Students&Companies Project

RASD

Student	ID
Guillermo Larsen	11077025
Yichen Lu	10764748
Mohamed Elbanna	10726137

Lesson
Software engineering II

Teacher:	Parallel
Matteo Giovanni Rossi	E-O

Date
9 de diciembre de 2024

Índice

1. Introduction

1.1. Purpose

Finding internships is becoming increasingly challenging for students as the number of companies in the world grows and human capacity to review all available options remains limited. This creates a need for a system that simplifies the search for opportunities, benefiting students and companies.

The system aims to streamline interactions between users and support the processes of searching, selecting, monitoring, and managing internships. It helps students find suitable opportunities more easily, assists companies in reaching potential candidates, and ensures smooth coordination throughout the internship process.

1.1.1. Goals

[G1] Students can search for and apply to internships.

[G2] Students receive recommendations based on their skills, experience, future goals, keywords and statistical analysis.

[G3] Companies can publish internship opportunities.

[G4] Companies receive candidate recommendations based on their stated needs , students'CVs, keywords and statistical analysis.

[G5] Students and companies can schedule and manage interview dates, including the use of questionnaires for assessment.

[G6] Students, companies, and universities can view and monitor the status and progress of internships.

[G7] Students, companies, and universities can report issues, file complaints, and provide feedback.

[G8] Universities can manage complaints and, if necessary, interrupt the internship to resolve critical issues.

1.2. Scope

The S&C platform has three types of users: Students, Companies, and Universities.

Students use the system to search for, apply to, and track internship opportunities. They can create a profile where they upload their CV, specify skills, future objectives, attitudes, and past work experiences. There are two options for searching: active and passive. In both cases, the profile information enhances recommendations. The active search option allows students to use keywords to find opportunities through the search function, while the passive search option analyzes their interaction history on the platform. This way, the system can send alerts to students about potential internships that match their preferences. When students find an interesting internship opportunity, they can apply. The system then assists them in scheduling interviews and responding to the company questionnaire.
While in an internship, students can update their progress and communicate with their university or the company to report any issues or concerns.
After completing an internship, students can provide feedback about their experience. This feedback helps the system analyze and improve the recommendation engine for future internships, enhancing the S&C

platform's overall user experience.

Companies use the system to post internships, search for candidates, and manage the selection process. When a company wants to post an internship, it needs to create a detailed listing. This includes specifying the project domains, required skills, tasks, and benefits (such as training, mentorship, or stipends).
Companies can search for candidates in two ways, both based on CVs, required skills, and past experiences. The first option is active: the system searches for candidates using the keywords that the company enters in the search function. The second option is passive: the system sends notifications to the company about potential candidates that match their requirements.
If a company wants more information about a student, it can view the student's profile to verify details and read additional information the student has shared.
The system also assists companies in organizing interviews, sharing company questionnaires with candidates, and facilitating communication between companies and students.
During the internship, the company can update the student's status and communicate any issues to the student or their university.
After the internship, the company can receive feedback from the student or university, as well as provide feedback about the student's performance or the system's functionality.

Universities use the system to supervise internships, monitor student progress, and manage any issues between the company and the student. The university has access to the activities, processes, and tasks assigned to the student during the internship. If an issue arises involving the student or company, the university receives a notification and can intervene if necessary.
When the internship ends, the university receives feedback from both the company and the student. This feedback helps the university improve its internal internship policies and support.

1.2.1. World Phenomena

[WP1] Users have access to an internet connection.

[WP2] Users have the necessary hardware to connect to the system.

[WP3] Students can search for internships.

[WP4] Companies can search for candidates.

[WP5] Universities can intervene in issues between students and companies.

[WP6] Universities supervise students' internships.

1.2.2. Shared Phenomena

World Controlled:

[SPWC1] Users can interact with the system.

[SPWC2] Students can update their profiles with information such as CVs, skills, experiences, attitudes, and future goals.

[SPWC3] Students can find interesting internships and apply for them.

[SPWC3] Companies can post internship offers with details, including descriptions, project domains, required skills, tasks, and benefits (e.g., training, mentorship, stipends).

[SPWC4] Students can use keywords in active searches to find internship opportunities.

[SPWC5] Companies can use keywords in active searches to find candidates for internships.

[SPWC6] Students can monitor their progress during internships.

[SPWC7] Students can communicate with their university or company to report issues.

[SPWC8] Students can provide feedback about their internship experiences.

[SPWC9] Students can respond to company questionnaires.

[SPWC10] Companies can view student profiles to verify details and access additional information.

[SPWC11] Companies can provide feedback about student performance.

[SPWC12] Companies can give feedback about the system.

[SPWC13] Companies can search for candidates using active search by entering keywords.

[SPWC14] Companies can report issues to students or universities.

[SPWC15] Companies can update a student's status in the system.

[SPWC16] Companies can update interview dates in the system.

[SPWC17] Companies can share questionnaires with students.

[SPWC18] Universities can access information about the activities, tasks, and progress of students during internships.

[SPWC19] Universities can receive feedback from students and companies after internships are completed.

Machine controlled:

[SPMC1] The system shares the interview date and the company questionnaire with the student.

[SPMC2] The system facilitates communication between users.

[SPMC3] The system assists in active candidate searches.

1.2.3. Machine Phenomena

[MCP1] The system provides recommendations to students for improving their profiles.

[MCP2] The system sends notifications to students about potential internship opportunities.

[MCP3] The system notifies companies about potential candidates for internships.

[MCP4] The system informs universities about new feedback or issue reports.

[MCP5] Passive search has access to the search history of active searches.

[MCP6] Passive search analyzes the search history of active searches to identify patterns.

[MCP7] Feedback provided by students and companies is analyzed to enhance recommendation algorithms.

[MCP8] The system assists companies in searching for candidates using passive search.

