ITC 3081 – ICT Project

Department of Information and Communication Technology

Faculty of Technology

University of Sri Jayewardenepura

UniSource

University driven Open-Source Community

SRS Document

Group 01

Table of Contents:

1. Introduction

Unisource bridges the gap between academia and the software industry by allowing university students to contribute to real-world open-source projects. In effect, this solves the challenge facing students in securing internships necessary for practical experience that would enhance technical skills and employability. Any industrial organization, startup, freelancer, or open-source community can upload projects in UniSource and then students can contribute, create portfolios, and receive certification upon successful completion. This document constrains the functional and non-functional requirements of the Unisource and guides its further development and implementation.

1.1 Purpose

This SRS targets UniSource, a University-Driven Open Source Community. In respect to this, the document will orientate all the stakeholders which are developers, project managers, university faculty, students, and industry partners on what pertains to functional and non-functional requirements of the system. This will ensure that the platform will serve relevant users and achieve the goals set up by the project in bridging the gap between academia and the software industry.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirement Specification Documents. Each system requirement has been assigned a unique requirement number and priority level. Priorities for higher-level requirements informed the development of detailed requirements.

- Main heading: Calibri, 16 sizes, Bold
- Sub-heading: Calibri, 14 sizes, Bold
- Writing: Calibri, 12 sizes, Regular

1.3 Intended Audience and Reading Suggestions

The created Software Requirement Specification Document is aimed at the development team, Testers, and End Users

- **Developer Team:** This group uses the document to identify the technical requirements and convert them into the design and finally develop the system
- **Project Managers:** To plan and manage the development process.
- **Tester:** This team uses the document to look at the system features required to perform the unit test and system integration testing and how the system responds to it.
- **End users:** This group will be more interested in the overall description of the system. Which will help them understand the final product

1.4 Product Scope

The UniSource platform should be an all in one comprehensive solution in connecting university students with open source projects hosted and maintained by University faculty. This system will be created as an opportunity for students to actively contribute to open source projects, increasing their technical skills and employability, which would eventually enhance collaboration with software organizations. Key features will then include user registration, project submission, reviewing of proposals, management of portfolios, and etc.

1.5 References

[1] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

2. Overall Descriptions

This section will give information about product perspective, product functions, User Classes and Characteristics, Operating Environment, Design and Implementation Constraints, User Documentation, Assumptions, and Dependencies.

2.1 Product Perspective

This application can be divided into four primary components based on user roles which are students, organizations, mentors, and faculty.

Student Dashboard: This interface is used by university students. It offers facilities for registration, profile management, search for available open source projects, support for proposal submission, and maintenance of a portfolio of contributions that they made. The students could view the feedback and recommendations left by industry mentors.

Organization Dashboard: This is the interface that would be exposed to the representatives of organizations. Organizations can register themselves, submit open source projects, and assign mentors for those projects. All their professional profile or project-related updates and mentorship.

Mentor Dashboard: This is the interface that would be exposed to the mentors that are assigned by the organizations. Mentors can view proposals from students, and assign students for those projects, and further recommendation possibilities are also open to them for ranking a student accordingly.

Faculty Dashboard: This serves as the interface for university faculty. Faculty administrators can register, view all the project details, approve or reject projects, verify student and organization accounts, track progress and performance for both students and project progress. This component takes responsibility to implement academic standards throughout the project lifecycle.

2.2 Product Functions

2.2.1 Student Focused Functional Requirements

- The system should allow a student to register.
- The system should allow a student to manage their profile.
- The system should allow a student to view project descriptions.
- The system should allow a student to submit proposals for projects.

2.2.2 Admin Focused Functional Requirements

- The system should allow an admin to register and manage administrative accounts.
- The system should allow an admin to review project details.
- The system should allow an admin to approve or reject projects.
- The system should allow an admin to monitor project statuses.
- The system should allow an admin to verify student accounts.
- The system should allow an admin to verify organization accounts.
- The system should allow an admin to verify freelancer accounts.

2.2.3 Organization Focused Functional Requirements

- The system should allow an organization to register.
- The system should allow an organization to submit projects.
- The system should allow an organization to provide recommendations to students.
- The system should allow an organization to manage their profile.
- The system should allow an organization to manage mentors.

2.2.4 Mentor Focused Functional Requirements

- The system should allow a mentor to assign students to projects.
- The system should allow a mentor to review student proposals.
- The system should allow a mentor to monitor student performance.

2.1.5 General System Functions

- The system should allow users to login.
- The system should check user credentials.
- The system should check roles of access.
- The system should support real time information updates.
- The system should ensure data encryption and access control.
- The system should be designed as a web based application for ease of access.
- The system should provide clear and concise instructions for using all features.
- The system should ensure high availability and reliability.

2.2 User Classes and Characteristics

- Students: University students looking to gain practical experience.
- Organizations: Industry partners providing open-source projects.
- Administrators: Platform administrators managing overall operations.
- Mentors: Reviewing and monitoring projects.

2.3 Operating Environment

- Servers: Cloud-based servers (AWS or Azure).
- Client Devices: Modern web browsers on desktops, laptops, and mobile devices.
- Development Tools: Spring Boot, Next.js, Docker, GitHub.

