

---

# 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

1. Introduction.....	2
1.1 Document Purpose .....	2
1.2 Product Scope .....	2
1.3 Intended Audience .....	2
1.4 Definitions, Acronyms and Abbreviations .....	2
1.5 Document Conventions .....	2
1.6 Project Repository .....	2
2. Overall Description .....	3
2.1 Overview .....	3
2.2 Product Functionality .....	3
2.3 System Constraints .....	3
2.4 Assumptions and Dependencies .....	3
3. Specific Requirements .....	4
3.1 External Interface Requirements .....	4
3.2 Functional Requirements .....	4
3.3 Use Case Model .....	12
3.4 Class Diagram.....	30
3.4 Database Diagram.....	31
4. System Requirements .....	32
4.1 Performance Requirements .....	32
4.2 Safety and Security Requirements .....	32
4.3 Software Quality Attributes .....	32
• Appendix A – Data Dictionary .....	33
• Appendix B - Group Log .....	33

# 1 INTRODUCTION

## 1.1 Document Purpose

This **Software Requirements Specification (SRS)** defines the objectives, scope, and requirements of the *Student Activity Points Management System*. It serves as a reference for stakeholders, ensuring clarity in development, implementation, and usage.

## 1.2 Product Scope

The *Student Activity Points Management System (SAP)* automates the tracking, verification, and approval of student activity points. It enables students to submit requests, faculty to review them, and faculty advisors to ensure compliance. SAP enhances efficiency, transparency, and fairness while minimizing manual effort.

## 1.3 Intended Audience

This SRS is intended for:

- **Software Developers** – Implement system functionalities.
- **Project Managers** – Oversee development and deployment.
- **Faculty Members & Advisors** – Review and approve student activity points.
- **System Administrators** – Maintain system operations.

## 1.4 Definitions, Acronyms and Abbreviations

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

## 1.5 Document Conventions

This document follows a structured format with:

- **Bold headers** for section titles.
- **Standard numbering** for easy reference.

## 1.6 Project Repository

[GitHub Repository](#)

## 2 OVERALL DESCRIPTION

### 2.1 Overview

A **web-based system** with **institutional authentication** and **role-based access control**, streamlining activity point approvals while ensuring transparency and efficiency.

### 2.2 Product Functionality

Primary Functionalities of SAP include

#### **Students:**

- Submit and track requests
- View points
- Receive notifications
- Access the leaderboard and calendar
- Contact the admin

#### **Faculty:** Review and approve/reject requests

- Upload certificates
- Manage events
- View the calendar
- Contact the admin

#### **Faculty Advisors:**

- Finalize approvals
- Track student progress
- View the calendar
- Contact the admin

### 2.3 System Constraints

- **Data Compliance:** Adheres to *institutional data policies* for security and privacy.
- **Workflow Standardization:** Implements *predefined approval workflows* for consistency.
- **Scalability:** Supports *high concurrent usage* for seamless accessibility.

### 2.4 Assumptions and Dependencies

- **Authentication:** Relies on institutional login services for secure access.
- **Review Process:** Requires timely approvals from faculty and faculty advisors.

## 3 SPECIFIC REQUIREMENTS

### 3.1 External Interface Requirements

- **User Interface:** Role-based dashboards for students, faculty, and faculty advisors.
- **Hardware Interface:** Accessible via standard web browsers with an active internet connection.
- **Software Interface:** Integration with authentication services for secure login and institutional databases for student and faculty records.
- **Communication Interface:** Automated email notifications for approvals, rejections, and deadline reminders.

### 3.2 Functional Requirements

#### 3.2.1 Student Request Management

##### FR1: Submit New Activity Point Request with Proof

###### Brief Description

A Student submits a request for activity points by filling out a form and attaching proof.

###### Initial Step-By-Step Description

Before this use case can be initiated, the Student must be **logged into the system**.

1. The Student selects "**Submit New Request**" from the dashboard.
2. The system presents the **Activity Request Form**.
3. The Student enters the required details:
  - **Event Name**
  - **Event Date**
  - **Category** (Institute Level, Department Level, etc.)
  - **Proof Document Upload** (PDF, Image, or Word file)
4. The Student clicks **Submit**.
5. The system **validates** the request, ensuring:
  - All required fields are filled.
  - The uploaded file is in an **acceptable format**.
6. If validation is **successful**, the system:
  - **Stores the request** in the database with **status = Pending Approval**.
  - **Notifies the assigned Faculty** about the new request.
  - Displays a **confirmation message** to the Student.

7. If validation **fails**, the system displays an **error message** and prompts for corrections.
- 

## FR2: Track the Status of Submitted Requests

### Brief Description

A Student can track the status of a submitted request.

### Initial Step-By-Step Description

Before this use case can be initiated, the Student must be **logged into the system**.

1. The Student navigates to **"My Requests"** from the dashboard.
  2. The system displays a **list of all submitted requests**, including:
    - Event Name
    - Submission Date
    - Current Status (**Pending, Approved, Rejected**)
  3. The Student selects a **specific request** to view details.
  4. The system displays:
    - **Request details** (Event Name, Date, Proof Document)
    - **Faculty/FA comments** (if applicable)
    - **Approval/Rejection Date**
  5. The Student **reviews** the current status and faculty feedback.
- 

## FR3: View Request History, Including Approvals and Rejections

### Brief Description

A Student can view a history of all past activity point requests, including approved and rejected submissions.