1.3. Definitions, Acronyms, Abbreviations

1.3.1. Definitions

1.3.2. Acronyms

- CVs : Curriculum Vitae
- S&C : platform Students&Companies

1.3.3. Abbreviations

- MCP = Machine Phenomena
- SPMC = Share Phenomena, Machine Control
- SPWC = Share Phenomena, World Control
- WP = World Phenomena
- D = Domain assumptions
- R = Requeriment
- UC = Use Case

1.4. Revision history

– **V1.0** (December 22, 2024): First delivery

1.5. Reference Documents

- Assignment RDD AY 2024-2025

1.6. Document Structure

1. **Introduction:** A brief presentation of the problem's context. This section includes the goals, phenomena, descriptions, notations, and acronyms relevant to the project.
2. **Overall Description:** An overview of the domain assumptions and various scenarios, accompanied by class diagrams and state diagrams. Additional details may also be provided as needed.
3. **Specific Requirements:** This section elaborates on the aspects introduced in Section 2, offering details useful for the development team. It includes use case diagrams, detailed use cases, related sequence and activity diagrams, and a mapping of these to the requirements. The necessary system interfaces are also specified here.
4. **Formal Analysis Using Alloy**: A detailed presentation of the main objectives, a formal model of the system, and a verification of its correctness and consistency using Alloy.
5. **Effort Spent:** This section provides information about the number of hours each group member contributed to creating this document.

2. Overall Description

2.1. Product perspective

2.1.1. Scenario

1. **Unregister Student sign-in to application**

   Leonardo, a student at Politecnico di Milano, opens the application and attempts to sign in to the system. However, an error message appears: "This user does not have an account."The platform displays an option to register. He clicks on the option and is redirected to the registration window. There, he enters his email, password, and role (student or company). After filling in the required fields, he clicks the **Register** button. The platform then sends an email for verification.

   Leonardo checks his email, opens the message, and clicks on the verification link. The link redirects him to the application, where he sees the student registration options. Here, he needs to input additional details: his full name, ID number, and university. Once he completes all the required information, he clicks the Save button. The system redirects him to his profile page, where Leonardo decides to edit and complete his profile.

   He clicks on the **Edit** button, and the application prompts him to provide his CV, skills, experiences, attitudes, and future goals. Once he has entered all the necessary information, he clicks the **Save** button again.

   With everything set up, Leonardo is ready to start searching for internships or receive tailored recommendations from the platform.

2. **Unregister Comapny sign-in to application**

   NovaTech Innovations is eager to attract talented college students through internship opportunities, using this approach to identify potential future employees. To make this happen, an employee from the Human Resources department is tasked with publishing an internship opportunity on the platform.

   The employee opens the application and attempts to sign in. However, an error message appears: "This user does not have an account."The system displays an option to register, and the employee clicks the **Register** button, which redirects them to the registration page.

   On the registration page, the system prompts the employee to provide their email, password, and client type (student or company). After submitting the required information, the system sends a confirmation email to the provided address.

   The employee then checks their inbox, opens the confirmation email, and clicks on the verification link. This link redirects them to the company's registration form, where additional details are required: the name of the company, the sector it operates in, its mission, and an overview. Once all the information is filled in, the employee clicks the **Save** button. The system then redirects them to the company's profile page.

   On the profile page, the employee decides to refine the profile further. By clicking the **Edit** button, they upload the company's logo and review the previously entered details, correcting any errors. After completing these updates, the employee clicks Save again to finalize the profile.

   With the company profile successfully set up, NovaTech Innovations is ready to leverage the platform. The Human Resources employee can now create and publish internship postings, enabling the company to connect with talented college students and identify promising candidates for future roles.

3. **Internship publication**

The company NovaTech Innovations wants to post an internship on the platform to attract new talent. To achieve this, a representative from NovaTech Innovations needs to publish an internship offer.

The company representative opens the application and clicks on the **Sign-In** option. The system redirects them to the sign-in page, where they enter the email and password associated with the company account. After entering the credentials, they click the **Sign-In** button to access the platform.

Upon signing in, the worker is taken to the company's profile page. From there, they navigate to the option to post an internship. The system prompts them to fill out the required information, including:

- A description of the internship
- Project domains
- Required skills
- Tasks to be performed
- Benefits offered (e.g., training, mentorship, stipends)
- A meeting schedule
- The company's questionnaire for candidates

After carefully completing all the required fields, the worker clicks the **Publish** button to make the internship offer available on the platform

4. **Internship Application and Recommendation**

   Leonardo needs to complete an internship because it is a university requirement. To achieve this, he actively searches for internships on the application using specific keywords. One day, the system sends him an alert about an internship that matches his previous searches and aligns with his profile.

   Curious about this opportunity, Leonardo clicks on the notification and accesses the internship publication, where he can view key information about the position, including:

   - A description of the internship,
   - The company offering the internship,
   - Project domains,
   - Required skills,
   - Tasks to be performed,
   - Benefits offered (e.g., training, mentorship, stipends).
   - The company's questionnaire for candidates

   After careful consideration and complete the questionnaire, Leonardo decides to apply for the internship by clicking the **Apply** button.

5. **Company respond ad postulation**

   A few days after Leonardo submits his application through the platform, the company reviews the submitted materials. This includes examining the student's profile, internship application, and responses to the questionnaire to verify if the applicant meets the eligibility criteria. Once this evaluation is complete, the company can choose between two actions: clicking on the **Accept** button to approve the application or the **Reject** button to decline it. In Leonardo's case, his application is accepted.

6. **Company report a issue**

Matias, a student at the Catholic University of Milan, frequently arrives late during his internship. As a result, the company files an issue report through the system. Both Matias and his university receive a notification regarding the report.

In response, Matias stops attending work and falsifies a medical leave request as a form of protest. Instead of addressing the concerns, he goes on vacation to another location.

The company discovers this misconduct and files a second issue report, prompting the university to intervene. After reviewing the evidence, the university decides to terminate Matias's internship due to his lack of professionalism, breach of trust, and unethical behavior.

7. **Student report a issue**

During his internship, Leonardo encounters a problem related to his coworkers. As a result, he decides to file an issue report through the system. The system promptly sends an alert to both Leonardo's university and the company.

Upon receiving the alert, the company takes immediate action to address the situation. They speak with Leonardo to understand the details of the issue and work toward a resolution.

In the end, the problem is successfully resolved, and Leonardo completes his internship without further complications.

8. **Feedback**

After Leonardo completes his internship, both he and the company receive notifications from the system requesting their feedback.

- **For Leonardo:** The system prompts him to share his opinion about his internship experience and the functionality of the application.
- **For the Company:** The system asks for feedback on Leonardo's performance as an intern and their experience using the application.

