SE1 Group Project - University Student Project Repository

Angelica Rings, Enzo Verdier Turrin, Louis Colbus, Robert Kiss, Sven Barnich

October 31, 2025

1 Topic Description

The project is a digital platform for storing and showcasing academic projects completed by students across different semesters and courses. It allows students to create accounts, using their university email address, and upload their project details including title, abstract, documentation, and media attachments such as images or videos. For group projects, the system provides an option to add the different collaborators and define their respective roles. These projects can be discovered and viewed by other students, professors, teaching assistants or external recruiters. Users can browse through the projects based on course, semester, or keywords. Another feature is for students (and other logged in users) to provide a 1-5 star rating for each project to assess its quality and highlight the best projects. Each student has a personal profile that aggregates all their submitted projects, creating a portfolio of their academic work.

2 Deliverable 1 - Requirements and Modeling

2.1 Requirements Elicitation

This section identifies and documents the needs and constraints of the project. We begin by identifying the key stakeholders and their interests, then translate those interests into actionable user stories, and derive specific functional and non-functional requirements that will guide the system's development.

2.1.1 Stakeholders

Name	Student
Type	Primary
Description	Main contributors who log in to the system to upload, manage, and display their academic projects. They provide project details such as abstracts, media files, and associated course information.

Name	Professors and Recruiters
Type	Primary
Description	Browse the repository to discover student projects. They need to search for students with specific skills (for instance "Java", "machine learning") demonstrated through project work. They value efficient search, clear project descriptions, and reliable contact information on student profiles.

Name	University
Type	External
Description	Owns and oversees the repository. The university's interest is to showcase the quality of its students' work, enhance its academic reputation, and provide a valuable resource for its community. It is concerned with brand consistency and data privacy.

Name	Website Administrator
Type	Hidden
Description	System maintainers responsible for the platform's health. They manage user accounts, monitor system activity, approve or remove content if necessary, and ensures the website operates securely and efficiently. Their primary concern is system security, stability, and maintainability.

2.1.2 User Stories

Name	Upload Project
Role	Student
Goal	Upload academic project, including source, abstract, media, topic
Reason	Showcase work to professors, recruiters, and other students
Scenario	*As a student, I want to upload my project with a title, abstract, and media files so that I can showcase my work to professors, recruiters and other students.*

Name	Manage Profile
Role	Student
Goal	To edit profile information including name, semester, contact
Reason	Such that others can correctly identify the student for academic and career opportunities
Scenario	*As a student, I want to edit my profile information, including my name, semester, and a short biography, so that others can correctly identify me for opportunities.*

Name	Browsing Projects
Role	Professors and Recruiters
Goal	Search and filter projects by topic
Reason	Quickly find relevant student work for evaluation or recruitment
Scenario	*As a professor/recruiter, I want to search for projects by course code and filter them by topic so that I can quickly find relevant work for recruitment.*

Name	Moderate Content
Role	Website Administrator
Goal	To approve new uploads and remove inappropriate or outdated content
Reason	So that the platform remains professional, accurate, and compliant with university standards
Scenario	*As a website administrator I want to review and approve new uploads and remove inappropriate or outdated content so that the platform remains professional, accurate, and compliant with university standards.*

Name	Rating Projects
Role	Student
Goal	Rate project other student's projects
Reason	Encourage students
Scenario	*As a student I want to give a star rating to a project I find impressive so that I can signal its quality to others and the student knows their work is valued.*

Name	View Portfolio
Role	Student
Goal	Access a single profile page with all of the students uploaded projects
Reason	Easily share student portfolio
Scenario	*As a student, I want to view all my uploaded projects on a single profile page so that I can easily share a link to my complete academic portfolio.*

Name	Add collaborators
Role	Student
Goal	Add collaborators to uploaded project
Reason	Credit all the students on a group project
Scenario	*As a student, I want to add my teammates as collaborators on a group project so that the project appears on all our profiles and we all get credit.*

2.1.3 Functional Requirements

ID	001
Type	Functional
Description	The system shall allow students to log in using their university email.
Need	Student
User Story	Upload Project

ID	002
Type	Functional
Description	The system shall enable students to upload project files including a title, abstract, media attachments, and course information.
Need	User
User Story	Upload Project

ID	003
Type	Functional
Description	The system shall allow students to edit or delete their own uploaded projects.
Need	Students
User Story	Upload Project

ID	004
Type	Functional
Description	The system shall provide a profile management page where students can update their personal and academic details (name, semester, bio).
Need	Student
User Story	Manage Profile

ID	005
Type	Functional
Description	The system shall provide a searchable project database that supports filtering by course, semester, and other free-text keywords.
Need	Professor/Recruiter
User Story	Browse Projects

ID	006
Type	Functional
Description	The system shall provide an interface for administrators to approve new project uploads before they become publicly visible.
Need	Website Administrator
User Story	Moderate Content

ID	007
Type	Functional
Description	The system shall allow administrators to remove inappropriate or outdated projects that have already been published.
Need	Website Administrator
User Story	Moderate Content

ID	008
Type	Functional
Description	The system shall maintain a log of all content modifications (upload, edit, delete) and administrative actions for auditing purposes.
Need	Website Administrator
User Story	Moderate Content

ID	009
Type	Functional
Description	The system shall allow students to add other students as collaborators to a project by providing their email address.
Need	Student
User Story	Add collaborators