### Initial Step-By-Step Description

Before this use case can be initiated, the Student must be **logged into the system**.

1. The Student navigates to **"Request History"** from the dashboard.
2. The system displays a **chronological list** of past requests.
3. The Student can filter the list by:
  - **Status**
  - **Date Range**
  - **Event Category**
4. The Student selects a **specific request** to view details.
5. The system displays:
  - Event Name
  - Submission Date
  - Approval/Rejection Date

- Faculty Comments (if any)
- Proof Document (if available)
- 6. If the request was **rejected**, the Student can view the **reason**.

### 3.2.2 Approval Workflow

#### FR1: Faculty Can Review, Approve, or Reject Student Requests

##### Brief Description:

Faculty members review and process activity requests submitted by students.

##### Initial Step-By-Step Description:

1. The Faculty logs into the system and accesses the **"Pending Requests"** section.
  2. The system displays a list of **all unprocessed requests**.
  3. The Faculty selects a request and reviews the details.
  4. The Faculty can:
    - **Approve**: The system updates status to **"Approved"** and notifies the Student.
    - **Reject**: The Faculty enters a **rejection reason**, and the system updates status to **"Rejected"**.
    - **Request More Info**: The Faculty enters **comments**, and the system notifies the Student.
- 

#### FR2: Faculty Advisors Can Oversee and Manage Approvals

##### Brief Description:

Faculty Advisors have the final say in the approval of student requests.

##### Initial Step-By-Step Description:

1. The Faculty Advisor navigates to **"Final Approval"**.
  2. The system displays a list of **all Faculty-approved requests**.
  3. The Faculty Advisor selects a request and either:
    - **Approves the request** → Final approval granted.
    - **Rejects the request** → Sent back to Faculty for revision.
-

### FR3: Approvers Can Add Comments or Request Additional Information

#### Brief Description:

Faculty and Faculty Advisors can request additional details or proof before approving an activity request.

#### Initial Step-By-Step Description:

1. The Faculty or Faculty Advisor navigates to **"Pending Requests"**.
2. The system displays a list of **all student requests awaiting review**.
3. The Approver selects a specific request to review.
4. Instead of approving or rejecting, the Approver selects **"Request More Information"**.
5. The system presents a **text box** where the Approver enters comments specifying what additional details are needed.
6. The system updates the request status to **"Needs More Information"**.
7. The system **notifies the student** about the required corrections or additional proof.

### 3.2.3 Role-Based Access Control

#### FR1: Provide Separate Dashboards for Students, Faculty, and Faculty Advisors

#### Brief Description:

Each user role has a dedicated dashboard with relevant features.

#### Initial Step-By-Step Description:

1. The user logs in and is redirected based on their role:
  - **Student Dashboard:** Request Submission, Leaderboard, Calendar.
  - **Faculty Dashboard:** Pending Requests, Request Tracking.
  - **Faculty Advisor Dashboard:** Final Approval, Compliance Reports.

---

#### FR2: Restrict Actions Based on User Roles

#### Brief Description:

The system ensures users can only access authorized functions.

#### Initial Step-By-Step Description:



1. The user attempts an action.
2. The system checks their **role permissions**:
  - Students **cannot approve requests**.
  - Faculty **cannot submit requests**.
  - Faculty Advisors **cannot submit or edit requests**.

### 3.2.4 Category-Wise Point Tracking

#### FR1: Categorize Activity Points Based on Predefined Criteria

##### Brief Description:

The system categorizes activity points based on predefined event types and assigns points accordingly.

##### Initial Step-By-Step Description:

1. The Student submits an **Activity Request**.
  2. The system presents a **dropdown menu** where the Student selects the **event category** (e.g., Workshop, Hackathon, Seminar, Sports, Cultural).
  3. The system automatically assigns points based on predefined rules, such as:
    - **Workshop** → 10 points
    - **Hackathon** → 20 points
    - **Cultural Event** → 15 points
  4. If the request is **approved**, the assigned points are **added to the Student's total score**.
- 

#### FR2: Ensure Students Meet Category-Wise Cut-Off Requirements

##### Brief Description:

The system ensures that students meet minimum point requirements for each category before final approval.

##### Initial Step-By-Step Description:

1. The Student navigates to "**Activity Points Summary**".
2. The system displays a **breakdown of points earned** per category.
3. The system checks whether the Student has met the **minimum point requirement** in each category (e.g., at least 30 points in Technical Events).
4. If a category requirement is **not met**, the system:
  - **Highlights the missing categories** in red.

- Suggests upcoming **events** where the Student can earn points.
- 5. If the Student meets **all cut-off requirements**, they are **eligible for final approval**.

### 3.2.5 Dashboard Management

#### FR1: Students Can View Total Points and Pending Requests

##### Brief Description:

Students can access a **dashboard** displaying their **total activity points** and any **pending activity requests**.

##### Initial Step-By-Step Description:

1. The Student logs into the system and navigates to the **Dashboard**.
  2. The system displays:
    - **Total earned points** across all categories.
    - **Pending activity requests** with their current status (Pending, Approved, Rejected).
  3. The Student can **click on a request** to view details.
- 

#### FR2: Faculty Can See Pending Approvals and Request Details

##### Brief Description:

Faculty members have access to a **dashboard** displaying all **pending student requests** awaiting review.