Once both the company and Leonardo submit their feedback, the university receives a summary of the responses. This information allows the university to evaluate the student's performance, identify areas for improvement in the internship program, and determine whether any updates to their internship policies are necessary.

2.1.2. Class Diagrams

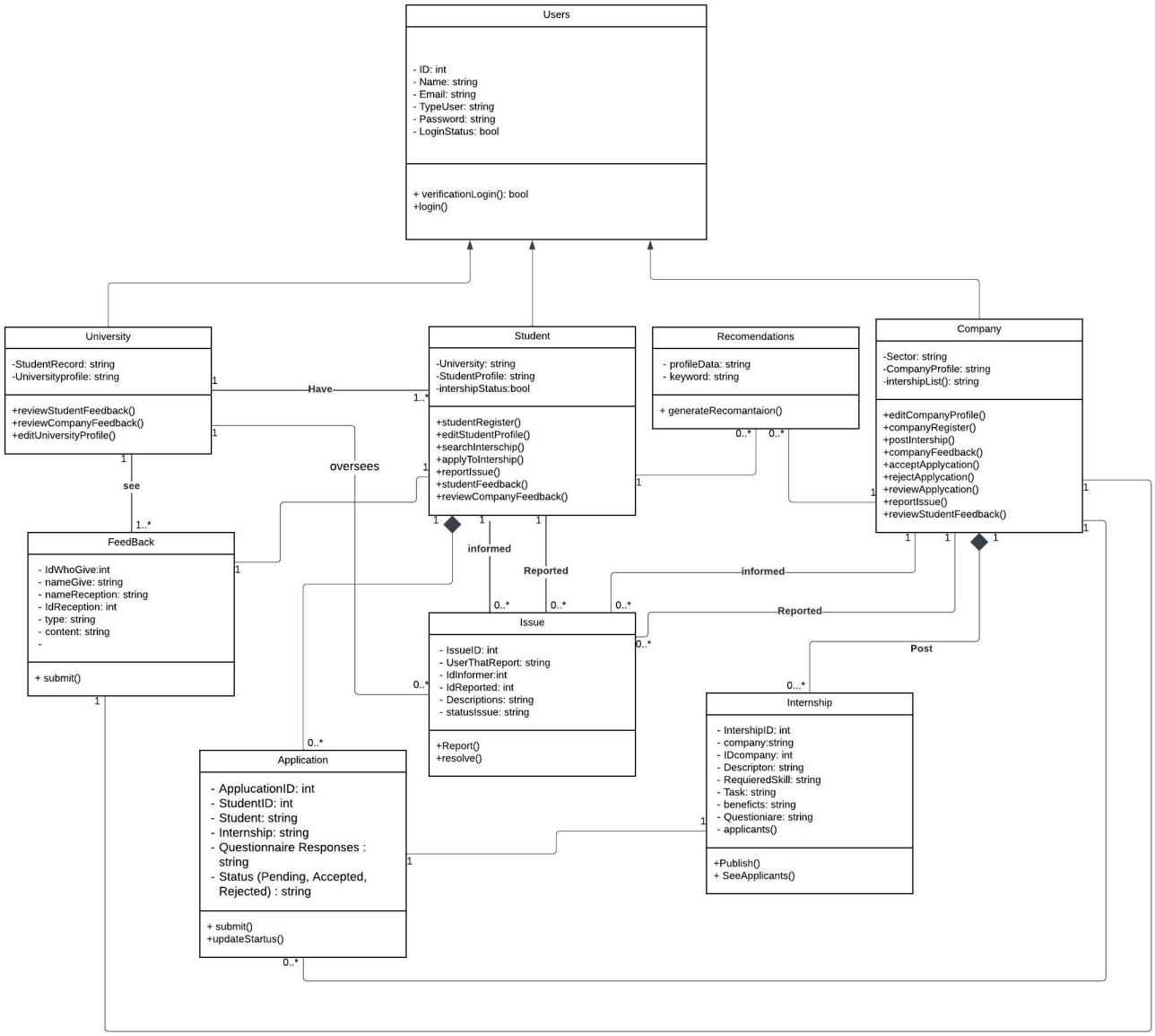


Figura 1: Class Diagram

This class diagram models a system with three primary types of actors: University, Students, and Company. These actors are generalized into a parent class called Users, with three specialized subclasses: University, Student, and Company.

The system includes additional classes to manage functionality.

- Feedback
- Issue
- Application
- Internship

- Recommendation

Class Relationships and Interactions:

1. **Users and Subclasses:**
   - University, Student, and Company inherit common attributes and methods. They are Users subclass.
2. **Internship and Company:**
   - Internship cannot exist without a Company,composition relationship, If the Company is removed, its associated Internships are also deleted
   - Company can post zero or more Internships, but each Internship is associated with exactly one Company
3. **Recommendations:**
   - Recommendations are exclusive to the Company for Internships they post. A Company can receive zero or more Recommendations, and each Recommendation is linked to a specific Internship, as a consequence to a company.
4. **Issues**
   - Issues can be informed or reported by a Student or Company. Each Issue is associated with exactly one Company or Student, but a Company or Student can public or receive multiple Issues.
   - Issues are also associated with the University of the Student performing the internship. The University oversees multiple Issues related to its Students.
5. **Applications:**
   - Applications are compositions of the Student and are only created if the Student applies for an Internship.
   - Each Application is linked to one Internship, and every Internship receives one Applications for student.
   - A Student can apply to multiple Internships, but each Application is associated with exactly one Student.
6. **Feedback:**
   - Feedback is given by both Students and Companies. A Company provides feedback on a Student's internship performance, and a Student gives feedback about the internship experience.
   - Each Feedback is associated with one Student or Company and is tied to the respective Internship.
   - The University can view all feedback provided by its Students and Companies
7. **University and Students:**
   - Each Student is associated with exactly one University.
   - A University can have many Students and oversees the Feedback and Issues related to their Students.