2.4 Design and Implementation Constraints

- Must comply with university IT policies and data privacy regulations.
- Integration with existing university and industry systems.
- Scalable to support a growing number of users and projects.

2.5 User Documentation

User guides provide guidance for students, faculty, and organizations on platform usage. API documentation for developers, and admin guides for platform administrators.

2.6 Assumptions and Dependencies

Several assumptions and dependencies are implied in the success of UniSource. First, an uninterrupted internet connectivity to the end-user will keep the platform in service, smoothly contributing to projects without a hitch, communicating with industry partners, and making use of any other resources. Such users are at least assumed to have a basic knowledge of how to use a computer and the basic operations of Web applications. This level of proficiency is important to navigate easily in the platform and to make effective open-source contributions. Finally, an important dependency is the availability of industry partners who can provide open source projects. Such industry partners would be very important to offer real-world projects for students to work on.

3 External Interface Requirements

This section of the S.R.S. documents describe the external interface requirements for User Interfaces, Hardware Interfaces, Software Interfaces, and Communications interfaces are defined.

3.1 User Interfaces

The web interface of UniSource will be user-friendly, responsive to ensure accessibility across different devices. Inside the interface, there will be role-based, individualized dashboards for students, faculty, organizations, mentors, and administrators. All this will be role-based to make it easy to use for users because relevant information and tools are made available for these groups to navigate and interact with the site effortlessly.

3.2 Hardware Interfaces

No specific hardware requirements beyond standard web-enabled devices.

3.3 Software Interfaces

UniSource will have the capability to fully integrate with the student information system of the university, allowing efficient data transferring and management. Student enrollment, project assignment, and tracking are all automatically streamlined processes with UniSource. Proper integration with the existing systems of universities may also secure precise and consistent student data and, hence, better functionality and user convenience.

3.4 Communications Interfaces

The UniSource platform will allow only secure HTTP for any exchange of information between the user's browser and the server, so that data transferred between the user and the server are encrypted and hence protected. This security feature reinforces protection of sensitive information and enhances privacy and integrity of user interactions and data interchange on the platform.

4 System Features of UniSource

This section will give requirements for the system's features. Especially requirements for description and priority, Stimulus/Response Sequences, and functional requirements

4.1 User Registration and Authentication

4.1.1 Description and Priority

This feature will enable the safety of creation and management of users' accounts within the UniSource platform. The system authenticates the username and password that have been provided by the user and opens the appropriate interface depending on the designation linked to the user's account.
Priority Level: High

4.1.2 Stimulus/Response Sequences

1. The user browses the web application.
2. The application displays the home page.
3. Navigate to the signup page by clicking signup button.
4. The user submits the registration form with required details.
5. The system validates the submitted information.
6. Upon successful validation, the system creates the user's account.
7. The user receives a confirmation email with account details.

4.1.3 Functional Requirements

REQ-01: The system shall allow users to register with username and password.
REQ-02: The system shall validate the submitted registration details.
REQ-03: The system shall authenticate the login username and password.
REQ-04: The system shall display the appropriate user interface based on user role after authentication.

4.2 Portfolio Management

4.2.1 Description and Priority

This feature allows students to build and manage their portfolios, identifying skills applied and contributed to in projects.
Priority Level: High

4.2.2 Stimulus/Response Sequences

1. The student navigates to the dashboard page.
2. The student adds or updates personal details, academic background, and skills.
3. The student saves the updated portfolio.
4. The system confirms the changes and updates the portfolio.

4.2.3 Functional Requirements

REQ-05: The system shall allow students to create and manage their portfolios.
REQ-06: The system shall enable students to add and update personal details, academic background, and skills.
REQ-07: The system shall confirm and save changes made to the portfolio.

4.3 Organization Profile management

4.3.1 Description and Priority

This feature allows organizations to manage their organization details in the Unisource platform.
Priority Level: High

4.3.2 Stimulus/Response Sequences

1. The Organization navigates to the dashboard page.
2. The organization add and update their organization details.
3. The Organization saves the updated details.
4. The system confirms the changes and updates.

4.3.3 Functional Requirements

REQ-08: The system shall allow Organizations to create and manage their organization details.
REQ-09: The system shall confirm and save changes.

4.4 Mentor Creation

4.4.1 Description and Priority

This feature allows organizations to add a new mentor to the system.
Priority Level: High

4.4.2 Stimulus/Response Sequences

1. The Organization navigates to the create new mentor in the dashboard page.
2. The organization added mentor details.
3. The Organization saves the mentor details.
4. The system confirms the changes and updates.

4.4.3 Functional Requirements

REQ-10: The system shall allow Organizations to add new mentors to their organization.
REQ-11: The system shall confirm and save.

4.5 Project Creation

4.5.1 Description and Priority

This feature allows organizations to create new projects.
Priority Level: High

4.5.2 Stimulus/Response Sequences

1. The Organization navigates to the create new project in the dashboard page.
2. The organization added project details.
3. The organization assigned the mentor to the project.
4. The Organization saves the project details.
5. The system confirms the changes and updates.