ID	010
Type	Functional
Description	The system shall allow authenticated users to assign a 1-5 star rating to a project.
Need	Student
User Story	Rating Projects

ID	011
Type	Functional
Description	The system shall display a student's profile, including their personal details and a list of all projects they have uploaded or collaborated on.
Need	Student, Professors, Recruiters, TAs
User Story	Manage Profile, View Own Projects

ID	012
Type	Functional
Description	The system shall allow users to view project details, including the abstract, documentation, media attachments, and collaborators
Need	Professors, Recruiters, Students, TAs
User Story	Browse Projects

ID	013
Type	Functional
Description	The system shall display the average star rating on each project's page.
Need	Student
User Story	Rating Projects

2.1.4 Non-Functional Requirements

ID	014
Type	Non-Functional
Description	The system shall ensure secure access through HTTPS and all user passwords need to be salted and hashed when stored.

ID	015
Type	Non-Functional
Description	The system shall be responsive and fully functional on screen widths from 360px (mobile) to 1920px (desktop).

ID	016
Type	Non-Functional
Description	The system shall provide response times of less than 3 seconds for core user actions such as login, search, and project browsing.

ID	017
Type	Non-Functional
Description	The system shall be available with an uptime of at least 99.5% during university business hours.

ID	018
Type	Non-Functional
Description	The system shall support at least 200 concurrent users browsing and searching for projects without noticeable degradation in performance.

ID	019
Type	Non-Functional
Description	The system shall adhere to the Web Content Accessibility Guidelines (WCAG) 2.2 Level AA to be accessible for people with disabilities.

ID	020
Type	Non-Functional
Description	The system shall use a modular architecture to simplify maintenance and future feature additions.

ID	021
Type	Non-Functional
Description	The system shall log all file uploads and potentially malicious files.

ID	022
Type	Non-Functional
Description	The system shall support uploading files up to 500MB.

ID	023
Type	Non-Functional
Description	The system shall maintain a history for all content modifications (upload, edit, delete) and administrative actions.

ID	024
Type	Non-Functional
Description	The system shall store project data in a structured database for efficient retrieval and management.

ID	025
Type	Non-Functional
Description	The system database shall be backed up monthly, with backups retained for at least 60 days.

2.2 UML Diagrams

2.2.1 Use Case Models

The use case models illustrate each use case from the perspective of the user. The use cases we have for this project are: upload project, project rating, and content moderation.

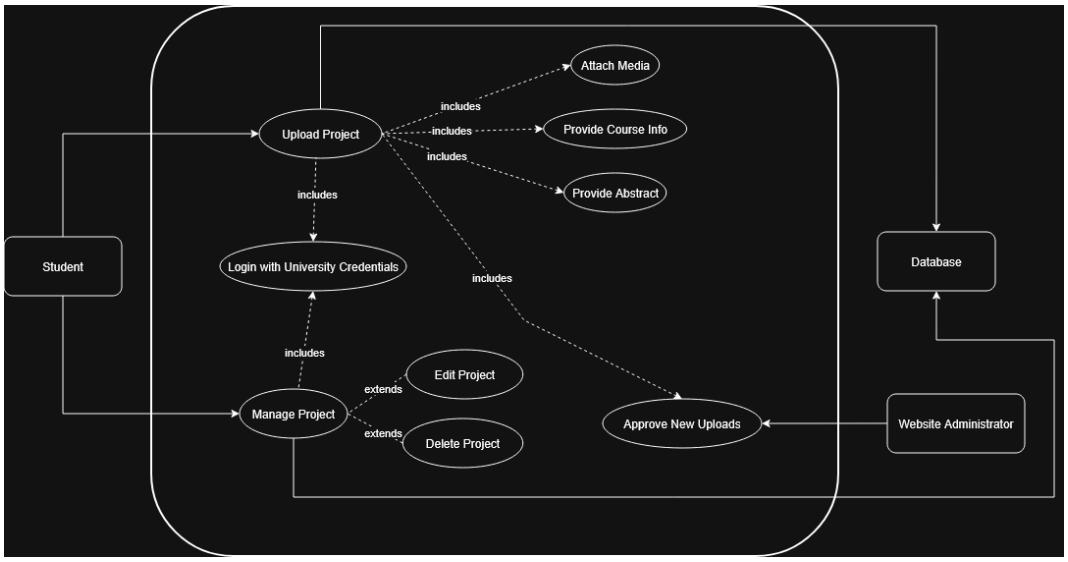

Figure 1: Use Case: Upload Project

The first use case model in Figure 1 displays the use case of a student uploading a project to the database for the student project repository. First, the student logs in, and then they can either manage their profile, upload a project, edit it, or delete it. When they upload the project, they can optionally add media attachments, the course information, and provide an abstract of the project. All of this will be stored in the database, where the website administrator can monitor it. Projects can only be uploaded after the administrator approves the new project or decides to remove it due to inappropriate material.