2.1.3. State diagrams

Internship and application

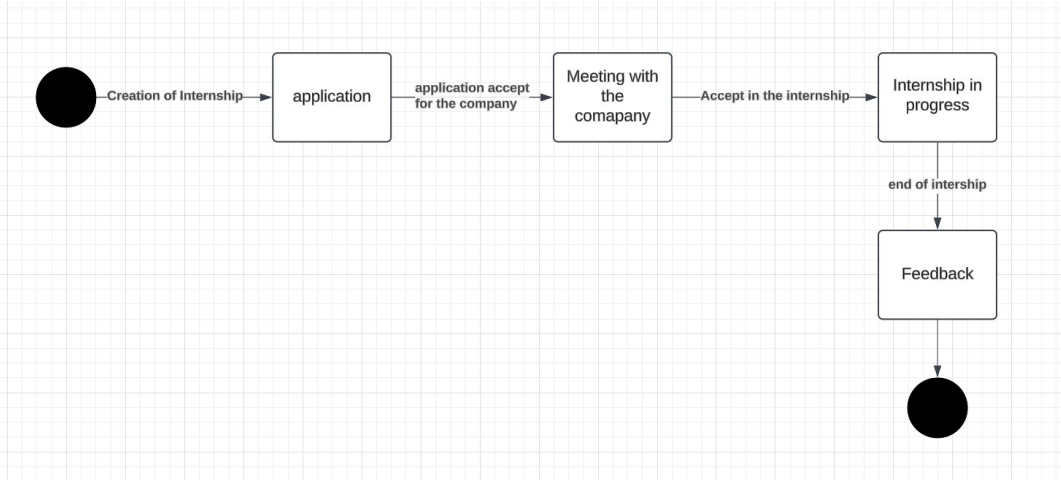


Figura 2: state diagram:application and internship

The company creates and posts an internship opportunity. Students then apply for the position, and the company reviews the applications. Once a candidate is selected, the company holds a meeting with the student. Afterward, the student completes the internship. Finally, both the student and the company provide feedback.

Report a issue

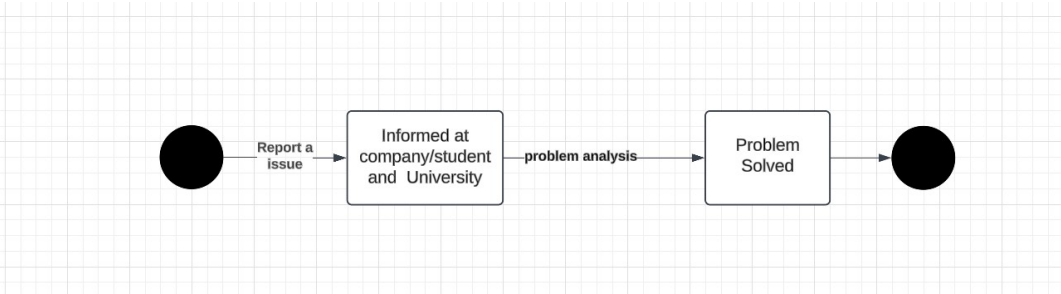


Figura 3: state diagram: Issue

If the student or company reports an issue, the other parties are informed. They work together to resolve the problem in a way they deem appropriate.

Registration

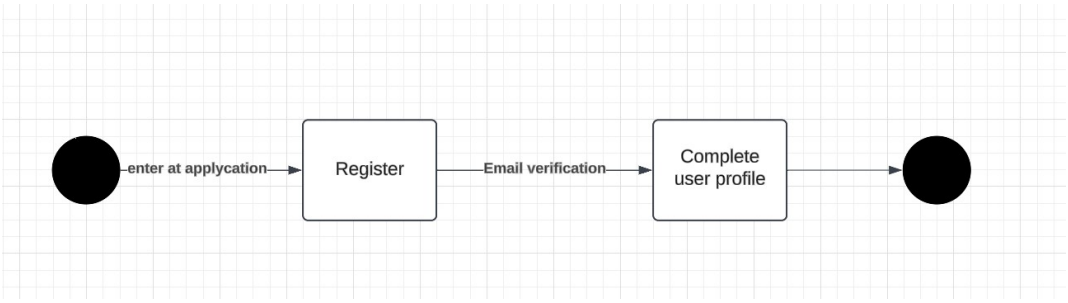


Figura 4: state diagram: Registration

The company or student accesses the app and starts the registration process by providing the required information as prompted by the system. After verifying their email, they complete their profile. Once completed, the user can search for or post internships.

2.2. Product functions

2.2.1. Sign up

The function is available only for students and companies that are not yet registered. To register, users must provide their email, password, and role (student or company). The system then sends a verification email to confirm the user's identity.

After verification, the system redirects the user to the appropriate profile page to complete the registration process.

- They must provide their full name, ID number, and university.
- They need to provide the company name, sector, mission, and overview.

Once registration is complete, users can update or edit their profile information as needed.

2.2.2. sign in

This function is available for all users (companies, students, and universities). It allows all users to access the platform.

To log in, users must provide their email and password. The system verifies the credentials and redirects the user to their respective profile page.

2.2.3. Internship Posting

Companies are the only users authorized to create and post internship offers. To publish an internship, the company must provide the following details:

- Internship description
- Project domains
- Required skills
- Tasks to be performed
- Benefits offered

- Meeting schedules
- Questionnaires for applicants

Companies are also allowed to edit, delete, or create additional internship postings as needed.

2.2.4. Internship Search and Recommendation

Students can actively search for internships using keywords and filters. This search data is utilized for future recommendations, along with the student's profile and preferences.

Internships displayed to students include the following details:

- Internship description
- Company details
- Project domains
- Required skills
- Tasks to be performed
- Benefits offered
- Questionnaires

2.2.5. Candidate search and recommendation

Companies can actively search for applications using keywords and filters. This search data is utilized for future recommendations that match the requirements of the internship.

2.2.6. Internship Application

Students are the only users who can apply for internships. To apply, they must complete the required questionnaires and submit their applications through the platform.

The system notifies the student when the company responds to their application.

2.2.7. Application Review and Response

The system notifies the company of new applications for the internship and provides tools to review the applications, questionnaires, profiles, and resumes.

The company has two options:

- Accept application: In this case, the system notifies the student of their selection and asks them to choose a meeting schedule provided by the company.
- Reject application: In this case, the system notifies the student of their status.