4.5.3 Functional Requirements

REQ-12: The system shall allow Organizations to create new projects for their organization.
REQ-13: The system shall confirm and save.

4.6 Approve and Rejects the Projects

4.6.1 Description and Priority

This feature allows administrators to review and approve or reject the projects created by organizations.
Priority Level: High

4.6.2 Stimulus/Response Sequences

1. The admin navigates to the pending projects in the dashboard page.
2. The admin reviews the project details.
3. The admin changes the states of the project reject or approve.
4. The admin updates the project status.
5. The system confirms the changes and updates.

4.6.3 Functional Requirements

REQ-14: The system shall allow administrators can reviews and change the status of the project.
REQ-15: The system shall confirm and save.

4.7 Student verification

4.7.1 Description and Priority

This feature allows administrators to review and verify the newly created student accounts.
Priority Level: High

4.7.2 Stimulus/Response Sequences

1. The admin navigates to the pending students in the dashboard page.
2. The admin reviews the student details.
3. The admin changes the states of the student verification true or false.
4. The admin updates the student verification status.
5. The system confirms the changes and updates.

4.7.3 Functional Requirements

REQ-16: The system shall allow administrators can reviews and change the student verification status.
REQ-17: The system shall confirm and save.

4.8 Organization verification

4.8.1 Description and Priority

This feature allows administrators to review and verify the newly created organization accounts.
Priority Level: High

4.8.2 Stimulus/Response Sequences

1. The admin navigates to the pending organizations in the dashboard page.
2. The admin reviews the organization's details.
3. The admin changes the states of the organization verification true or false.
4. The admin updates the organization verification status.
5. The system confirms the changes and updates.

4.8.3 Functional Requirements

REQ-18: The system shall allow administrators can reviews and change the organization verification status.
REQ-19: The system shall confirm and save.

4.9 Proposal submissions

4.9.1 Description and Priority

This feature enables students to submit their project proposals for the relevant projects they are interested in contributing to.
Priority Level: High

4.9.2 Stimulus/Response Sequences

1. The student navigates to the project proposal submission in the dashboard page.
2. The student reviews the project details.
3. The student fills in the proposal details and submits the form.
4. The mentor receives a notification of the new proposal.

4.9.3 Functional Requirements

REQ-20: The system shall allow students to submit project proposals.
REQ-21: The system shall forward validated proposals to the relevant organization.
REQ-22: The system shall notify organizations of new student proposals.

4.10 Student Proposal Review

4.10.1 Description and Priority

This feature allows mentors to review proposals submitted by students for organizational projects and assess their suitability based on project needs and criteria.
Priority Level: Medium

4.10.2 Stimulus/Response Sequences

1. The mentor receives a notification of a new student proposal.
2. The mentor reviews the proposal details.
3. The mentor assesses the suitability of the proposal based on predefined criteria
4. The mentor assigns the student for that project.

4.10.3 Functional Requirements

REQ-23: The system shall notify mentors of new student proposals.
REQ-24: The system shall allow mentors to review proposal details.
REQ-25: The system shall update the proposal status and notify students of the mentor's feedback or decision.

4.11 Recommendations Provision

4.11.1 Description and Priority

This feature allows mentors to provide recommendations and feedback to students based on their contributions and performance in projects.
Priority Level: Medium

4.11.2 Stimulus/Response Sequences

1. The mentor navigates to the student's project contribution page.
2. The mentor reviews the student's contributions and performance.
3. The mentor provides recommendations and feedback.
4. The system records the feedback and notifies the student.

4.11.3 Functional Requirements

REQ-26: The system shall allow mentors to access student project contribution pages.
REQ-27: The system shall enable mentors to review student contributions and performance.
REQ-28: The system shall allow mentors to provide recommendations and feedback.
REQ-29: The system shall record the feedback and notify the student.

4.12 Project Monitoring

4.12.1 Description and Priority

This feature allows administrators to continuously monitor all project statuses to ensure that project progress satisfies its timelines, objectives, and student contributions.
Priority Level: Low

4.12.2 Stimulus/Response Sequences

1. The administrator navigates to the project monitoring dashboard.
2. The administrator views the status of all ongoing projects.
3. The administrator selects a project to view detailed progress and contributions.
4. The administrator identifies any issues or delays and takes necessary actions.

4.12.3 Functional Requirements

REQ-30: The system shall provide a project monitoring dashboard accessible to administrators.
REQ-31: The system shall display the status of all ongoing projects on the dashboard.
REQ-32: The system shall allow administrators to view detailed progress and contributions of individual projects.

4.13 Reset Password

4.13.1 Description and Priority

This feature allows users to reset their passwords if they forget them, ensuring secure and easy access to their accounts.
Priority Level: Low

4.13.2 Stimulus/Response Sequences

1. The user navigates to the login page and clicks on the "Forgot Password" link.
2. The system prompts the user to enter their registered email address, old password and new password.
3. The user enters their email address, old password and new password and submits the form.
4. The system updates the password and notifies the user of the successful reset.