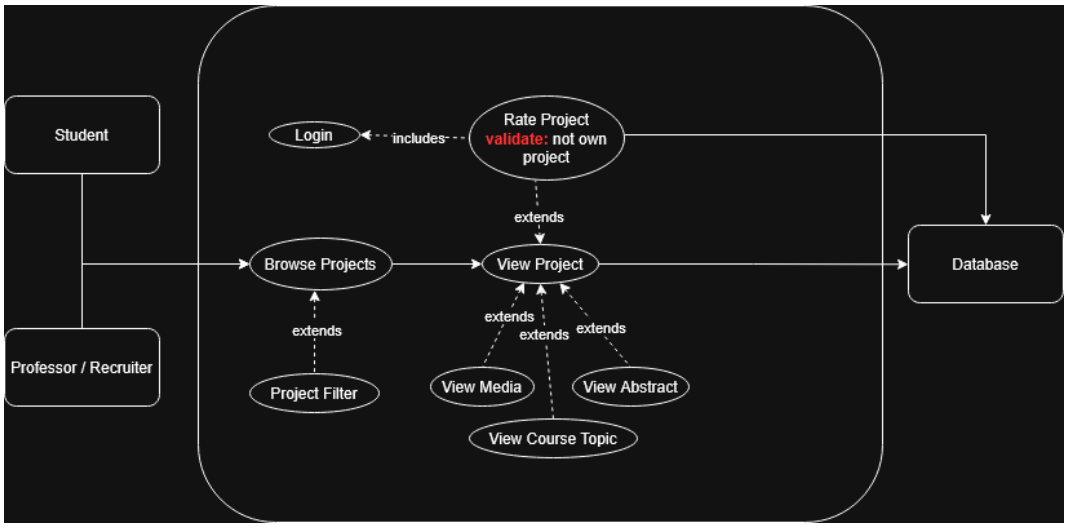

Figure 2: Use Case: Student, Professor/Recruiter Project Rating

The second use case model in Figure 2 displays the use case of students and professors/recruiters browsing through the projects. They log in and can browse through uploaded projects, and optionally apply filters to look for more specific projects. Then they can check out a project and view the media, abstract, and course topic. Furthermore, professors and recruiters can rate the projects of the students. This rating is stored in the database.

It is important to note that students are not allowed to rate their own projects.

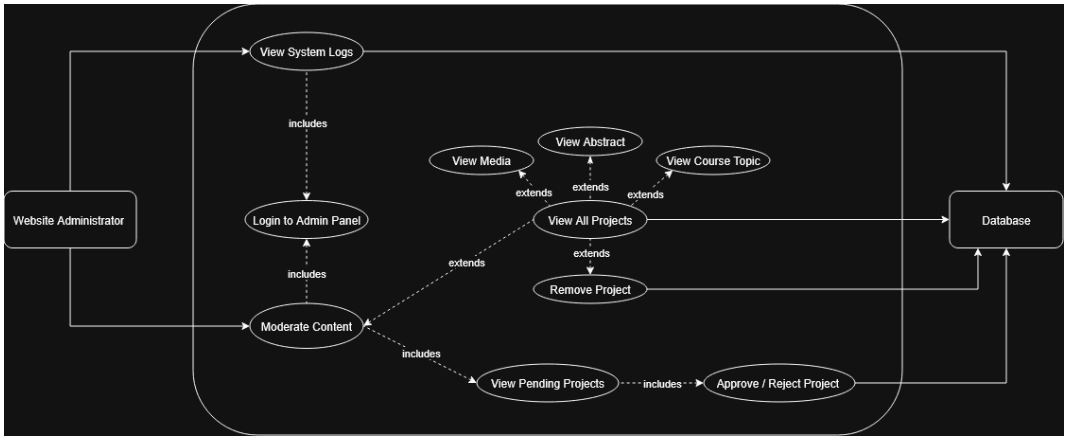

Figure 3: Use Case: Administrator Content Moderation

The third use case model in Figure 3 displays the use case of administrator content moderation, specifically enabling administrators to view all projects uploaded by students, including the attached media, abstract, and course topic, and an option of whether to delete a specific project. Additionally, administrators can see pending projects of students that still need approval or rejection. Finally, administrators also have access to the System Logs for further moderation.

2.2.2 Domain Models

The domain model (Figure 4) illustrates the different classes in the system along with their respective attributes and relationships.

The User class includes the attributes userID, name, and email. This class is a superclass that extends into three specialized types: Student, Administrator, and ExternalUser. The Student class adds the attributes semester and bio, and it is associated with the Project class, as students can author multiple projects. The ExternalUser class adds the attribute organization, representing users outside the university. The Administrator class, on the other hand, is responsible for approving projects.

The main aspect of the system is the Project class. It contains attributes such as projectID, title, abstract, uploadDate, status, averageRating and voteCount. The projects are authored by the Student and approved by the Administrator. Projects can also include MediaAttachments, which are defined by the attributes attachmentID, fileName, fileType, and url. The votes that were given to the project by users are recorded in Vote. It includes the attributes ratingValue and timestamp, allowing the system to track and evaluate project ratings over time. Projects are also tagged with one or more Topics, each having a topicID and name, enabling the classification and search. The Course class is characterized by courseCode, courseName, and semesterOffered. It acts as a reference for projects, showing which course or semester each project was developed under.