2.2.8. Issue Reporting

This function involves interaction between all three users (company, student, and university). There are two possible cases:

- Company Reports:
  Companies can report issues with students during their internship. The system notifies both the student and their university about the problem.
- Student Reports:
  Students can report internship-related issues. The system notifies both the university and the company where the student is doing the internship.

The university oversees these issues and can intervene and take action if deemed necessary.

2.2.9. Feedback Collection

The system requests feedback from both the student and the company after the internship ends.

- Student Feedback: Feedback on the student's experience during the internship and the application's performance.
- Company feedback:
  Feedback on the student's performance and the usability of the application.

Once the feedback is submitted, it is sent to the respective other users for review.

2.2.10. Notifications (student)

The system sends alerts to the student for the following:

- Internship recommendations
- Application statuses
- Issue reports and updates
- Feedback

2.2.11. Notifications (company)

The system sends alerts to the company for the following:

- Candidate recommendations
- New application
- Issue reports and updates
- Feedback

2.2.12. Notifications (University)

The system sends alerts to the University for the following:

- Issue reports and updates
- Feedback

2.2.13. Profile Management

Both the student and the company can edit and update their profiles. This includes updating information, uploading documents, and correcting existing details.

2.3. User characteristics

The platform has three types of users who interact with it: students, universities, and companies.

2.3.1. Students:

Students can register and log in to the platform, where they can update their profiles with information such as CVs, skills, experiences, attitudes, and future goals. They can also search for internships and receive personalized recommendations.
Students can apply for internships directly through the platform. If accepted by a company, they receive meeting details and a company questionnaire that they must complete.
During the internship, students can track their progress and communicate with their companies or universities to report any issues. Upon completing the internship, they can provide feedback on their experience with the program and the platform.

2.3.2. Companies:

Companies can register and log in to the platform. They can post internship offers with detailed information, including descriptions, project domains, required skills, tasks, and benefits (e.g., training, mentorship, stipends). These offers may also include a questionnaire and a meeting date for students who pass the initial screening.
Additionally, companies can search for potential candidates manually or receive recommendations that match their criteria. If they find a suitable candidate, they can send an internship opportunity for the student to review and either accept or reject.
During the internship, companies can update the student's status in the system or report issues to the university or the student directly.
At the end of the internship, companies provide feedback on the student's performance and share their experience with the platform. They can also receive feedback from students about their internship experience.

2.3.3. Universities:

Universities can register and log in to the platform. They can monitor the progress of students during their internships and access detailed information about activities, tasks, and overall progress.
If issues arise during an internship and no easy solution is found, the university can intervene and, if necessary, cancel the internship.
After internships are completed, universities can receive feedback from both students and companies. This feedback helps improve their internal internship policies and processes.

2.4. Assumptions, dependencies and constraints

2.5. Domain assumptions

[D1] Universities are registered in the system.

[D2] Users have access to the internet.

[D3] Users have the necessary hardware (e.g., smartphone, computer) to connect to the application.

[D4] Students provide truthful data about their capacities, such as CVs, skills, experiences, attitudes, and future goals.

[D5] Companies provide truthful data about their internship requirements, including descriptions, project domains, required skills, tasks, and benefits (e.g., training, mentorship, stipends).

[D6] Companies are interested in offering internships.

[D7] Students are interested in doing internships.

[D8] Students actively search for internships using keywords.

[D9] Students maintain updated information on their profiles, such as CVs, skills, experiences, attitudes, and future goals.

Students can track their progress during internships.

[D10] Students communicate with their university or company when they encounter issues.

[D11] Students provide honest and constructive feedback about the software's performance and their internship experience.

[D12] Companies communicate with students doing internships or with their universities if issues arise regarding the student.

[D13] Companies actively search for candidates using keywords.

[D14] Companies provide honest and constructive feedback about the software's performance and the performance of students during internships.

[D15] Companies update the status of students and internships in the system.

[D16] Companies update interview dates as needed.

[D17] Universities monitor students during internships. This includes tracking information about their activities, tasks, and progress.

[D18] Universities use feedback provided by students and companies to improve internship policies.

[D19] Universities handle issues that arise between students and companies. If necessary, they terminate internships to resolve critical problems.

3. Specific Requirement

3.1. External Interfaces Requirements

3.1.1. User Interfaces

In this section the UI of the web application is presented, there are two types of user interfaces, one for “companies” and the other one for “students”. Companies have the possibility to post internship offers. Students have the possibility to upload their CVs, search and apply for internships offered by the companies. Each type needs to provide the correct credentials to have access to the platform. In case any loss of credentials, “Forgot Password” mechanism would be needed.

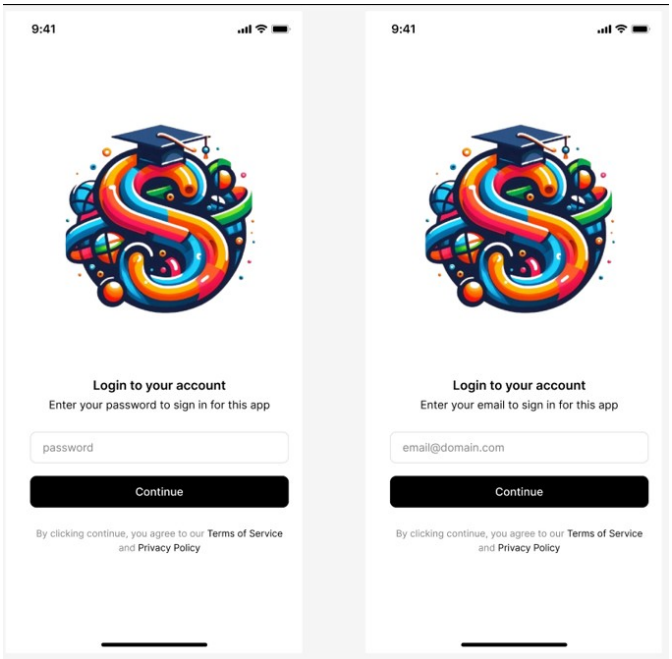


Figura 5: User Login

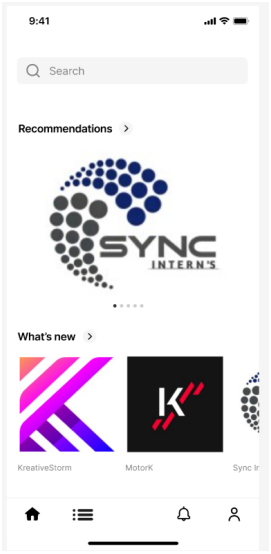


Figura 6: Student Interface.