##### Initial Step-By-Step Description:

1. The Faculty logs into the system and accesses the **Faculty Dashboard**.
  2. The system displays:
    - **Total pending requests** for review.
    - **A list of all pending requests** with details such as:
      - Student Name
      - Event Name
      - Submission Date
      - Proof Document
  3. The Faculty selects a request to review its details.
-

## FR3: Faculty Advisors Can Monitor Overall Compliance and Student Progress

### Brief Description:

Faculty Advisors can access a **dashboard** to monitor student compliance with activity point requirements.

### Initial Step-By-Step Description:

1. The Faculty Advisor logs into the system and navigates to the **FA Dashboard**.
  2. The system displays:
    - **Compliance Summary**: A chart showing students who meet or do not meet minimum point requirements.
    - **Student Progress Tracker**: A list of students with their earned points and required points for completion.
    - **Option to generate reports** on compliance.
- 

## FR4: View Leaderboard

### Brief Description

Students can view their ranking based on activity points.

### Initial Step-By-Step Description

1. The Student navigates to "**Leaderboard**" in the dashboard.
  2. The system retrieves **top-ranked students** based on activity points.
  3. The system presents a **ranked list** of students with filters for:
    - Overall leaderboard
    - Batch-wise ranking
    - Department-wise ranking
- 

## FR5: View Event Calendar

### Brief Description

Students and Faculty can view upcoming and past events.

### Initial Step-By-Step Description

1. The user navigates to "**Calendar**" from the dashboard.
2. The system displays a **monthly/weekly view of scheduled events**.
3. Users can **click on an event** to view details such as:
  - Event Name

- Date & Time
- Category (Workshop, Hackathon, etc.)
- Registration Requirements

### 3.2.6 Automated Notifications

#### FR1: Alert Students About Approval Status and Missing Requirements

##### Brief Description:

The system automatically **notifies students** about the approval status of their submitted activity requests and alerts them if they are missing required points or documents.

##### Initial Step-By-Step Description:

1. A Student submits an **Activity Request** for approval.
2. The system **sends a notification** when:
  - The request is **approved** → "Your request for [Event Name] has been approved."
  - The request is **rejected** → "Your request for [Event Name] has been rejected. Reason: [Faculty Comments]."
  - More information is needed → "Faculty has requested additional details for your request."
3. The Student receives the notification via:
  - **Dashboard alert** (visible upon login).
  - **Email notification** (if enabled).
4. If a Student is missing **required activity points**, the system sends an alert:
  - "You need [X] more points in [Category Name] to meet the minimum requirements."

### 3.2.7 Reporting and Analytics

#### FR1: Generate Reports on Student Activity Points and Compliance

##### Brief Description:

Faculty Advisors generate **compliance reports** to track student participation.

**Initial Step-By-Step Description:**

1. The Faculty Advisor accesses the **Reports** section.
2. The Faculty Advisor generates the report.

### **3.3 Use Case Model**

The **Student Activity Points Management System (SAP)** defines key interactions between user roles and system functionalities.

- **U1 - Login**

Actors	Student, Faculty/Faculty Advisor, Admin
Trigger	The actor selects the login option.
Precondition	The actor has valid login credentials.
Basic Path	1. The system prompts for a username and password. 2. The actor enters credentials and submits. 3. The system verifies details and grants access.
Alternative Paths	If incorrect credentials are entered, the system asks for a retry.
Postcondition	The actor is successfully logged in.
Exception Paths	The actor may exit without logging in.
Other	The system includes a "Forgot Password" option.

- **U2 - View History**

Actors	Student.
Trigger	The student selects the option 'Request History' on the dashboard to view their past

	activity records.
Precondition	The student has logged into the system.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays a list of the student's past requests.</li> <li>2. The student selects an entry to view details.</li> <li>3. The system retrieves and displays relevant information.</li> </ol>
Alternative Paths	If no history is available, the system displays a message indicating no records found.
Postcondition	The student has successfully viewed their activity history.
Exception Paths	The student may exit without selecting an entry.
Other	History includes event names, participation status, and earned points.

- **U3 - Track Request Status**

Actors	Student.
Trigger	The student selects the option 'Request Status' on the dashboard to check the status of a submitted request.
Precondition	The student has logged into the system and previously submitted a request.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays a list of submitted requests with status of each.</li> <li>2. The student selects an entry to view details.</li> <li>3. The system retrieves and displays relevant information.</li> </ol>
Alternative Paths	If no requests are found, the system notifies the student.
Postcondition	The student has successfully checked the request status.
Exception Paths	The student may exit without selecting a

	request.
Other	Status updates include pending, approved, or rejected.

- **U4 - View Calendar**

Actors	Student, Faculty/Faculty Advisor.
Trigger	The actor selects the option 'View Calendar' on the dashboard
Precondition	The actor has logged into the system
Basic Path	1. The system displays an interactive calendar with scheduled events. 2. The actor can select an event to view details.
Alternative Paths	If no events are available, the system displays an empty calendar.
Postcondition	The actor has successfully viewed event details..
Exception Paths	The actor may exit without selecting an event.
Other	Events include academic and extracurricular activities. Student can also send request from the Calendar.

- **U5 - Submit Request**

Actors	Student.
Trigger	The student selects the option to 'Send a New Request' on Dashboard or Selects the option 'Send Request' from Calendar.
Precondition	The student has logged into the system.
Basic Path	1. The system prompts the student to enter activity details. 2. The student provides necessary information and uploads proof.

	3. The system confirms submission.
Alternative Paths	If incomplete information is provided, the system asks for corrections.
Postcondition	The request is submitted for approval.
Exception Paths	The student may exit before submitting the request.
Other	Submitted requests are reviewed by faculty advisors.

