Software Requirements Specification

for

Student Details

Version 3.0

Group Name: FN-1

Date	Version
20.2.25	1.0
26.2.25	2.0
03.3.25	3.0

Course: Software Engineering Lab

Date: 03.03.2025

Sriya Vinaya Koppada

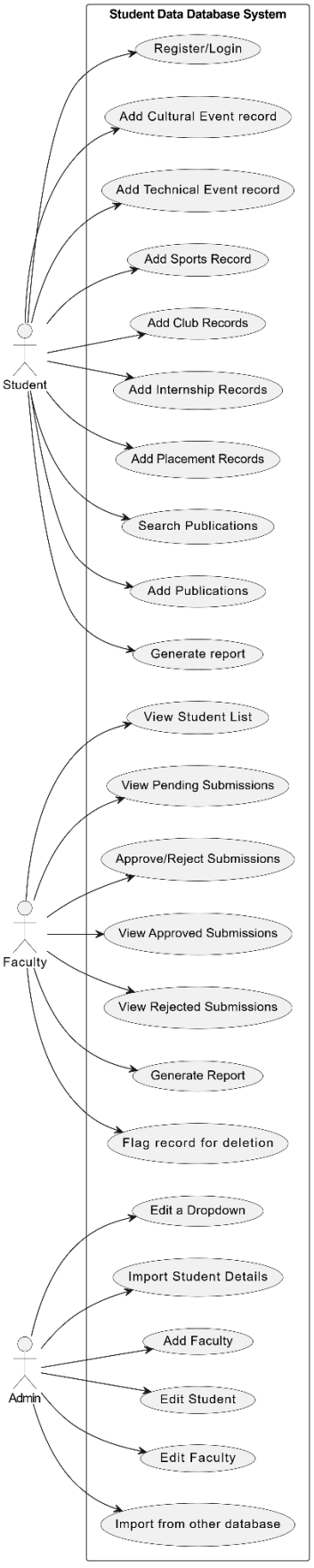
Abhinsha P V

Rajat Nambiar

Contents

1. Use Case Diagram: 3
2. Functional Requirements: 4
3. Non-Functional Requirements 7
4. Database Design 9
5. Class Diagram 10

1. Use Case Diagram:



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.
- **Creation of Student Profiles**:
  - Admin should have an option to upload an Excel file containing student details through a designated upload section.
  - Only files in the specified format (.xlsx or .csv) should be accepted.
  - The file must contain basic details of the students such as name, roll no., email, degree, etc.
- 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 details in technical events, cultural events, clubs, and sports events.
- Students should be able to upload certificates and related documents as proof of participation in events and clubs.
- Uploaded documents should be in standard formats such as PDF, JPG, or PNG, with a specified file size limit.
- Students can view their rejected, approved, pending records.
- Faculty should be able to verify and approve the submitted participation records.

4. Placement & Internship Management

- Students should be able to add details of placements and internships, such as company name, role, start date, etc.
- 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

• As a student types the title of a publication, suggestions should be provided from the existing faculty publications database.

• If a matching paper is found, the student can select it, and relevant details such as title, authors, and publication date should be automatically imported.

• If no matching paper is found, the student should have the option to manually enter publication details and submit them for approval.

• Upon submission, faculty should receive a notification to review the new publication entry.

6. Data Import from External Databases

- Admin should be able to import student details from other existing databases.
- Data import should support merging records from multiple sources, such as technical event participation, cultural activities, placements, internships, professional society memberships, and publications.
- Duplicate detection mechanisms should identify existing records based on unique identifiers such as roll number or student ID, ensuring that data is updated instead of creating duplicate entries.
- If conflicting data exists (e.g., different placement details for the same student), an option should be provided for the admin to manually resolve discrepancies.

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, clubs, 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 clubs.

11. Document Management

- Students should be able to upload certificates, offer letters and other documents of proof and research papers.
- 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.
- Faculty can request deletion by flagging a record for admin review.

