports tournaments.

## Appendix B - Group Log

Date	Outcome
12 January 2025	Prepared initial Draft of <b>PDD</b>
21 January 2025	Submitted the <b>PDD</b> by understanding the default point allocation system.
03 February 2025	Done the <b>UI design</b> of the project including detailed workflow, ensuring a structured and user-friendly interface.
20 February 2025	Completed the <b>Software Requirements Specification (SRS)</b> for the project, outlining the system's functionality, scope, intended users, external and functional requirements, and use case model.