- **U6 - Upload Proof**

Actors	Student, Faculty
Trigger	The actor selects the option to upload proof for an activity.
Precondition	The actor has logged into the system and is submitting an activity request(if student) or approving a request(if faculty).
Basic Path	<ol style="list-style-type: none"> <li>1. The system prompts the actor to attach proof.</li> <li>2. The actor uploads the required document or image.</li> <li>3. The system validates and confirms upload.</li> </ol>
Alternative Paths	If an invalid file type is uploaded, the system notifies the actor.
Postcondition	Proof is successfully attached to the request.
Exception Paths	The actor may exit without uploading proof.
Other	Proof can be certificates, screenshots, or attendance records.

- **U7 - View Leaderboard**

Actors	Student
--------	---------



Trigger	The student visits the dashboard
Precondition	The student has logged into the system.
Basic Path	1. The system retrieves and displays the leaderboard with top-performing students. 2. The student scrolls through rankings and views detailed scores.
Alternative Paths	If no data is available, the system notifies the student.
Postcondition	The student successfully views the leaderboard.
Exception Paths	The student may exit without viewing rankings.
Other	Leaderboards display activity points and rankings.

- **U8 - FAQs**

Actors	Student, Faculty/Faculty Advisor
Trigger	The actor clicks on the 'doubts?' option on the dashboard.
Precondition	The actor has logged into the system.
Basic Path	The system displays a list of frequently asked questions and answers.
Alternative Paths	If no relevant answer is found, the actor may drop question under 'Any Doubts?'.
Postcondition	The actor successfully accesses FAQ information.
Exception Paths	The actor may exit without selecting a question.
Other	FAQs help the actors understand system functionalities.

- **U9 - Overview**

Actors	Faculty, Faculty Advisor
Trigger	The actor selects the <b>"Overview"</b> option to review student requests and activity approvals.
Preconditions	The actor must be logged in.
Basic Path	<p>1.The system displays a <b>summary of pending, approved, and rejected requests</b>.</p> <p>2.The actor selects a <b>specific request</b> for detailed review.</p> <p>3.The system presents <b>request details</b>, including:</p> <ul style="list-style-type: none"> <li>● Student Name</li> <li>● Event Name</li> <li>● Submission Date</li> <li>● Proof Document (if available)</li> <li>● Faculty Comments (if any)</li> </ul> <p>4.The actor can <b>filter or sort requests</b> by date, student, or event type.</p> <p>5.The actor can choose to <b>approve, reject, or request more information</b>.</p>
Alternative Paths	<p>If there are <b>no pending requests</b>, the system displays <b>"No Requests Pending"</b>.</p> <p>If the actor applies <b>filters</b>, the system updates the displayed results accordingly.</p>
Postconditions	The system displays a summary of student submissions.
Exception Paths	<p>The actor may exit the overview without taking any action.</p> <p>If the system fails to retrieve data, an <b>error message is displayed</b>.</p>
Other	<p>The actor can <b>download reports</b> on student approvals and activity tracking.</p> <p>Approved requests are moved to the <b>Final Approval stage</b> if required.</p>

- **U10 - Approve/Reject Requests**

Actors	Faculty, Faculty Advisor
Trigger	The actor selects a <b>pending student request</b> from the dashboard for review.
Preconditions	The actor must have pending student requests to review.
Basic path	<p>1.The system displays a <b>list of pending student requests</b>.</p> <p>2.The actor selects a request to review.</p> <p>3.The system presents <b>request details</b>.</p> <p>4.The actor chooses one of the following actions:</p> <ul style="list-style-type: none"> <li>● <b>Approve:</b> <ul style="list-style-type: none"> <li>○ The system updates the request status to <b>"Approved"</b>.</li> <li>○ The system <b>notifies the Student</b>.</li> </ul> </li> <li>● <b>Reject:</b> <ul style="list-style-type: none"> <li>○ The actor enters a <b>reason for rejection</b>.</li> <li>○ The system updates the request status to <b>"Rejected"</b>.</li> <li>○ The system <b>notifies the Student</b>.</li> </ul> </li> </ul>
Alternative Paths	If there are <b>no pending requests</b> , the system displays <b>"No Requests Pending"</b> .
Postconditions	The system updates the request status.
Exception Paths	<p>The actor <b>exits without approving or rejecting</b> the request.</p> <p>If the system fails, the request <b>remains in pending status</b>, and an error message is displayed.</p>
Other	The faculty can add documents.

- **U11 - My Events**

Actors	Faculty
Trigger	The Faculty selects the <b>"My Events"</b> option to view and manage events they have created.