4.13.3 Functional Requirements

REQ-33: The system shall provide a "Forgot Password" link on the login page.
REQ-34: The system shall allow users to enter and confirm a new password.
REQ-35: The system shall update the user's password upon successful submission of the new password.
REQ-36: The system shall notify the user of the successful password reset.

5 Other Non-functional Requirements

This section will give non-functional requirements for the system's features. Especially non-functional requirements for Performance Requirements, Safety Requirements, Security Requirements, and Software Quality Attributes. In addition, business rules are defined.

5.1 Performance Requirements

The System should ensure that Real-time Information is updated promptly without noticeable Delays. Minimize the loading times, ensuring that Pages and Features load within a short period under the normal load Conditions. The System should support a larger number of concurrent Users without performance degradation.

5.2 Safety Requirements

If there is any error in the computer, it can damage the database of the system. To prevent it, measures like database backup can be followed.

5.3 Security Requirements

All sensitive Data, including Personal Information and Project Details are encrypted both in transit and at Rest. The System should implement Access Control strategies to restrict access to the different Functionalities based on User Roles such as Students, Faculty and Administrators. The System must only accept Student Registrations from a verified University Email addresses to ensure that only the legitimate Students can access the System. The System should ensure

Data isolation with robust Access Controls, allowing Organizations to access only their own Project plans and preventing unauthorized viewing or modification by Others Data partitioning.

5.4 Software Quality Attributes

Usability Requirements. Design intuitive and User-friendly Interfaces and adhering to standard design principles for ease of navigation. The System must be designed as a Web-based Application to ensure ease of access and User engagement, minimizing resource waste by eliminating the need for additional Software installations. Provide clear and concise instructions for using all platform features. The System should be accessible to users with disabilities

Maintainability. This system should be developed so that it can be easily maintained. A correct Standard should be followed while coding the system, and it should be prepared so that other developers can easily annotate it.

Reliability. The system ensured high availability for Users. Scheduled maintenance activities to off-peak hours.

5.5 Business Rules

The system should contain at least the following roles that would relate to accesses to and dealings with the system which are Admin, Student, Organization, and Mentor. Following are the business rules that shall apply to these roles:

Admin Role

- Can view all configuration settings available in the system.
- View and approve or reject project submissions
- Update records regarding project status and contribution of a student
- Verify authenticity and eligibility of student, organization, and mentor accounts.

Student Role

- Registers and manages their personal and academic details within their profile.
- View the description of the project and requirements.
- Submit project proposals on accepted projects.
- Update and maintain contributions in the project portfolio.

Organization Role

- Registers and manages organization accounts detailed information.
- Submits project for review and approval by the faculty.
- Manages mentor assignment and communication with students.

Mentor Role

- Reviews student proposals for organizational projects.
- Assigns students to projects based on skills, interests, and project requirements.
- Tracks and monitors student performance throughout the duration of the project.

- Facilitates communication and collaboration between the students and the organization.
- Provides feedback and recommendations to students based on their performance.

General Business Rules

- All sensitive data, which includes personal information and details of projects, must be encrypted both in-transit and at-rest.
- Independent access control to different functionalities of the system based on user roles which are student, faculty, organization, mentor.
- More elaborate, organizations would see only their corresponding project details and proposals to the same by students, letting none of its content be accessed or mangled without authority.

6 Other Requirements

It should comply with different regulations on data protection, such as the GDPR, hence ensuring the secure handling of all personal and sensitive information regarding the storage of data in a manner that follows the set legal standards.

Appendix A: Glossary

- Admin- This is a user role responsible for overall administration and supervision of the Unisource platform.
- API (Application Programming Interface) - A set of rules and protocols used in building and interacting with software applications.
- AWS (Amazon Web Services) - A cloud computing platform provided by Amazon.
- Encryption - The process of converting information or data into a code.
- Faculty Dashboard - An interface used by university faculty administrators.
- GitHub - A platform for version control and collaboration.
- HTTP (Hypertext Transfer Protocol) - The protocol used for transferring web pages on the internet.
- Mentor - A representative assigned by organizations to guide and monitor student contributions to projects.
- Portfolio - A collection of a student's work and contributions to projects, showcasing their skills and achievements.