2.2.3 Behavioral Models

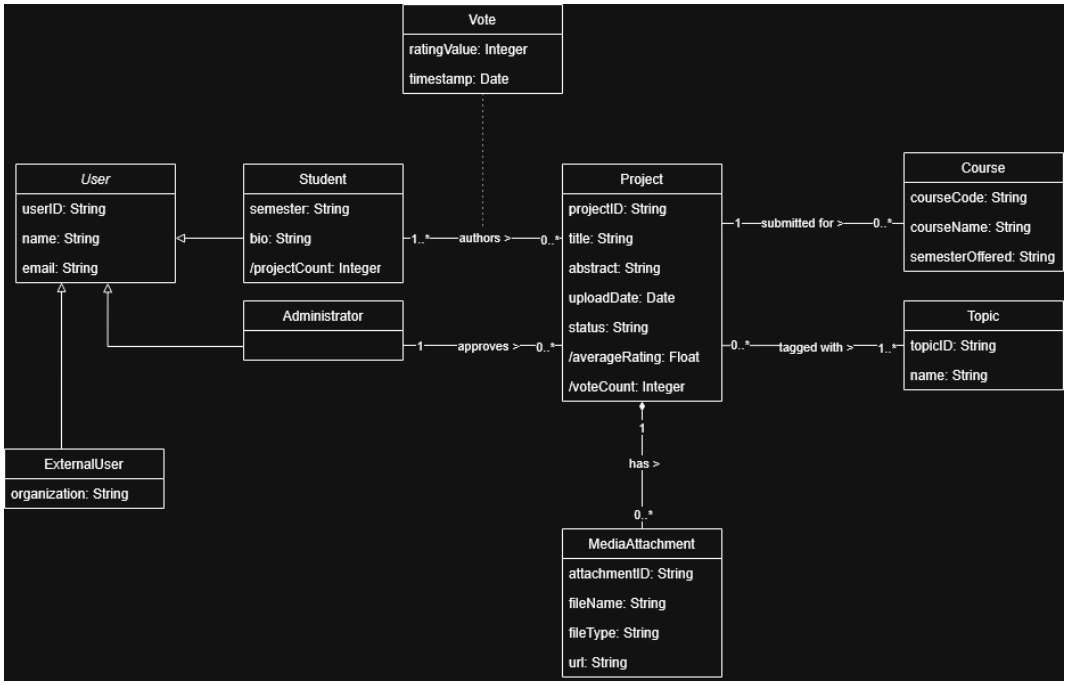

Figure 4: Class Diagram of the Project Platform

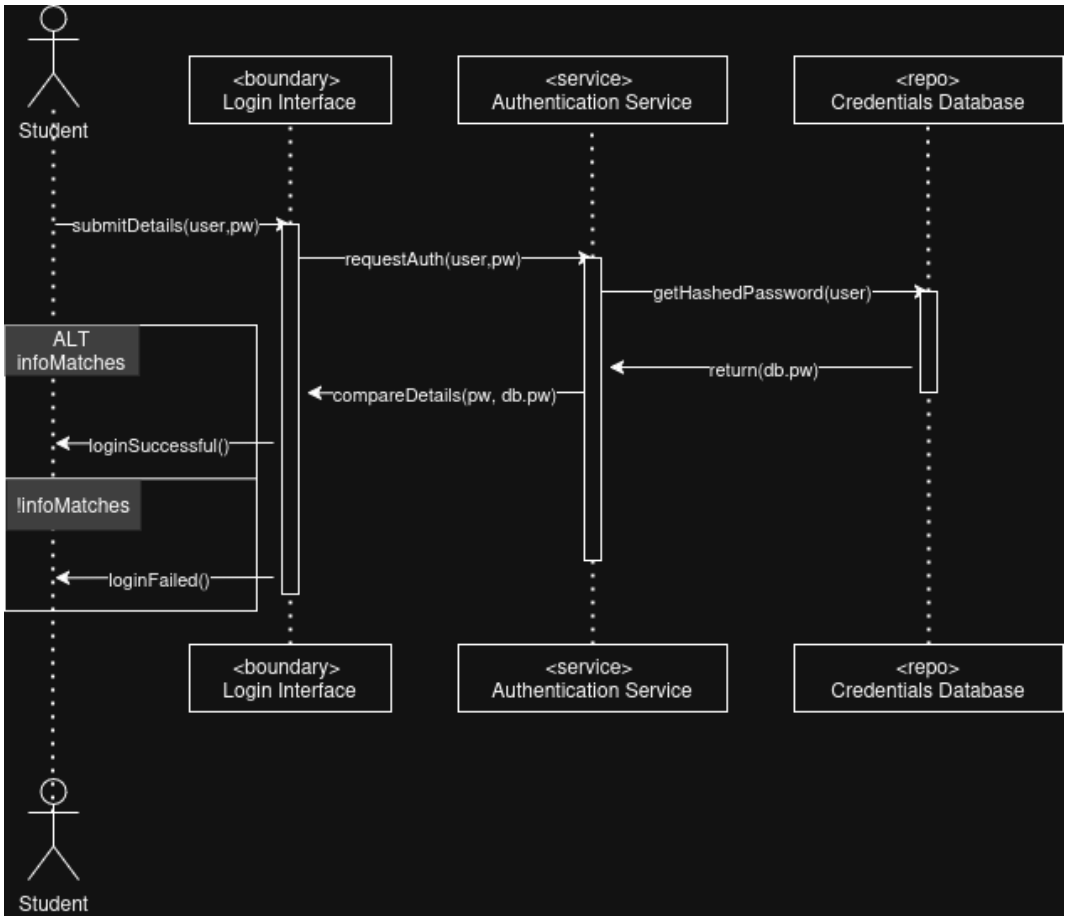

Figure 5: Login Sequence Diagram

The first behavioral model in Figure 5 displays the behavioral model for the login process for students accessing the system. Students provides their email and password using the web interface, which sends the data to the authenticator for validation. The authenticator retrives the user's credentials from the database and verifies if they are correct. If they match, the system grants access to the student. If not, an error message is returned.