Preconditions	The faculty must be logged in with the required permissions.
Basic path	<p>1.The Faculty navigates to the <b>"My Events"</b> section.</p> <p>2.The system displays a <b>list of events organized by the Faculty</b>, showing:</p> <ul style="list-style-type: none"> <li>• Event Name</li> <li>• Event Date &amp; Time</li> <li>• Event Category</li> <li>• Number of Participants (if applicable)</li> <li>• Feedback Summary (if feedback is available)</li> </ul> <p>3.The Faculty selects an event to:</p> <ul style="list-style-type: none"> <li>• <b>View event details.</b></li> <li>• <b>Edit event information</b> (e.g., update date, description).</li> <li>• <b>Delete the event</b> (if no students are registered).</li> <li>• <b>View student feedback and ratings.</b></li> </ul>
Alternative Paths	<p>If no events have been created, the system displays <b>"No events found."</b></p> <p>If the Faculty <b>filters events by date or category</b>, the system updates the displayed list.</p>
Postconditions	Categories are updated in the system.
Exception Paths	<p>If the Faculty <b>exits before making changes</b>, no modifications are saved.</p> <p>If the <b>system fails to load event details</b>, an <b>error message is displayed</b>.</p>
Other	<p>The system allows the Faculty to <b>export event details and feedback</b> for reporting purposes.</p> <p>Events that <b>have already taken place</b> may have a <b>"Completed"</b> tag for reference.</p>

● **U12 - View Feedback**

Actors	Faculty
--------	---------

Trigger	The actor selects the option to <b>view feedback</b> on an activity request.
Preconditions	The faculty must have pending requests to review.
Basic Path	<p>1.The Faculty navigates to the <b>"My Events"</b> section.</p> <p>2.The system displays a <b>list of past events</b> organized by the Faculty.</p> <p>3.The Faculty selects an event to <b>view feedback</b> which includes:</p> <ul style="list-style-type: none"> <li>• Event Name</li> <li>• Submission Date</li> <li>• Approval/Rejection Status</li> <li>• Proof Document (if available)</li> </ul> <p>4.The Faculty reviews the feedback and can add comments based on his/her experience.</p>
Alternative Paths	If <b>no feedback has been submitted</b> , the system displays <b>"No feedback available for this event."</b>
Postconditions	The system records feedback for the student.
Exception Paths	<p>If the <b>Faculty exits without viewing feedback</b>, no action is recorded.</p> <p>If the <b>system fails to load feedback</b>, an <b>error message is displayed</b>.</p>
Extends	My Events

### U13 - Add Events

Actors	Faculty
Trigger	The Faculty selects the <b>"Add Event"</b> option from the <b>"My Events"</b> to create a new event.
Preconditions	The faculty must be logged in.
Basic path	<p>1.The Faculty navigates to <b>"My Events"</b> and clicks <b>"Add Event"</b>.</p> <p>2.The system displays the <b>Event Creation Form</b>, requiring:</p> <ul style="list-style-type: none"> <li>• <b>Event Name</b></li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Event Date &amp; Time</b></li> <li>• <b>Event Category (Workshop, Seminar, Hackathon, etc.)</b></li> <li>• <b>Event Description</b></li> <li>• <b>Maximum Participants (if applicable)</b></li> <li>• <b>Location (if offline) / Meeting Link (if online)</b></li> </ul> <p>3.The Faculty fills in the details and submits the form.</p> <p>4.The system <b>validates</b> the event details. If valid, the system:</p> <ul style="list-style-type: none"> <li>• <b>Saves the event</b> in the database.</li> <li>• Updates the <b>Event Calendar</b>.</li> <li>• Sends <b>notifications to students</b> about the new event.</li> </ul> <p>5.The system displays a <b>confirmation message</b> to the Faculty.</p>
Alternative paths	<p>If any required field is <b>left blank</b>, the system prompts the Faculty to complete it.</p> <p>If an event <b>already exists on the same date and time</b>, the system warns about scheduling conflicts.</p>
Postconditions	The event is created and visible in the system.
Exception paths	<p>If the <b>Faculty exits before submitting</b>, the event is <b>not created</b>.</p> <p>If the <b>server fails</b>, an <b>error message is displayed</b>, and the event is not saved.</p>
Includes	My Events
Other	<p>The Faculty can later <b>edit or delete the event</b> if necessary.</p> <p>The system may allow <b>event registrations</b> for students.</p>

- **U12 - Overview**

Actors	Faculty Advisor
Trigger	The FA visits the dashboard
Precondition	The Faculty Advisor has logged into the system.
Basic Path	<ol style="list-style-type: none"> <li>1. The system retrieves and displays an overview of student activity points, pending approvals, and faculty-related actions.</li> <li>2. The Faculty Advisor reviews the summary by selecting 'View Details'.</li> </ol>
Alternative Paths	If data is unavailable, the system displays a message indicating no records found.
Postcondition	The Faculty Advisor has an updated view of relevant system activities.
Exception Paths	The Faculty Advisor may exit without reviewing the overview.
Other	This helps in quick decision-making by providing a summarized view.

- **U14 - Approve/Reject Requests**

Actors	Faculty Advisor, Faculty
Trigger	The actor selects the option to approve or reject a student request.
Precondition	The actor has logged into the system.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays a list of pending student requests.</li> <li>2. The Faculty Advisor selects a request.</li> <li>3. The Faculty Advisor reviews the submitted proof and details.</li> <li>4. The Faculty Advisor selects either "Approve" or "Reject" and optionally adds comments.</li> <li>5. The system updates the request status.</li> </ol>
Alternative Paths	If no requests are pending, the system informs the Faculty Advisor.

Postcondition	The request status is updated in the system.
Exception Paths	The Faculty Advisor may exit without taking action.
Other	Requests may be reconsidered later based on student appeals.

