

# STUDENT ACTIVITY POINTS MANAGEMENT

## Software Requirements Specification

### GROUP - 9

**Project lead:** Veda Vijayshankar

| Group Members              | Roll Number |
|----------------------------|-------------|
| Keerthana Lekshminarayanan | B220038CS   |
| Panguluri Tharun Chowdary  | B221097CS   |
| Veda Vijayshankar          | B220584CS   |

### Revision History

| Version | Date       | Description of Changes |
|---------|------------|------------------------|
| 1.0     | 19-02-2025 | SRS DOCUMENT           |

# CONTENT

| S.no | Topic                       | Pg.no |
|------|-----------------------------|-------|
| 1    | Functional Requirements     | 3     |
| 2    | Non functional Requirements | 5     |
| 3    | Use case diagram            | 7     |

# FUNCTIONAL REQUIREMENTS

## USER REQUIREMENTS

### 1. Student Requirements

#### 1.1. Signup & Login

- Sign up and log in to access the system.

#### 1.2. Activity Points Management

- Request activity points for completed activities.
- Optional ability to upload certificates when requesting points.
- Track their earned activity points over time.
- View their activity history (previously submitted activities and their status).

#### 1.3. Information & Engagement

- Recieve automated notifications about approvals, rejections, deadlines, or pending actions.
- View announcements posted by faculty advisors.
- See available activities they can participate in to earn points

### 2. Faculty Advisor Requirements

#### 2.1. Signup & Login

- Sign up and log in to access the system(role based).

#### 2.2. Student Activity Verification & Management

- Verify and approve/reject student activity submissions.
- View student progress, including points earned, pending activities, and history.
- Upload a roll list of students under their supervision.
- View activity history for tracking and auditing purposes.

#### 2.3. Communication & Notifications

- View and make announcements for students regarding activities, deadlines, or updates.
- Receive automated notifications about new activity submissions or pending approvals.

## **SYSTEM REQUIREMENTS**

- Store records and certificates securely.
- Send automated notifications to students and faculty regarding approvals, announcements, and deadlines.
- Manage student and faculty logins with secure authentication.
- Allow faculty to make announcements visible to students.
- Shows students available activities they can participate in.

## **OTHER FUNCTIONAL REQUIREMENTS**

### **1. Student Activity Submission (Input & Process Requirement)**

- Select or enter a custom activity.
- Enter activity name, type, date, and organizer.
- Upload proof (PDF) with validation.
- Submit for Faculty Advisor approval.
- Prevent duplicate submissions.

### **2. Faculty Advisor Activity Approval (Input & Process Requirement)**

- View pending activity submissions.
- Approve, reject, or request modifications with comments.
- Upload roll lists for validation.

### **3. Activity Points Tracking & Display (Output Requirement)**

- Show total activity points and category-wise breakdown.
- View activity history with status (Approved, Pending, Rejected).
- Display Faculty Advisor comments on rejected activities.
- Generate student progress and activity participation reports.

### **4. Notifications & Alerts (Output & System Process Requirement)**

- Pop-up notifications for approvals, rejections, and announcements.
- Automatic reminders for incomplete submissions.
- Alerts for Faculty Advisors on pending approvals.
- Automated overdue submission notifications.

### **5. Upcoming Activities & Events (Output Requirement)**

- Show list of upcoming activities (mandatory/optional).
- Faculty Advisors publish event-related announcements.

## **6. Announcement Management (Input & Process Requirement)**

- Faculty Advisors create/edit announcements.
- Include title, description, and attachments.
- Display announcements on the student dashboard.

## **7. System Data Management (Data Requirement)**

- Store student details (Name, Roll Number, Department).
- Store faculty details (Name, Email, Department).
- Maintain records of submitted activities and approval status.
- Store activity categories and point allocations.
- Log announcements (title, description, date).

# **NON FUNCTIONAL REQUIREMENTS**

## **1. Performance Requirements**

- The system should handle at least 50 concurrent users smoothly.
- The dashboard should load within 3-5 seconds under normal conditions.

## **2. Scalability Requirements**

- Initially designed for up to 200 users, but should allow future upgrades.
- API calls should be efficient, avoiding unnecessary database queries.

## **3. Security Requirements**

- Spring Security should be used for login authentication.
- JWT (JSON Web Token) should be used for session management.
- Passwords should be hashed (e.g., BCrypt) before storing in the database.

## **4. Usability Requirements**

- The UI should be responsive (work well on laptops, tablets, and mobiles).
- Basic help documentation should be available for users.

## **5. Maintainability Requirements**

- The backend code should be modular (separate controllers, services, and repositories).
- The code should follow proper comments and documentation.

## **6. Compliance Requirements**

- The system should follow basic data privacy rules (no public exposure of sensitive data).

## **7. Integration Requirements**

- The system should integrate with college databases later, but for now, it can use a dummy student dataset.

#### **8. Portability Requirements**

- The backend should run on Windows/Linux servers.
- The frontend should be compatible with Chrome, Firefox, and Edge.

## USE CASE DIAGRAM