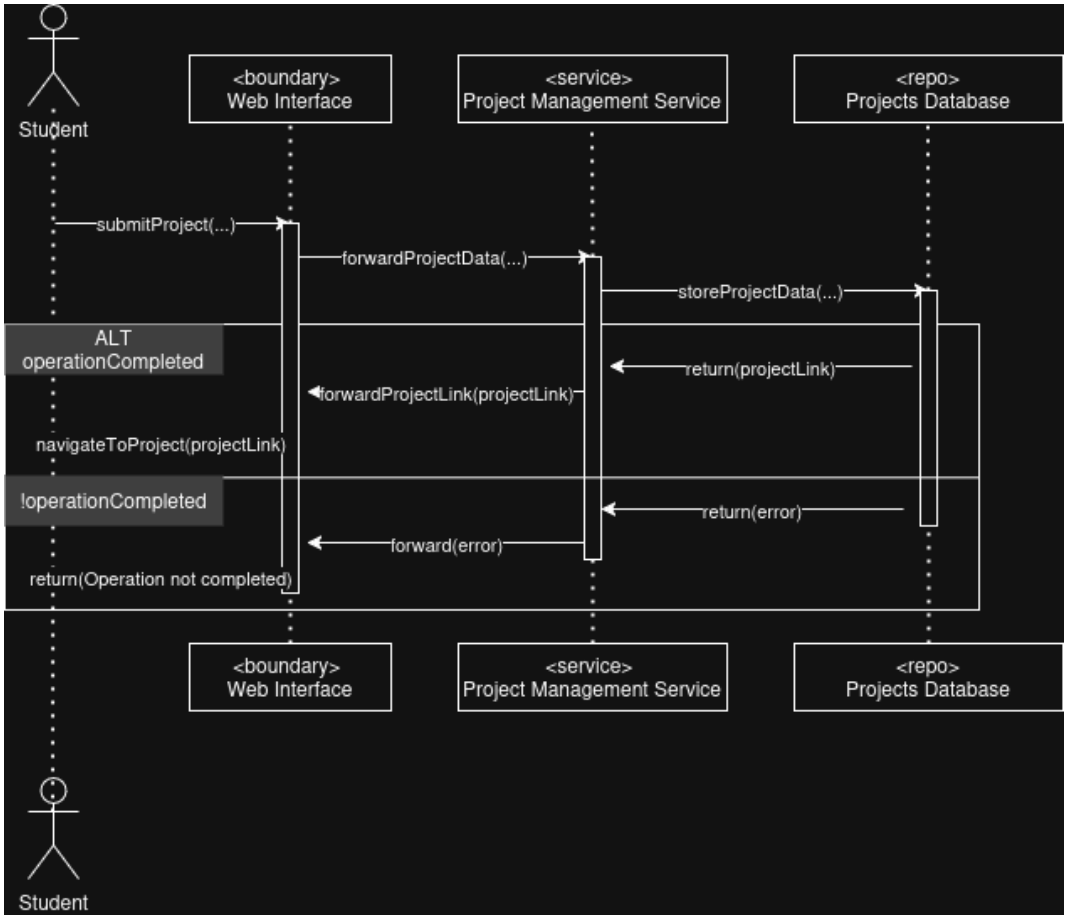

Figure 6: Upload Diagram

The second behavioral model in Figure 6 displays the behavioral model for the process by which a student uploads a new project. The student submits project details such as the title, the abstract, medias or the course information using the web interface. The system validates this data and stores the project in the database. Once the upload is complete, a confirmation message is returned to the student. If the upload couldn't be validated or if an error occurred, an error message is displayed.