- **U15 - Add/Waive Faculty**

Actors	Faculty Advisor
Trigger	The Faculty Advisor selects the option to add or waive a faculty member.
Precondition	The admin has logged into the system and accessed the records management section.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays the current list of faculty members.</li> <li>2. The Faculty Advisor selects the option to add a new faculty member or waive an existing one.</li> <li>3. If adding, the Faculty Advisor enters the faculty name and submits the request.</li> <li>4. If waiving, the Faculty Advisor selects a faculty member and confirms the action.</li> <li>5. The system updates the faculty records.</li> </ol>
Alternative Paths	If faculty details are incomplete, the system prompts the user to provide missing information.
Postcondition	The faculty list is updated in the system.
Exception Paths	The Faculty Advisor may exit without making any changes.
Other	NIL

- **U16 - Manage Categories**

Actors	Faculty Advisor
--------	-----------------



Trigger	The Faculty Advisor selects the option to manage activity categories.
Precondition	The FA has logged into the system.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays the existing list of activity categories.</li> <li>2. The Faculty Advisor selects an option to add, modify, remove a category or deciding the activity points for a student.</li> <li>3. If adding, the Faculty Advisor enters the category details and submits.</li> <li>4. If modifying, the Faculty Advisor updates the category information and saves the changes.</li> <li>5. If removing, the Faculty Advisor selects a category and confirms deletion.</li> <li>6. The system updates the category list.</li> </ol>
Alternative Paths	If required fields are missing, the system prompts the Faculty Advisor to complete them.
Postcondition	The activity category list is updated.
Exception Paths	The Faculty Advisor may exit without making changes.
Other	Categories help in organizing student activities effectively.

- **U17 - Student Details**

Actors	Faculty Advisor
Trigger	The Faculty Advisor selects the option to view student details from the Dashboard.
Precondition	The admin has logged into the system.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays a search option for students.</li> <li>2. The Faculty Advisor enters a student's name, roll number, or other identifiers.</li> <li>3. The system retrieves and displays the student's activity records, request history, and performance details.</li> </ol>

Alternative Paths	If no matching records are found, the system notifies the Faculty Advisor.
Postcondition	The Faculty Advisor has reviewed the student's details.
Exception Paths	The Faculty Advisor may exit without viewing details.
Other	This helps Faculty Advisors track and guide students under his/her FA ship effectively.

- **U18 - Managing Records**

Actors	Admin
Trigger	The admin selects the option to manage student activity records.
Precondition	The admin has logged into the system and accessed the records management section.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays existing student activity records.</li> <li>2. The admin selects a record to view, edit, or delete.</li> <li>3. If editing, the admin modifies the details and submits the changes.</li> <li>4. The system updates the database accordingly.</li> </ol>
Alternative Paths	If the admin selects to delete a record, the system asks for confirmation before proceeding.
Postcondition	The student activity records are updated or deleted as per the admin's actions.
Exception Paths	The admin may abandon the operation at any time.
Other	Records include student details, event participation, and points awarded.

- **U19 - Add Data**

Actors	Admin
Trigger	The admin selects the option to add new student activity data.
Precondition	The admin has logged into the system and accessed the records management section.
Basic Path	<ol style="list-style-type: none"> <li>1. The system presents a form for adding new student activity data.</li> <li>2. The admin fills in student details, event information, and points earned.</li> <li>3. The system validates the input data.</li> <li>4. If validation passes, the system saves the data in the database.</li> </ol>
Alternative Paths	If validation fails, the system prompts the admin to correct errors before submission.
Postcondition	The new student activity data is successfully added to the system.
Exception Paths	The admin may cancel the process before submitting.
Other	The data includes student ID, event details, approval status, and points awarded.