Appendix B: Analysis Models

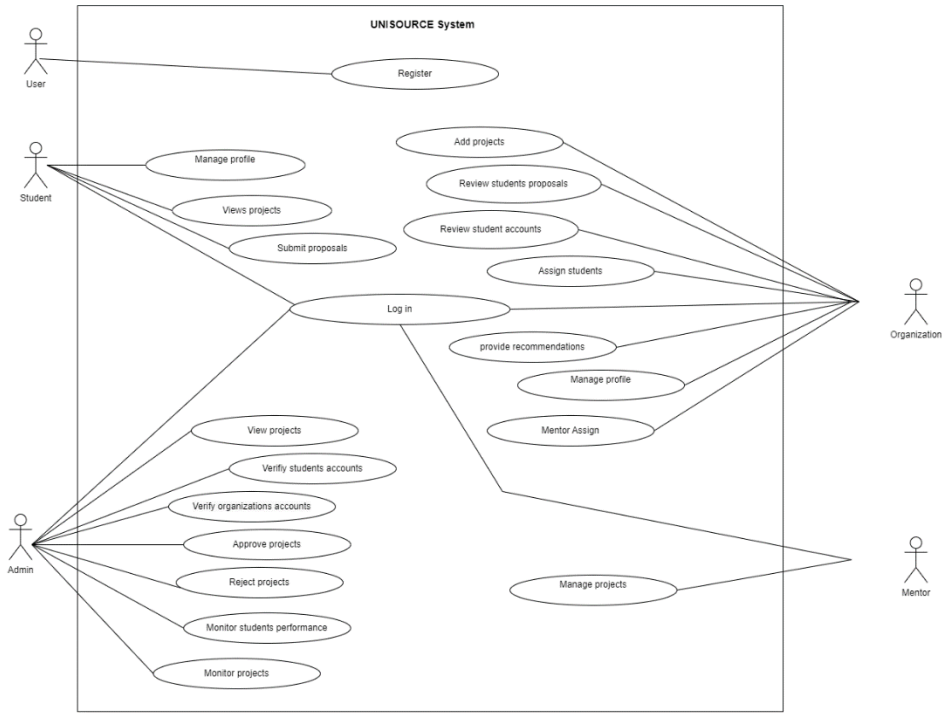

Figure 1: UniSource Use Case Diagram