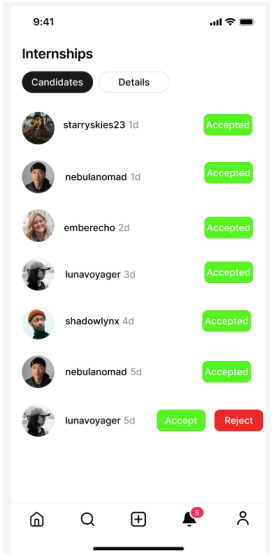


Figura 7: Accept / Reject Candidates

3.1.2. Hardware Interfaces

As a web app, the platform doesn't require any specific hardware interface rather than a computer and a device with a web browser.

3.1.3. Software Interfaces

The system needs some software interfaces to be able to do the functionality:

- Static analysis tool API: To analyze feedbacks and information collected to get statistics, in order to be able to provide suggestions to companies about project description and students about CVs.

3.1.4. Communication Interfaces

Users are going to need a good internet connection in order to have access to the platform:

- Companies need to upload internship offers.
- Students need to upload their CVs, search and apply for offers.
- Communicate with each other during the selection process.

The platform must use HTTPS to guarantee working properly and safety.

3.2. Functional Requirements

3.2.1. Sign up and Log in

[R1] The system allows "Companies" to register by providing their information (company name, company description, certifications, . . .).

[R2] The system allows "Students" to register by providing personal information (full name, academic email, password, phone number, date of birth, nationality, address, . . .).

[R3] The system allows "Students" to upload their CVs.

[R4] The system allows registered “Companies/ Students” to log in.

3.2.2. Publishing internship offers

[R5] Registered companies can post an internship offer by providing information needed (title, description, requirements, type, duration, location, number of students needed, key words).

3.2.3. Search and Apply for internship

[R6] The system allows “Students” to search for internship offers by writing a meaningful search word (job title, company name, . . .) and specifying if its paid or not.

[R7] The system allows registered “Students” to apply for an internship offer by providing all information needed from his profile(name, phone number, email, CV, . . .), and by filling up any additional requirement provided by the company.

3.2.4. Recommendations

[R8] The system regularly sends recommendations to the students based on information provided during the registration process(age, address, . . .), information in the CV(experience, skills, projects, . . .), and search keywords they use.

[R9] The system regularly sends recommendations about potential candidates based on the information, requirements, and key words provided in the offer post.

3.2.5. Acceptance and Interview scheduling

[R10] Students can accept or reject an internship offer provided by a company.

[R11] Companies can accept or reject candidates based on the company needs provided in the offer post.

[R12] Students must be notified about their request status(accepted, rejected), based on company selection.

[R13] After the acceptance process, the system sends a questionnaire to students and companies, so they will be able to schedule an interview appointment (possible dates, possible times, online/offline), to proceed with selection process.

3.2.6. View and Monitor status of internship

[R14] Each registered student has a list of all internships he is accepted in with a report of status, rates, projects, and feedbacks.

[R15] The system should have a list of all internships with all its information (company name, company profile, offer details, number of students, start date, end date, status, students rates).

[R16] Companies, and Universities should have access to view the status of each internship based on the information provided from the reports.