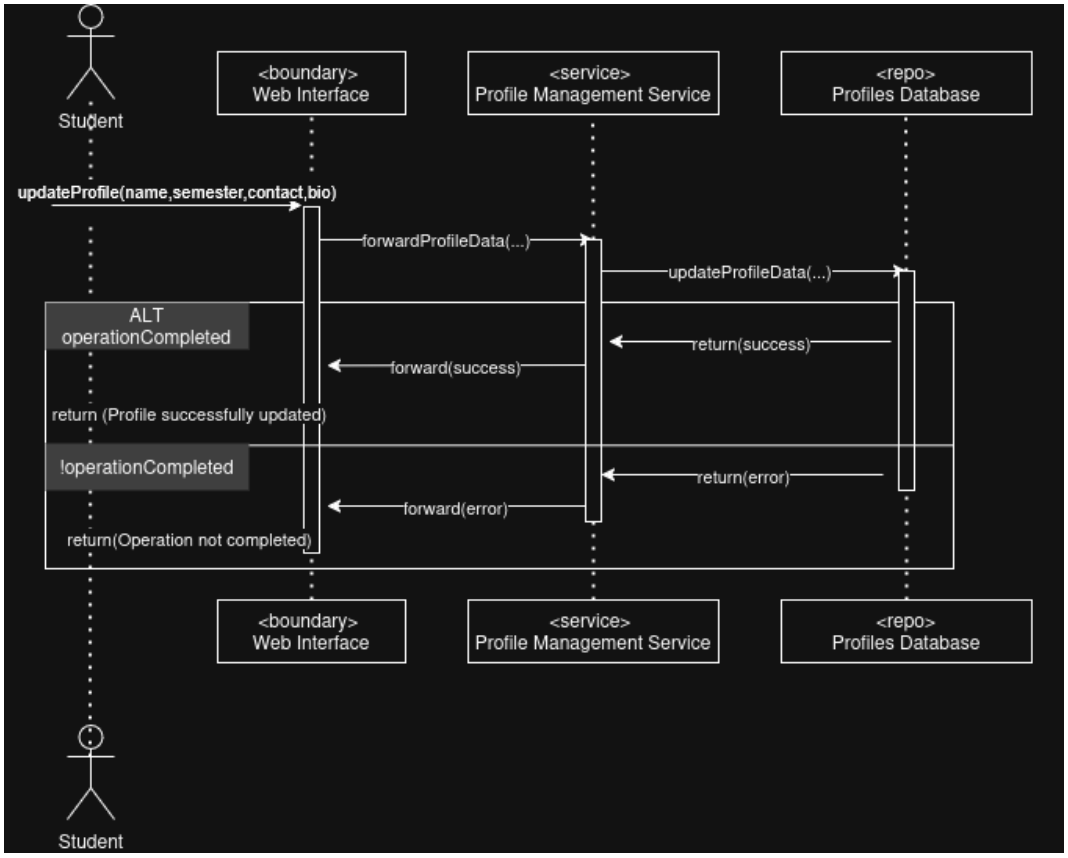

Figure 7: Manage Profile Diagram

The third behavioral model in Figure 7 displays the behavioral model for the process of managing a student's profile informations. The student can upload details such as their name, semester or contact information using the web interface. The date is validated and saved in the database. The system then confirms the success or failure of the update operation.

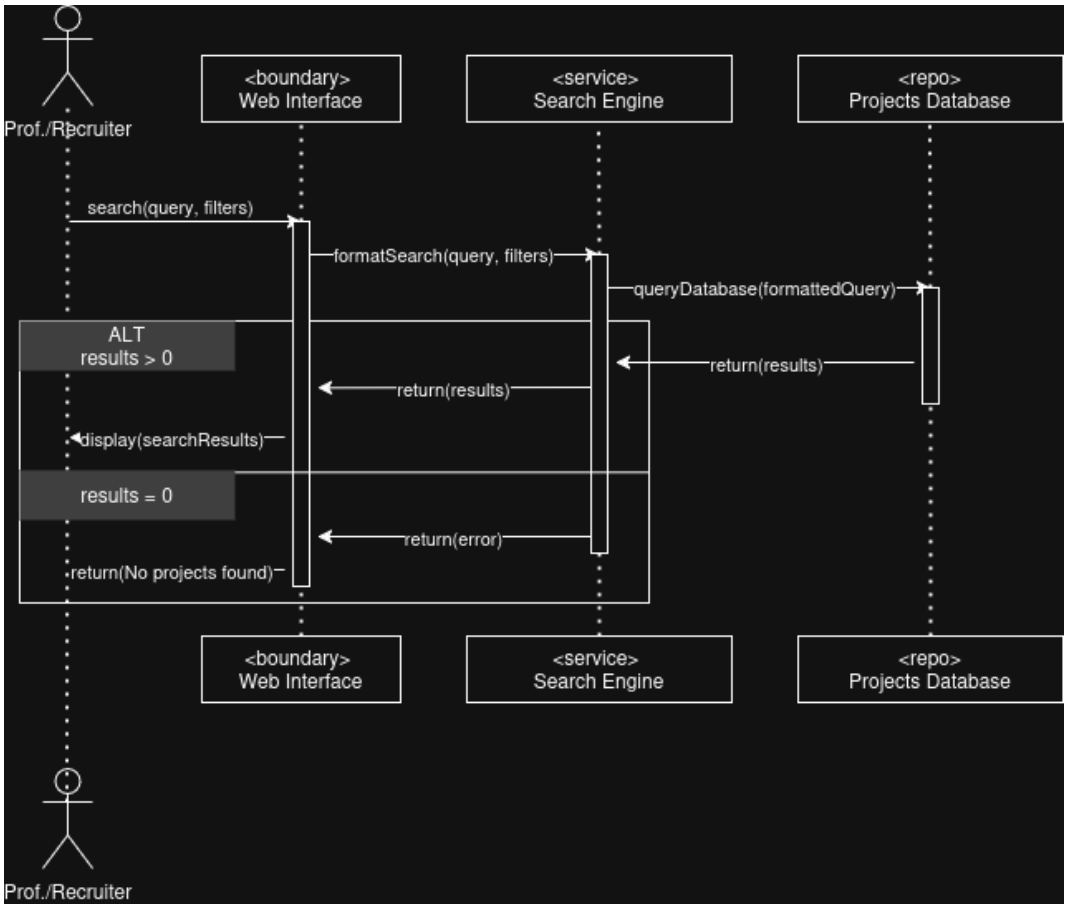

Figure 8: Browsing Projects Diagram

The fourth behavioral model in Figure 8 displays the behavioral model for the process of browsing and filtering the uploaded projects for the professors and recruiters. After logging in, they can apply filters and keywords to narrow down the search results. The query is processed by the system and the matching projects are retrieved from the database and displayed.