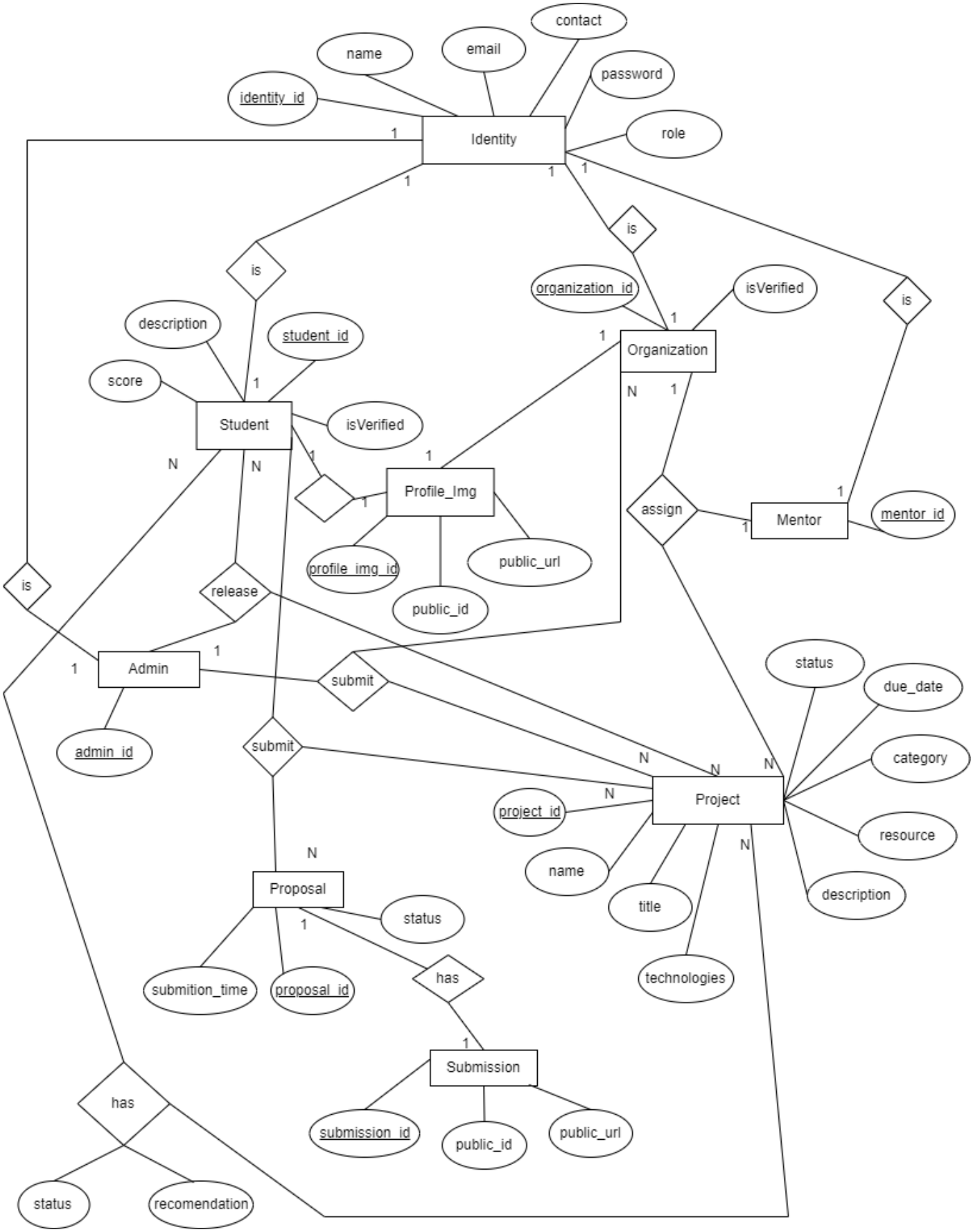

Figure 2: UniSource ER- Diagram

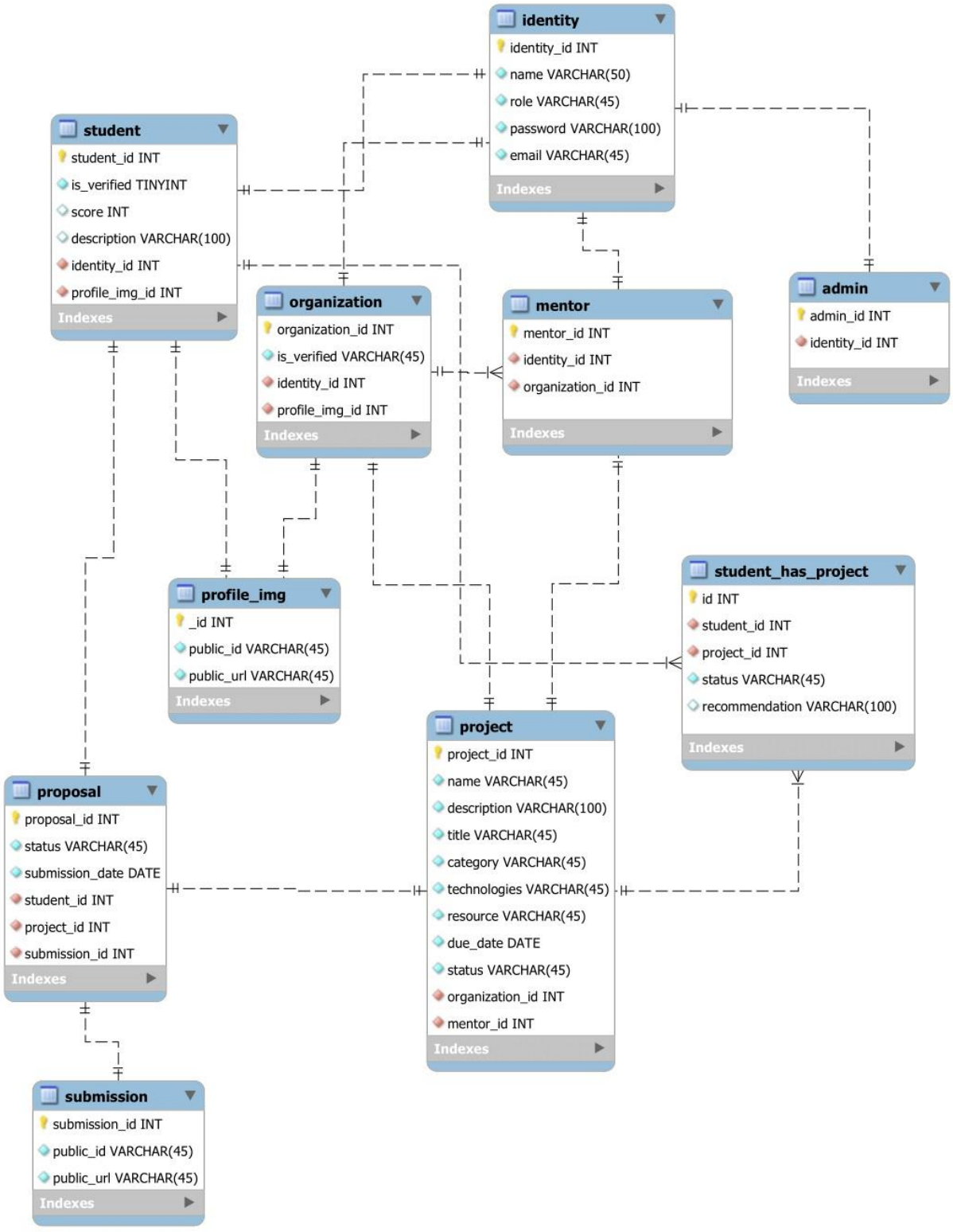

Figure 3: UniSource Database Diagram

