# Software Requirements Specification

## for

# Student Details

**Version 1.0**

**Group Name: FN-1**

1. **Sriya Vinaya Koppada**
2. **Abhinsha PV**
3. **Rajat Nambiar**

**Instructors: Joe Cheri Ross, Anu Mary Chacko, Manjusha**

**Course: Software Engineering Lab**

**Date: 20.02.2025**

# Contents

# 1. Use Case Diagram:

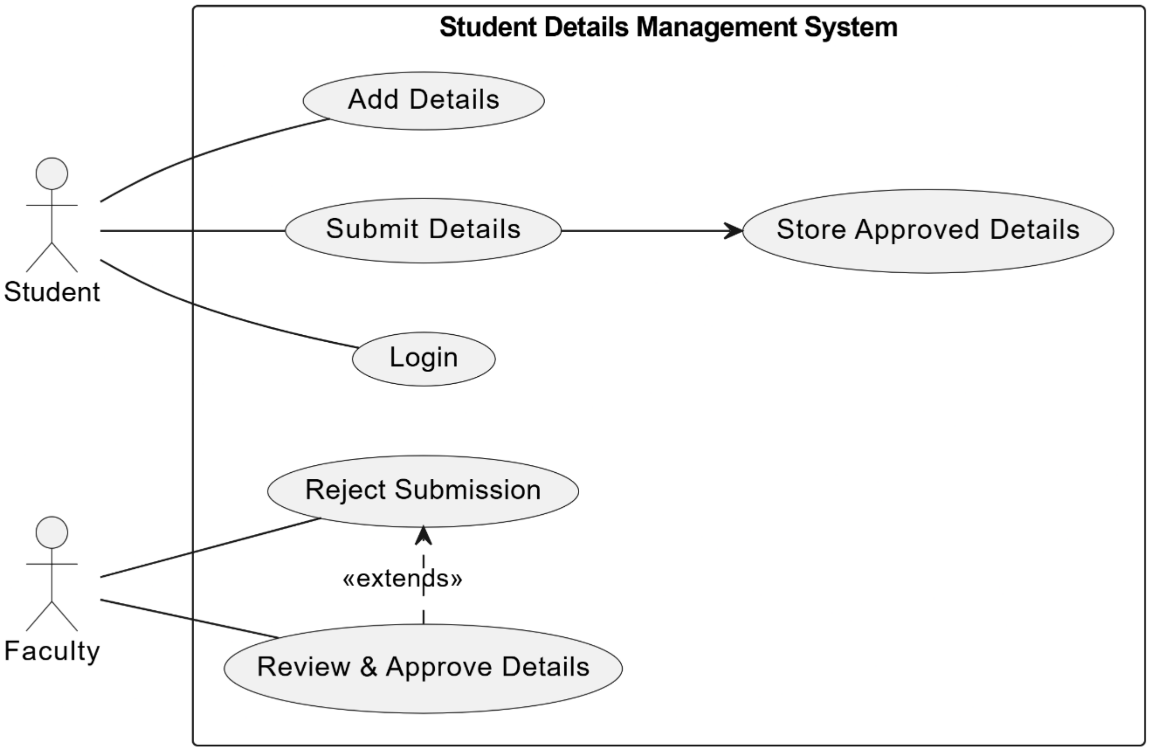
## Use Case: Adding Student Details

### Actors:

- **Student**: The primary actor who logs into the system and submits details about their participation in events, internships, and achievements.
- **Faculty**: The secondary actor responsible for reviewing and approving/rejecting the submitted details.

### Description:

This use case outlines the process where students add their details to the system. The system allows students to log in, fill in their information regarding participation in events, internships, placements, and achievements, and submit it for verification. Faculty members review the submissions and either approve or reject them. Once approved, the details are stored in the database and reflected in the student's profile.



### Flow:

1. The **Student** logs into the system.
2. The **Student adds details** such as event participation, internships, and certifications.
3. The **Student submits the details** for verification.

4. The **Faculty** reviews the submitted details.
5. If the details are **valid**, the **Faculty approves** them.
6. If the details are **incomplete or incorrect**, the **Faculty rejects** them.
7. Approved details are **stored in the system**.

# 2. Functional Requirements:

## 1. User Login

- Students, faculty, and administrators should be able to login to their accounts via institute Gmail on the platform.
- Different roles should be assigned based on user type (Student, Faculty, Admin).