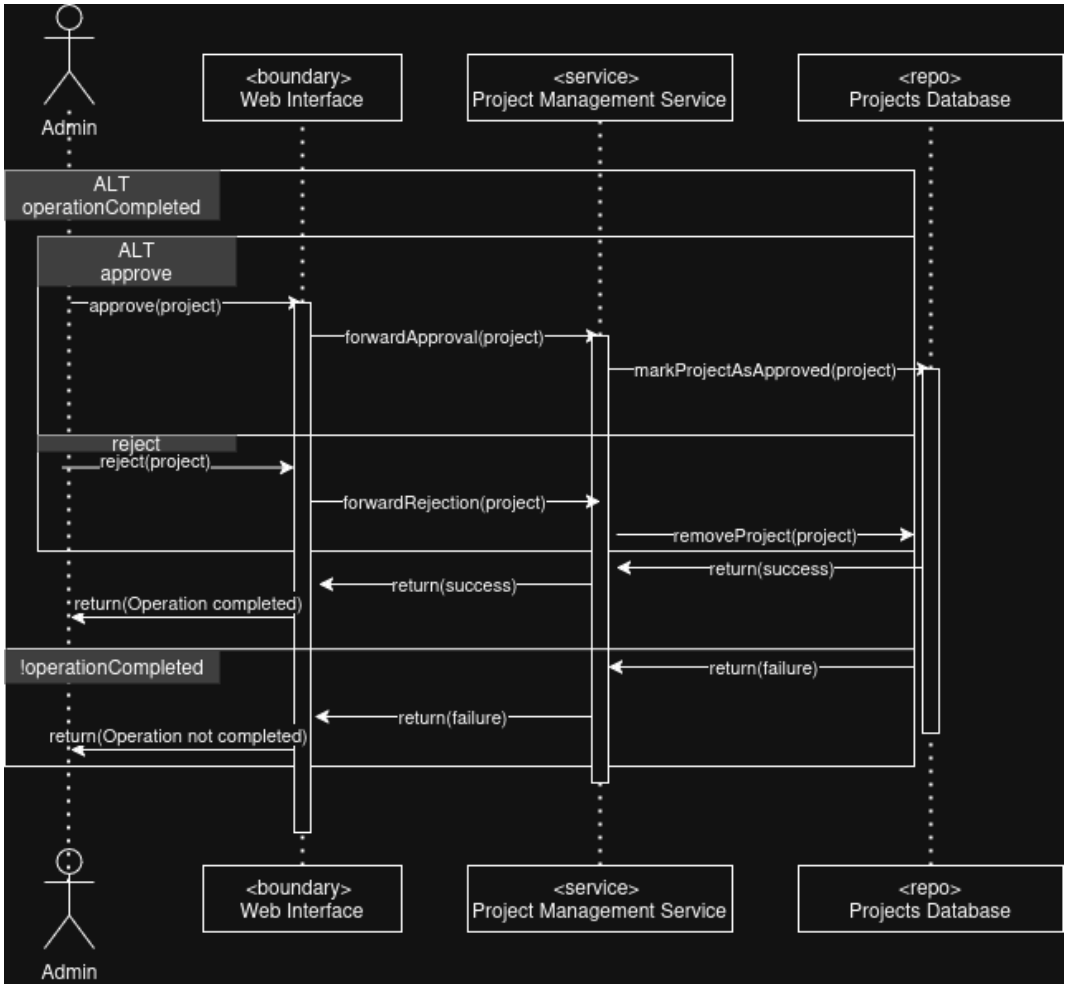

Figure 9: Moderate Content Diagram

The fifth behavioral model in Figure 9 displays the behavioral model for the process of moderating the contents for administrators. The administrator can approve new uploads or remove outdated or inappropriate projects using the web interface. The actions are processed by the system and reflected in the database. A confirmation message is then returned to indicate if the operation was successful.