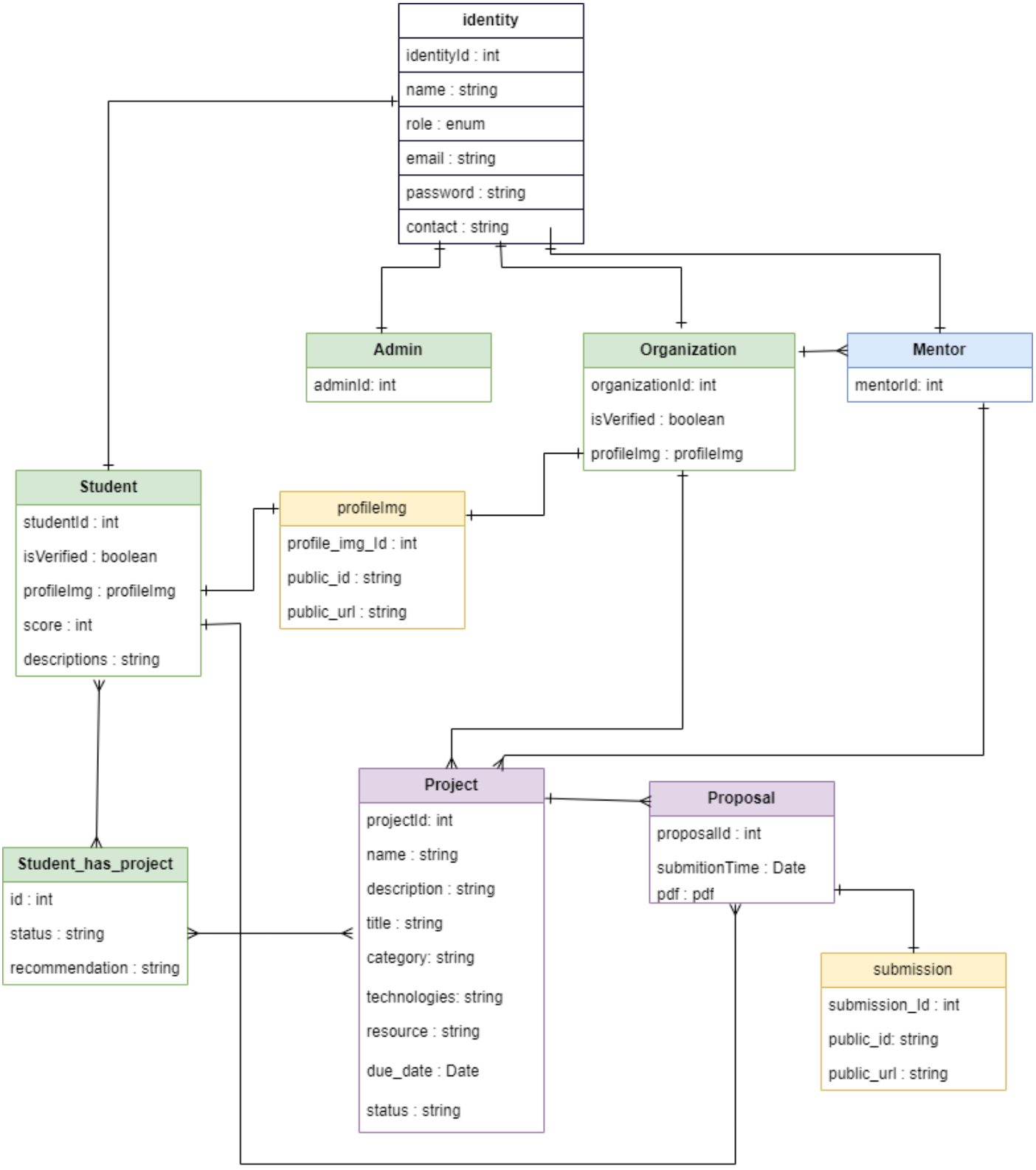

Figure 4: UniSource Class Diagram

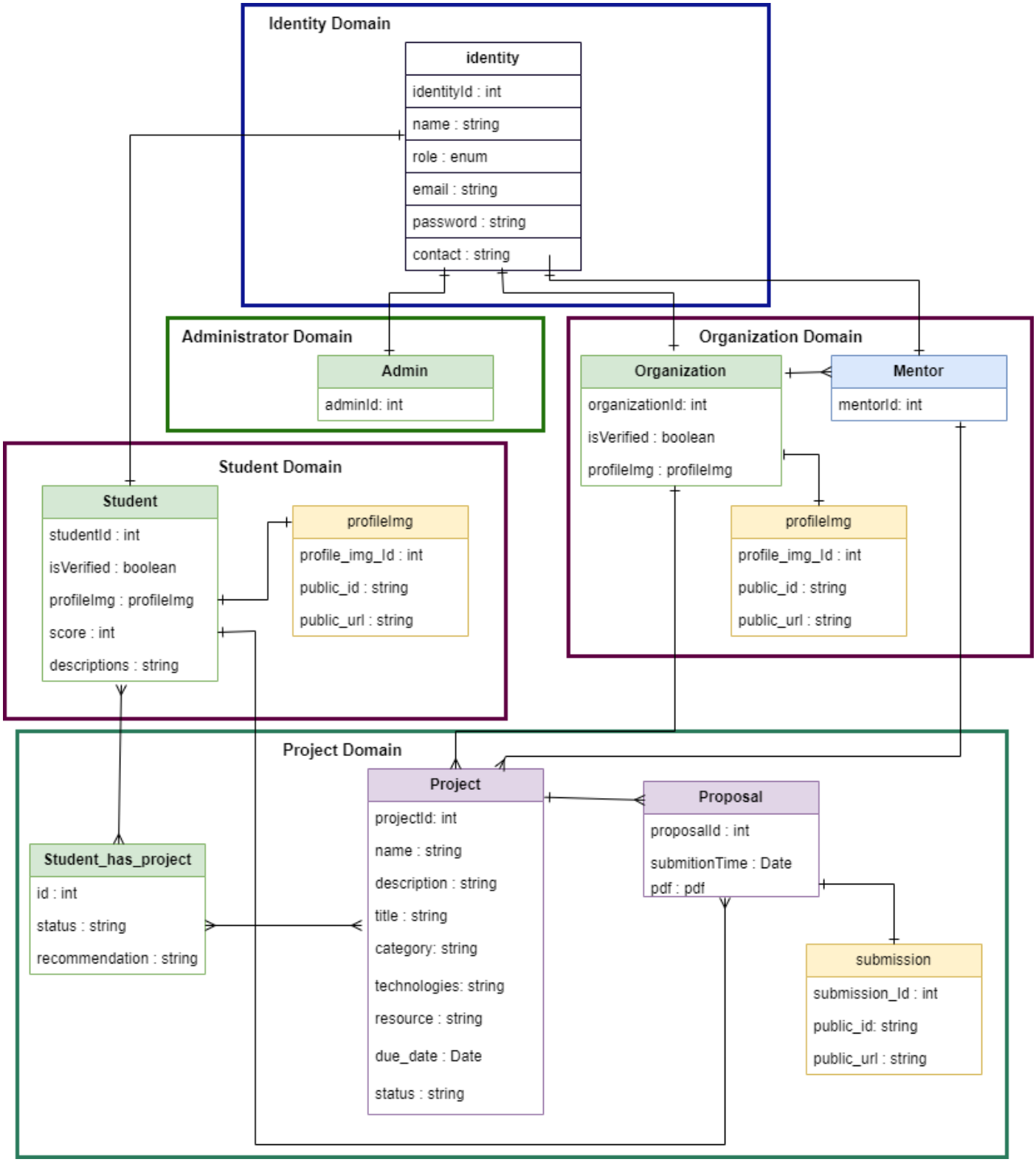