3.3. Use Case

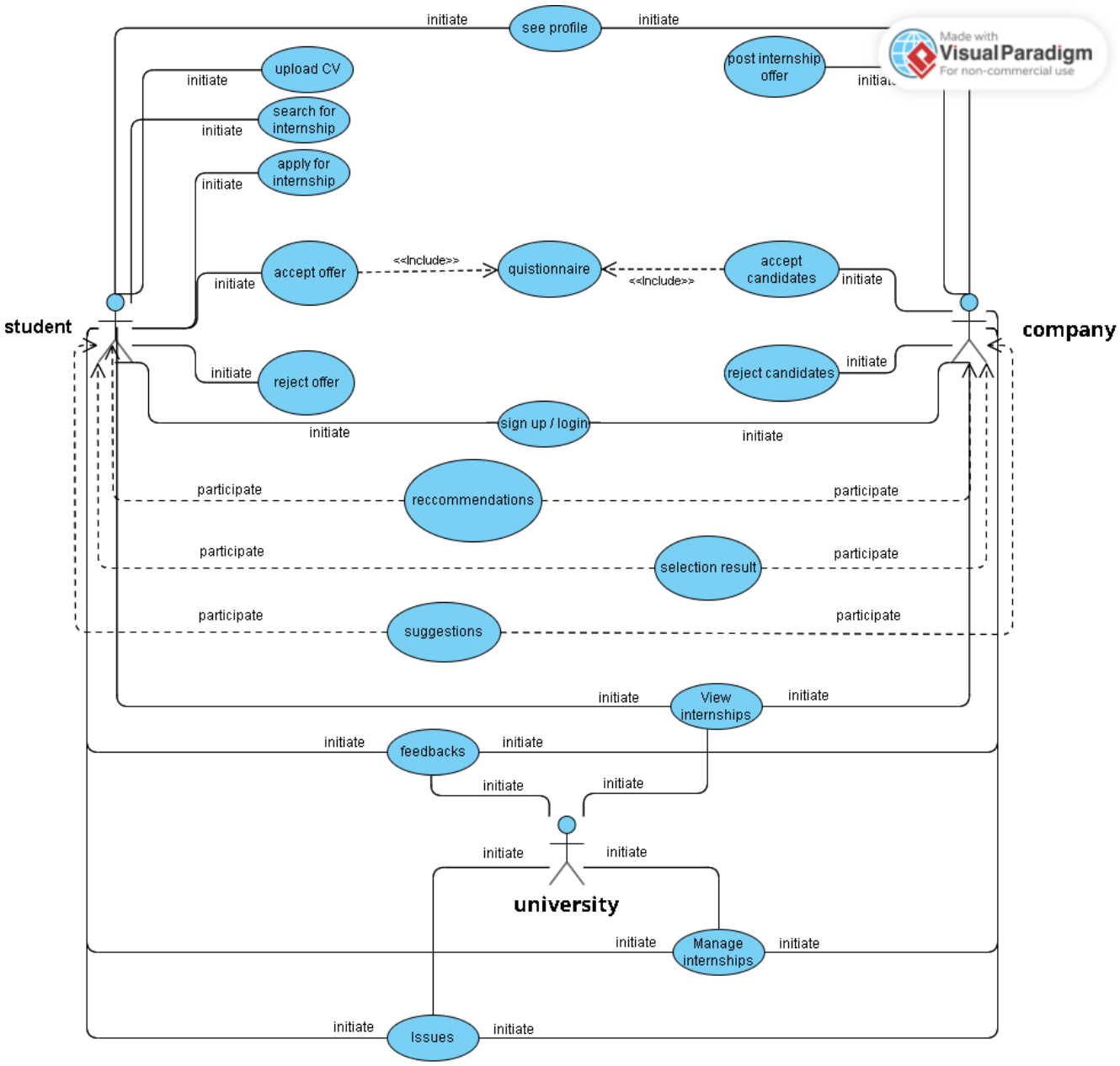


Figura 8: Use Case Diagram

3.3.1. [UC1] User Sign Up

Actors	▪ Student / Company ▪ System
Entry Condition	Student/Company enters the platform.
Event Flow	1. Student/Company presses Sign up button. 2. The system shows a form to be filled. 3. Student/Company fills the form by adding their credentials. 4. Student/Company presses “Register” to complete the registration. 5. The system redirects to the profile page.
Exit Condition	Student/Company successfully registered to the platform.
Exception	Student/Company is already registered.

3.3.2. [UC2] User Login

Actors	▪ Student / Company ▪ System
Entry Condition	Student/Company enters the platform.
Event Flow	1. Student/Company presses “Sign in” button. 2. The system shows a form to be filled. 3. Student/Company fills the form by adding their credentials. 4. Student/Company presses “Sign in” to complete the login. 5. The system redirects to the profile page.
Exit Condition	Student/Company successfully accessed the platform.

3.3.3. [UC3] Post Internship Offer

Actors	▪ Company ▪ System
Entry Condition	Company pressed the “Post” button.
Event Flow	1. Company presses Post button. 2. The system shows a form to be filled with all offer details. 3. Company fills all form fields. 4. Company presses “Post” to submit the form.
Exit Condition	Company successfully posted the internship offer.
Exception	Company didn’t fill all the fields in the form or filled with invalid information.

3.3.4. [UC4] Search for Internship

Actors	▪ Student ▪ System
Entry Condition	Student enters the platform.
Event Flow	1. Student writes meaningful search words in the search bar. 2. Student presses on the “Search” button. 3. The system shows results based on information provided.
Exit Condition	System shows the search results.
Exception	Student is not registered in the system.

3.3.5. [UC5] Apply for Internship

Actors	▪ Student ▪ System
Entry Condition	Student presses “Apply” button.
Event Flow	1. System shows a form with necessary fields of information to be filled. 2. System automatically fills some fields based on information presented in the student profile and CV. 3. Student fills any other unfilled necessary field. 4. Student presses the “Submit” button to complete the application.
Exit Condition	System successfully sends the application and redirects the student to the searching page.
Exception	Student didn’t fill all the necessary fields in the form or filled with invalid information.

3.3.6. [UC6] Accept an Internship Offer

Actors	▪ Student ▪ System
Entry Condition	Student presses the “Offer” button.
Event Flow	1. Student presses on the “Internship” button to enter the internships section. 2. Student presses on the “Offer” button inside the internships section to view the received offers. 3. System shows a list of received offers if any. 4. Student selects an offer. 5. System redirects student to the offer details page. 6. Student presses “Accept” button. 7. System saves the choice and sends an acceptance notification to the company that sent the offer. 8. System updates the offer status and removes the rejected offers from the received offers list.
Exit Condition	System successfully saves the choice, notifies the company, and redirects the student to the offers section page.
Exception	System didn’t manage to save the choice and send the notification successfully, so the system notifies the student about the unsuccessful process and redirects them to the offers section page.

3.3.7. [UC7] Reject an Internship Offer

Actors	▪ Student ▪ System
Entry Condition	Student presses the "Offer" button.
Event Flow	1. Student presses on the "Internship" button to enter the internships section. 2. Student presses on the "Offer" button inside the internships section to view the received offers. 3. System shows a list of received offers if any. 4. Student selects an offer. 5. System redirects student to the offer details page. 6. Student presses "Reject" button. 7. System saves the choice and sends a rejection notification to the company that sent the offer. 8. System updates the offer status and removes the rejected offers from the received offers list.
Exit Condition	System successfully saves the choice, notifies the company, and redirects the student to the offers section page.
Exception	System didn't manage to save the choice and send the notification successfully, so the system notifies the student about the unsuccessful process and redirects them to the offers section page.

3.3.8. [UC8] Accept Candidates

Actors	▪ Company ▪ System
Entry Condition	Company presses the “Candidates” button.
Event Flow	1. Company presses on the “Internship” button to enter the internships section. 2. System shows a list of all internship offers the company posted if any. 3. Company selects an internship offer from the list. 4. System redirects company to the internship page. 5. Company presses on the “Candidates” button to enter the candidates section. 6. System shows a list of all candidates applied for the internship if any. 7. Company presses “Accept” button for each candidate to be accepted. 8. System saves the choices and sends an acceptance notification to the accepted candidates (students). 9. System updates the status of each candidate and removes the rejected candidates from the candidates list.
Exit Condition	System successfully saves the choice, notifies the accepted candidates (students), and redirects the company to the same page after the update.
Exception	System didn’t manage to save the choices and send the notifications successfully, so the system notifies the company about the unsuccessful process and redirects to the internship section page.