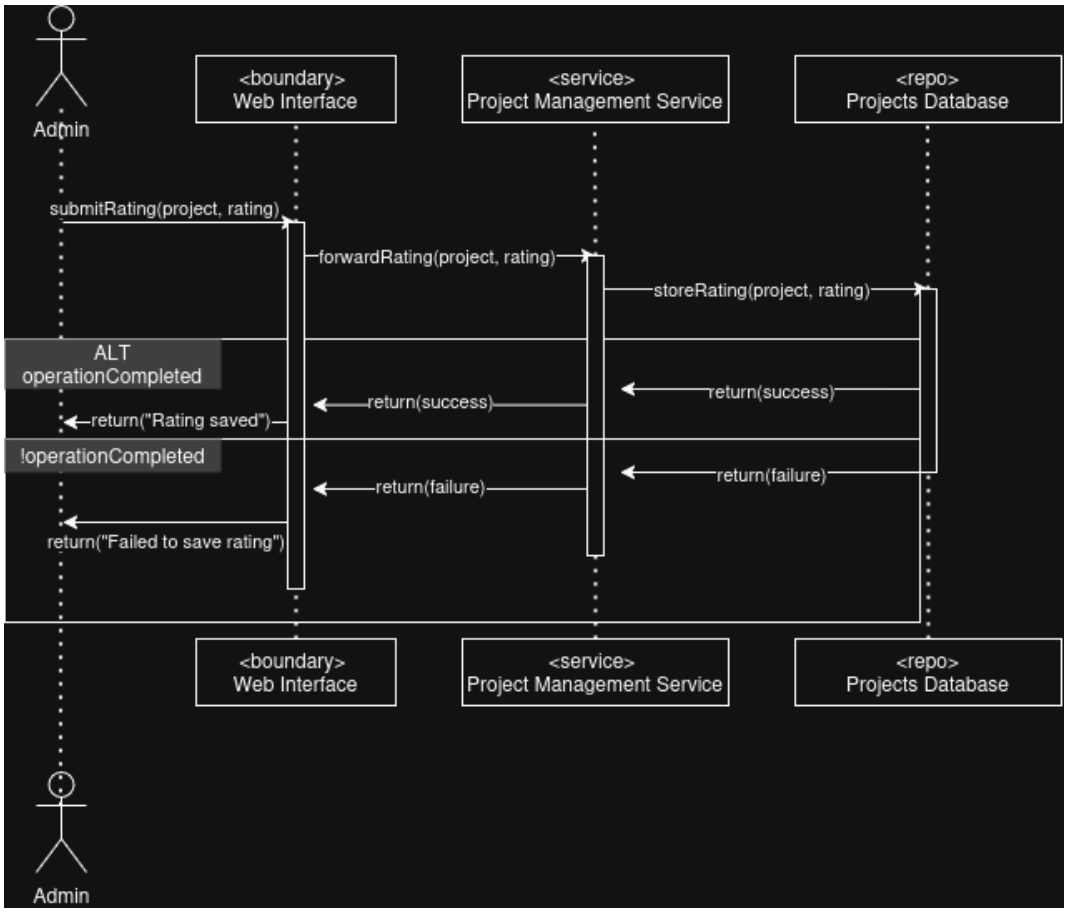

Figure 10: Rate Projects Diagram

The sixth behavioral model in Figure 10 displays the behavioral model for the process of rating student projects by students. After viewing a project, they can assign a rating using the web interface. The system validates the rating and stores it in the database, associating it with the corresponding project. The user then receives confirmation of the successful submission, or an error message if the rating submission failed.

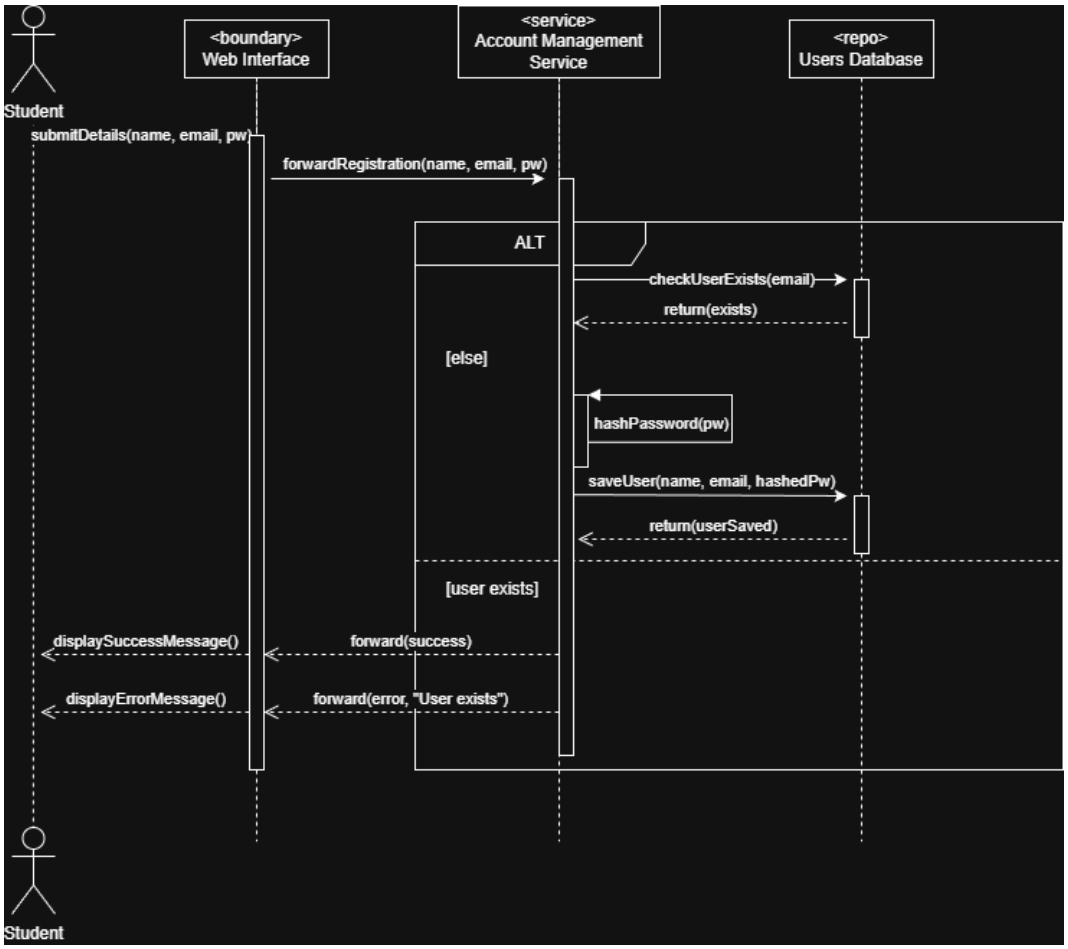

Figure 11: Create Account Diagram