Figure 5: UniSource Domain Class Diagram

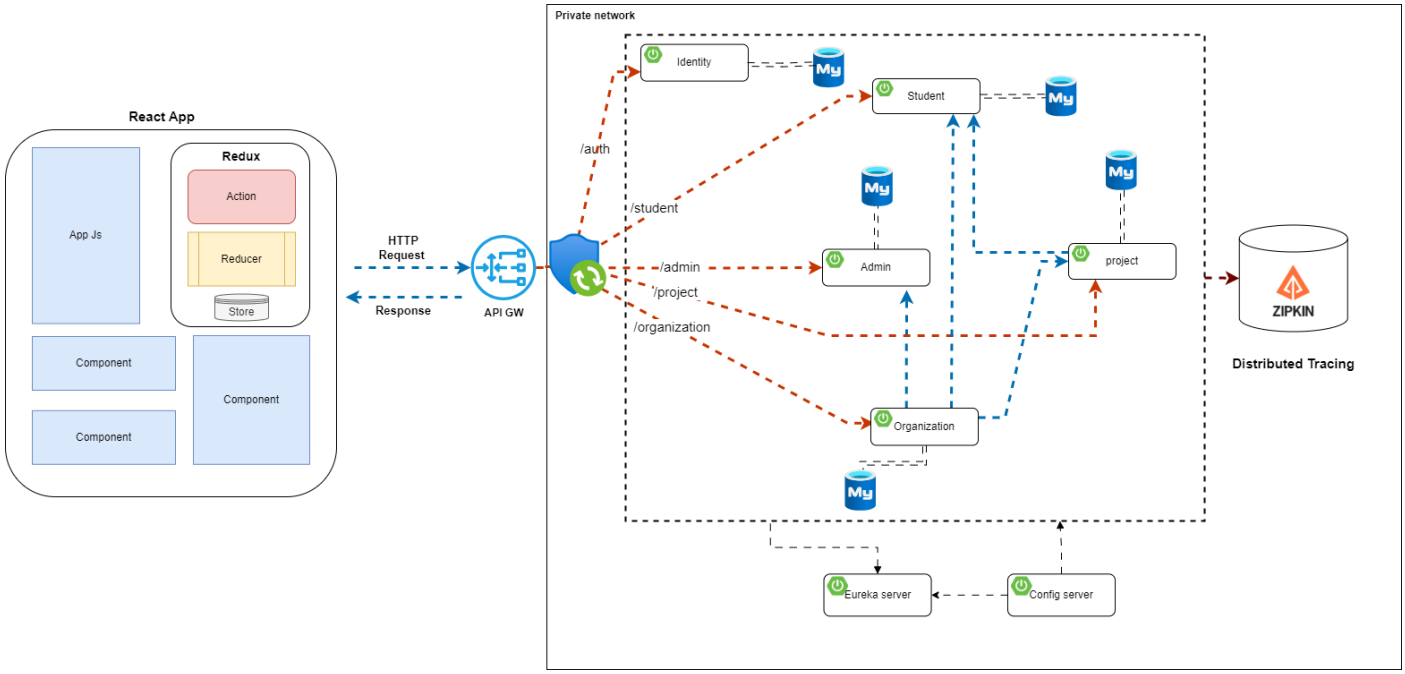

Figure 6: UniSource Global Architecture Diagram