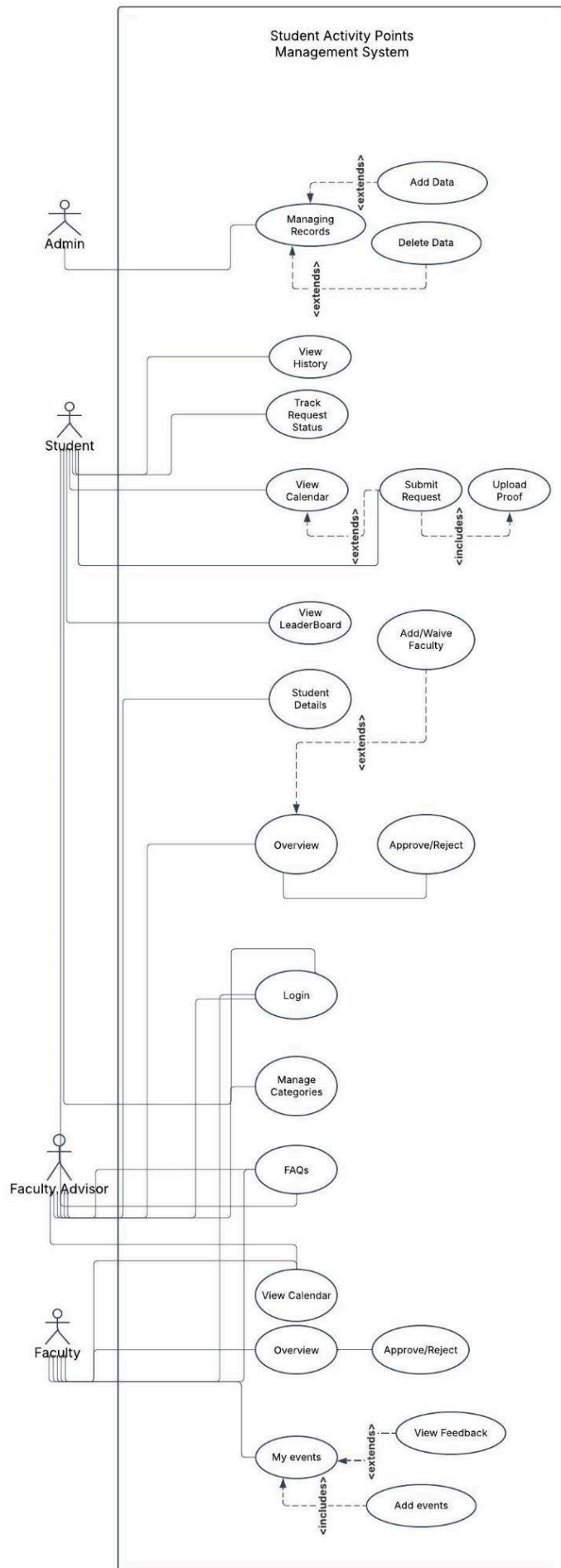
- **U20 - Delete Data**

Actors	Admin
Trigger	The admin selects the option to delete a student activity record.
Precondition	The admin has logged into the system and accessed the records management section.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays a list of student activity records.</li> <li>2. The admin selects a record to delete.</li> <li>3. The system asks for confirmation before proceeding.</li> <li>4. If confirmed, the system deletes the record from the database.</li> </ol>
Alternative Paths	If the admin chooses not to confirm deletion,

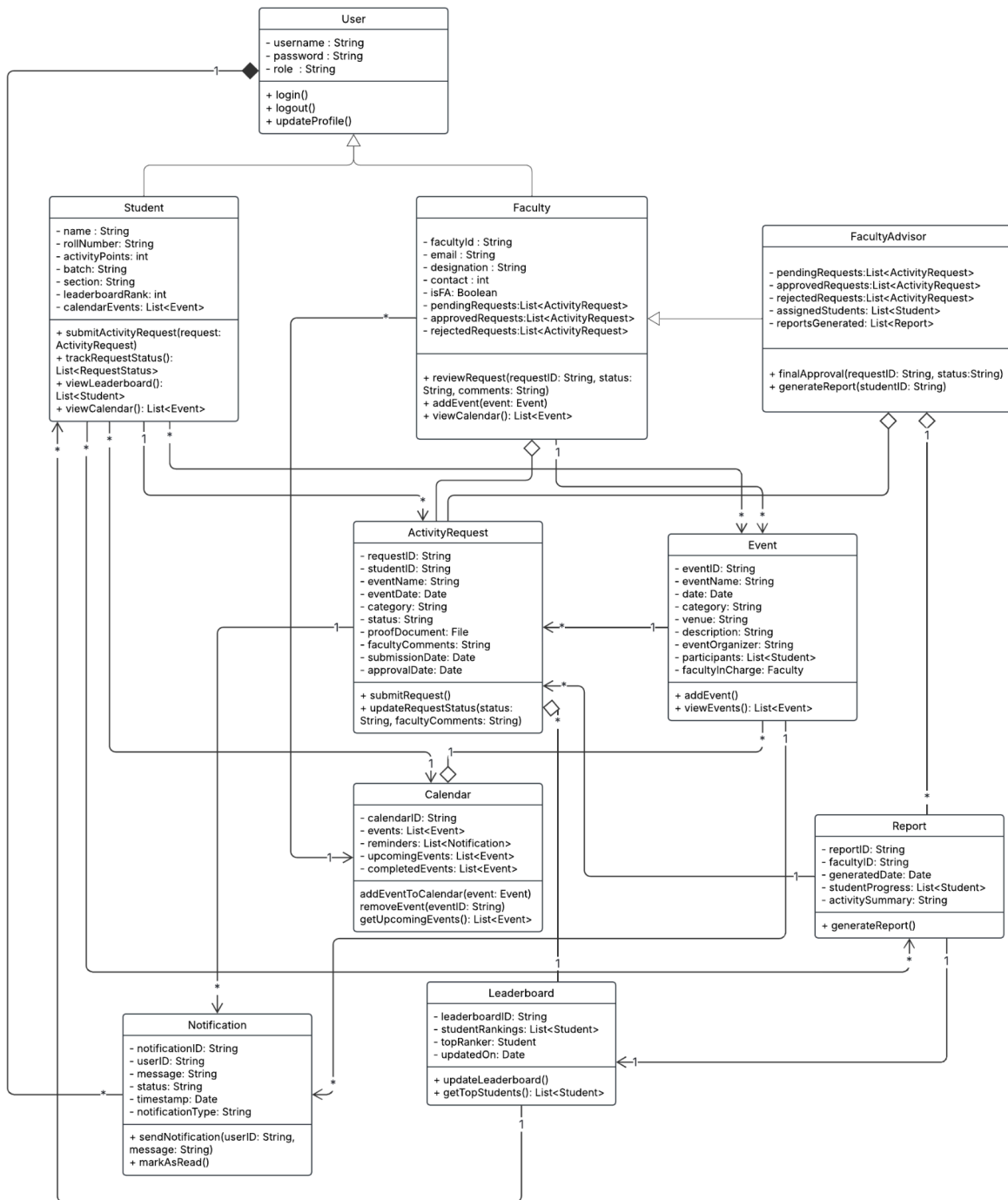
	the process is canceled.
Postcondition	The selected student activity record is removed from the system.
Exception Paths	The admin may abandon the operation before confirmation.
Other	Deletion is permanent and cannot be undone.