13. Dynamic Dropdown Management

- Admin should have the ability to manage dropdown values dynamically If a record is incorrect, faculty can reject it with comments, allowing the student to correct and resubmit.
- An admin interface should allow:
  - Adding new dropdown values.
  - Editing existing values.
  - Removing outdated or incorrect values.

3. Non-Functional Requirements

1. Performance Requirements

- The system should be able to handle simultaneous access by a large number of 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.

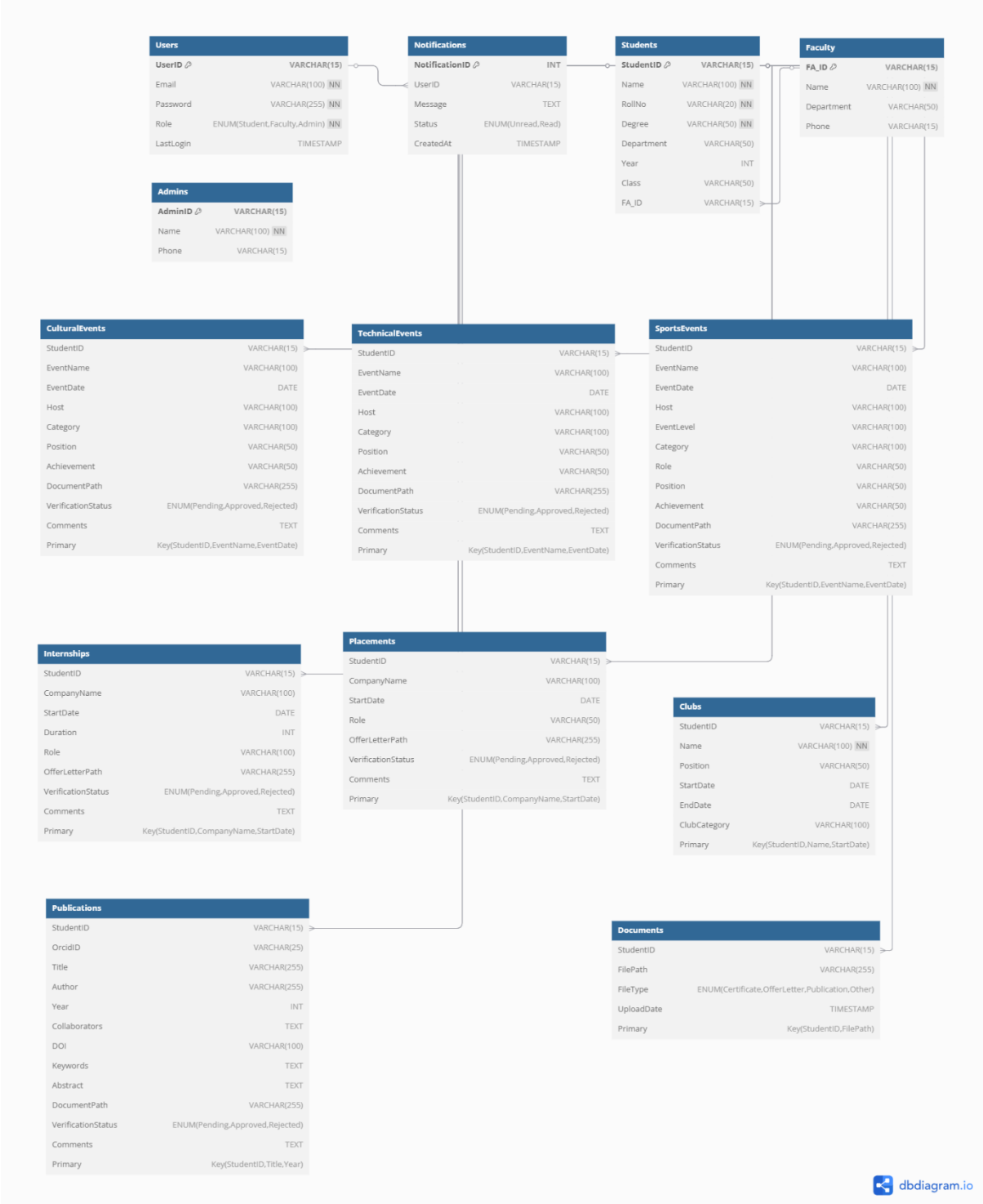
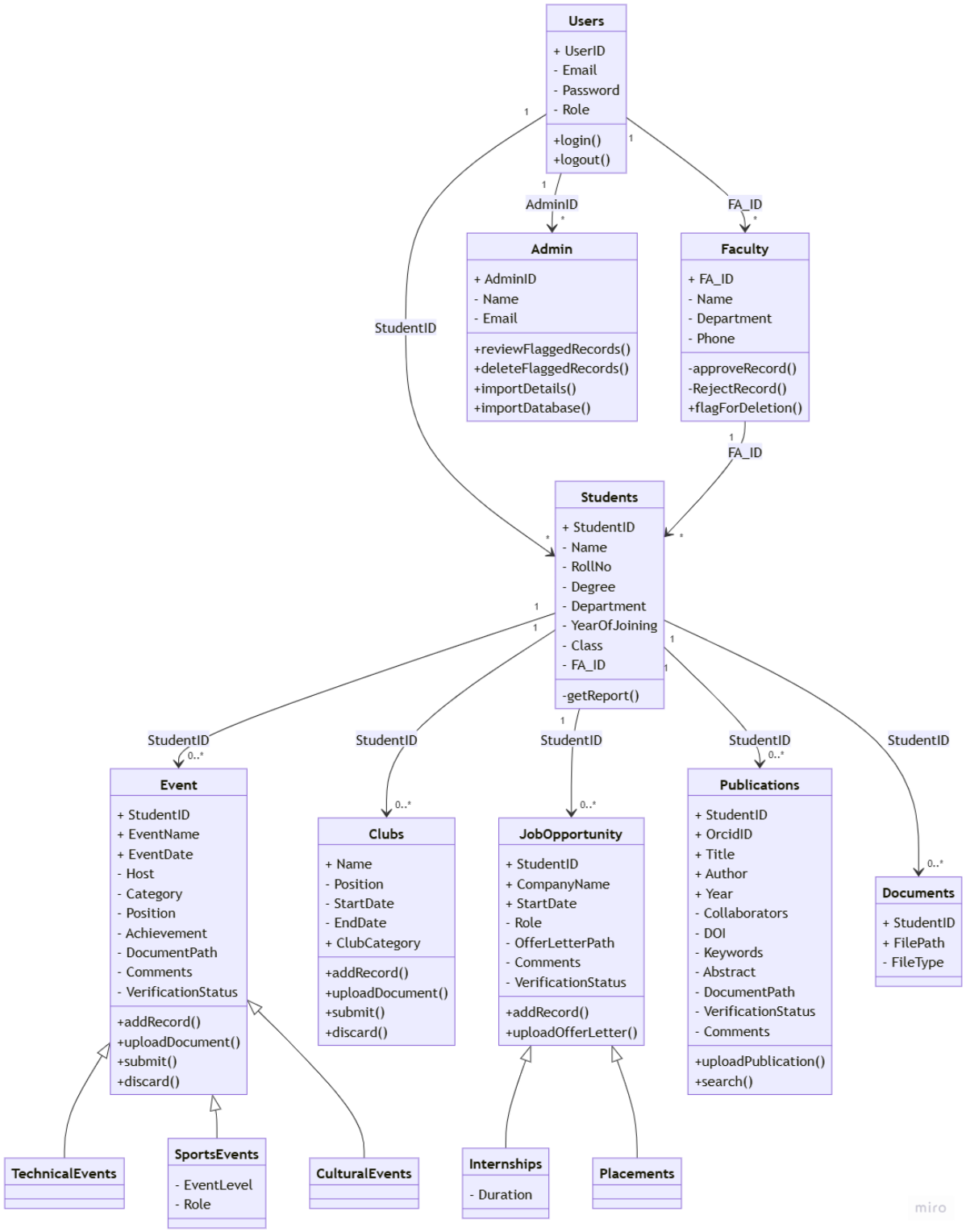
6. Data Storage & Integrity

- 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.

4. Database Design



5. Class Diagram