## 2. User Profile Management

- Students cannot manually create or edit their profiles.
- Admin will upload an Excel sheet containing student data, which will be used to automatically create and update student profiles.
- Once imported, students should be able to view their profiles but not edit them.
- Faculty should have read access to student profiles for verification purposes.

## 3. Student Participation Records

- Students should be able to add and update their participation in technical events, cultural events, clubs, and sports events.
- The system should allow uploading of certificates and related documents.
- Faculty should be able to verify and approve participation records.

## 4. Placement & Internship Management

- Students should be able to add details of placements and internships.
- Uploading offer letters and other relevant documents should be supported.
- Faculty should have the ability to validate and approve placement records.

## 5. Student Publications Management

- The system should suggest titles from an existing faculty publications database when a student starts typing.
- If a matching paper is found, the student can select it, and details should be auto-filled.
- Faculty should be able to verify and approve publication details.

## 6. Data Import from External Databases

- Admin should be able to import student details from other existing databases.
- The system should allow proper data mapping and validation before importing.

## 7. Role-Based Access Control

- Students: Can manage their own profiles and participation records.
- Faculty: Can view, verify, and approve student records of students under them.

- Admin: Has full control over data management, user access, and system settings.

## 8. Search & Filter Functionality

- Users should be able to search for students, events, placements, internships, professional societies, and publications.
- Filters should be available based on academic year, department, event type, etc.

## 9. Notifications

- Students should receive notifications when:
    - Their event participation, placement, internship, or publication entry is approved or rejected.
- Faculty should receive notifications when:
    - A student uploads new details (e.g., event participation, placement, internship, publication).
    - A student requests verification or approval of their records.
- Admin should receive notifications when:
    - There is an error in profile data that needs correction.
    - A faculty member reports incorrect student details.
- Notifications will be delivered through in-platform alerts displayed on the dashboard.

## 10. Report Generation and Export

- Student, faculty and admin should be able to generate reports for accreditation purposes, student achievements, and placements.
- Faculty and Admin should be able to generate and download comprehensive reports of all students, including their event participation, placements, internships, publications, and professional society memberships.

## 11. Document Management

- Students should be able to upload certificates, offer letters and other documents of proof .
- Faculty and Admin should be able to view and verify uploaded documents.

## 12. Record Management

- Faculty can approve or reject records (event participation, placements, internships, publications) but cannot delete them directly.
- If a record is incorrect, faculty can reject it with comments, allowing the student to correct and resubmit.
- Only the Admin has the authority to delete records to maintain data integrity.
- Faculty can request deletion by flagging a record for admin review.

# 3. Non-Functional Requirements

## 1. Performance Requirements

- **Response Time:** The system should load student profiles and event records within 2 seconds under normal load conditions.
- **Throughput:** The system should be able to handle simultaneous access by at least 500 users without performance degradation.

## 2. Security Requirements

- **Authentication**:
  - Only authorized users (students, faculty, and admin) should be able to log in using secure authentication mechanisms.
- **Authorization:**
  - Role-based access control (RBAC) should be implemented to ensure that:
    - Students can view their own profiles and submit participation records.
    - Faculty can verify and approve/reject student records.
    - Admin has full control over user management and data imports.
  - Sensitive documents (offer letters, certificates) should be accessible only to authorized faculty and admin.

## 3. Usability Requirements

- **User Interface Design:**
  - The system should have a clean and intuitive UI to allow easy navigation for students, faculty, and admin.
  - Forms for submitting participation records should be simple, well-structured, and provide validation messages.
- **Accessibility:**
  - Keyboard navigation and screen reader compatibility should be supported.

## 4. Reliability & Availability

- The system should be available 99.5% of the time, ensuring minimal downtime.
- Automatic database backups should be taken daily to prevent data loss.

## 5. Scalability

- The system should be able to scale to support more students, faculty, and event records as the institution grows.
- The database should be optimized to handle an increasing number of records efficiently.

## 6. Data Storage & Integrity

- Data should be stored in a relational database with proper indexing to allow fast retrieval.
- Imported student profiles should be validated to avoid duplicate or incorrect entries.

## 7. Maintainability & Support

- The system should be modular and well-documented to allow easy future updates and maintenance.
- Admin should have an interface to update system settings, manage users, and import new student data.