- **U21 - Answer FAQs**

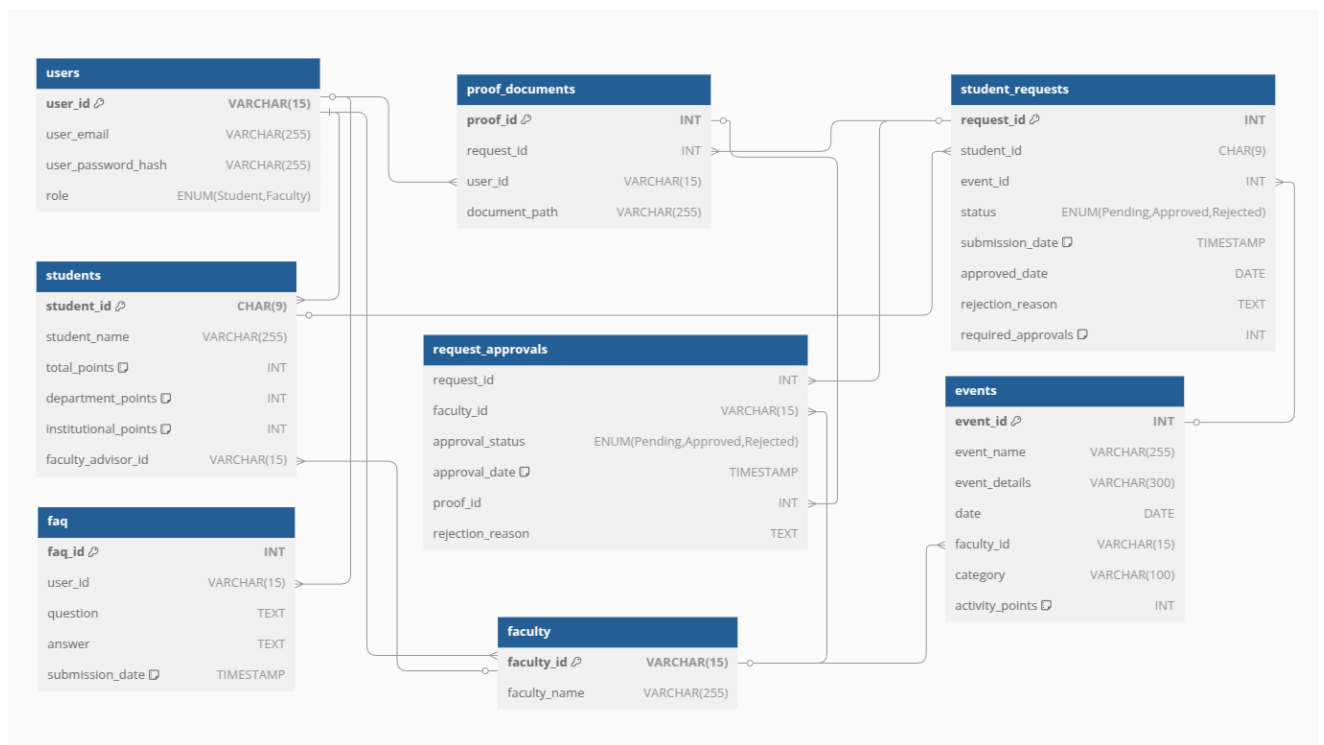
Actors	Admin
Trigger	The admin selects the option 'Answer FAQs'.
Precondition	The admin has logged into the system.
Basic Path	<ol style="list-style-type: none"> <li>1. The system displays a list of unanswered or existing FAQs.</li> <li>2. The admin selects an FAQ to answer or edits an existing answer.</li> <li>3. The admin enters or updates the answer and submits it.</li> <li>4. The system saves the response and updates the FAQ section.</li> </ol>
Alternative Paths	If an FAQ is no longer relevant, the admin can delete it.
Postcondition	The FAQ section is updated with the new or modified answer.
Exception Paths	The admin may exit without submitting an answer.
Other	FAQs help students and faculty understand system functionalities.



## 3.4 Class Diagram



## 3.5 Database Design



## 4 SYSTEM REQUIREMENTS

### **4.1 Performance**

The system must support over 500 concurrent users without noticeable latency.

### **4.2 Safety and Security**

- Regular data backups to prevent loss.
- Strong authentication and authorization controls.
- End-to-end encryption for data in transit and at rest, ensuring compliance with institutional security standards.

### **4.3 Software Quality Attributes**

- **Reliability:** High availability with minimal downtime.
- **Usability:** Intuitive and accessible user interface.
- **Scalability:** Seamlessly accommodates growing data and user demands.



## Appendix A – Data Dictionary

Terms	Description
User	Any individual interacting with the system, including students, faculty members, faculty advisors, and administrators, each with defined roles and access permissions.
Activity Points	Numerical values awarded to students for extracurricular participation, contributing to institutional engagement requirements.
Faculty Advisor (FA)	Faculty member overseeing and approving student activity requests.
Category	Classification of activities into <b>Department Level</b> and <b>Institute Level</b> .
Department Level Activity	Events specific to a department, such as workshops, technical club activities, and departmental competitions.
Institute Level Activity	College/university-wide events, including inter-departmental competitions, national 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> after analyzing 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> ,detailing system functionality, scope, intended users, external and functional requirements, and the use case model.
26 February 2025	We have refined the Software Requirements Specification (SRS) by incorporating detailed use cases, well-defined functional requirements. Additionally, we have included a structured class diagram and a database schema diagram to provide a clear representation of the system's architecture and data relationships.