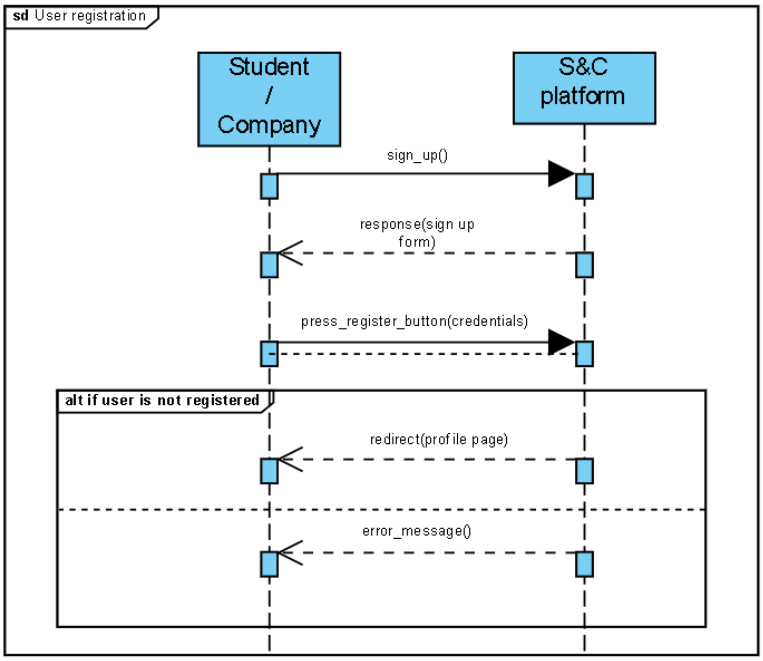
3.3.9. [UC9] Reject Candidates

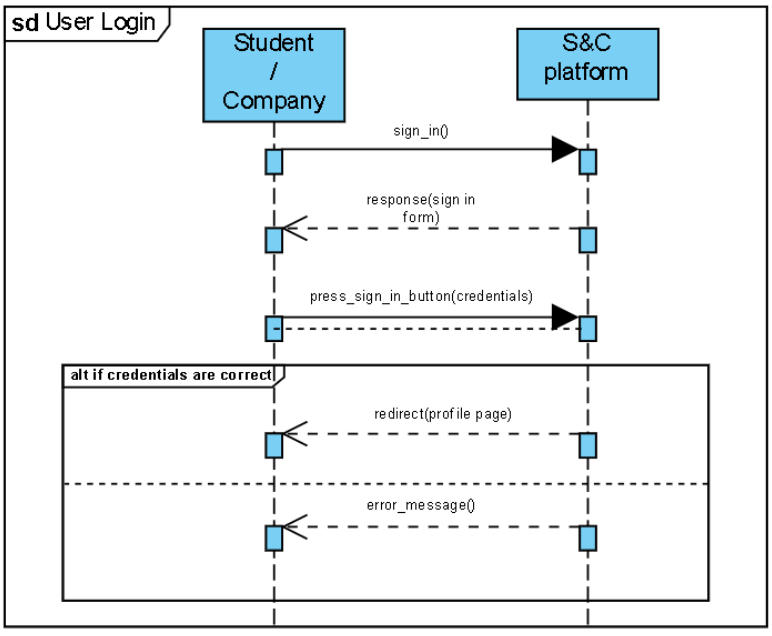
Actors	▪ Company ▪ System
Entry Condition	Company presses the “Candidates” button.
Event Flow	1. Company presses on the “Internship” button to enter the internships section. 2. System shows a list of all internship offers the company posted if any. 3. Company selects an internship offer from the list. 4. System redirects company to the internship page. 5. Company presses on the “Candidates” button to enter the candidates section. 6. System shows a list of all candidates applied for the internship if any. 7. Company presses “Reject” button for each candidate to be rejected. 8. System saves the choices and sends a rejection notification to the rejected candidates (students). 9. System updates the status of candidates and removes the rejected ones from the candidates list.
Exit Condition	System successfully saves the choice, notifies the rejected candidates (students), and redirects the company to the same page after the removal of the rejected candidates.
Exception	System didn’t manage to save the choices and send the notifications successfully, so the system notifies the company about the unsuccessful process and redirects to the internship section page.

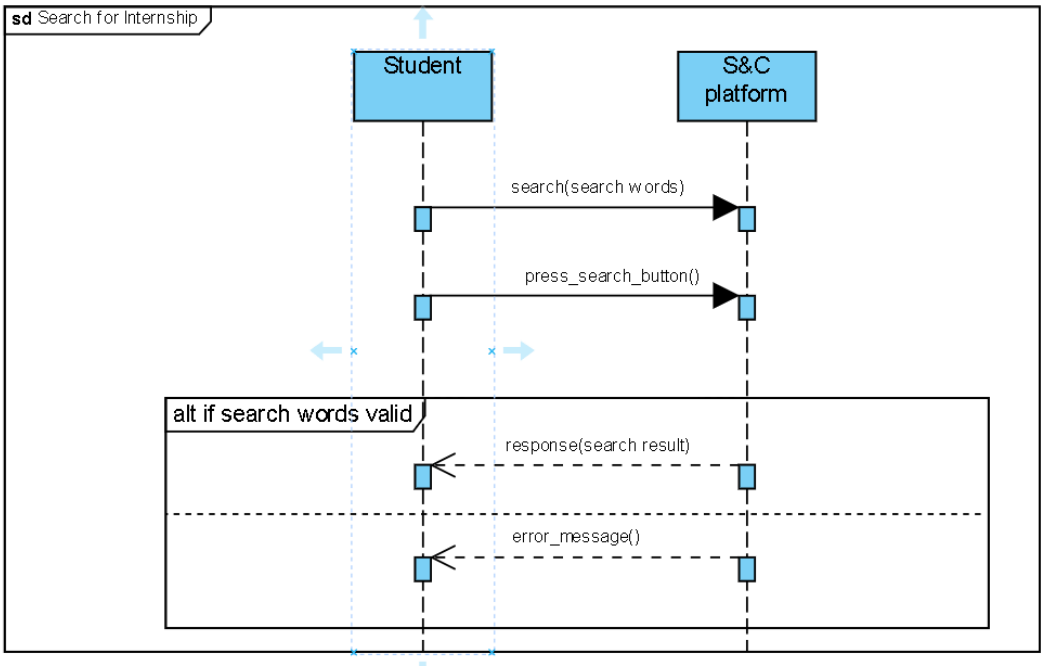
3.3.10. [UC10] Send Selection Results

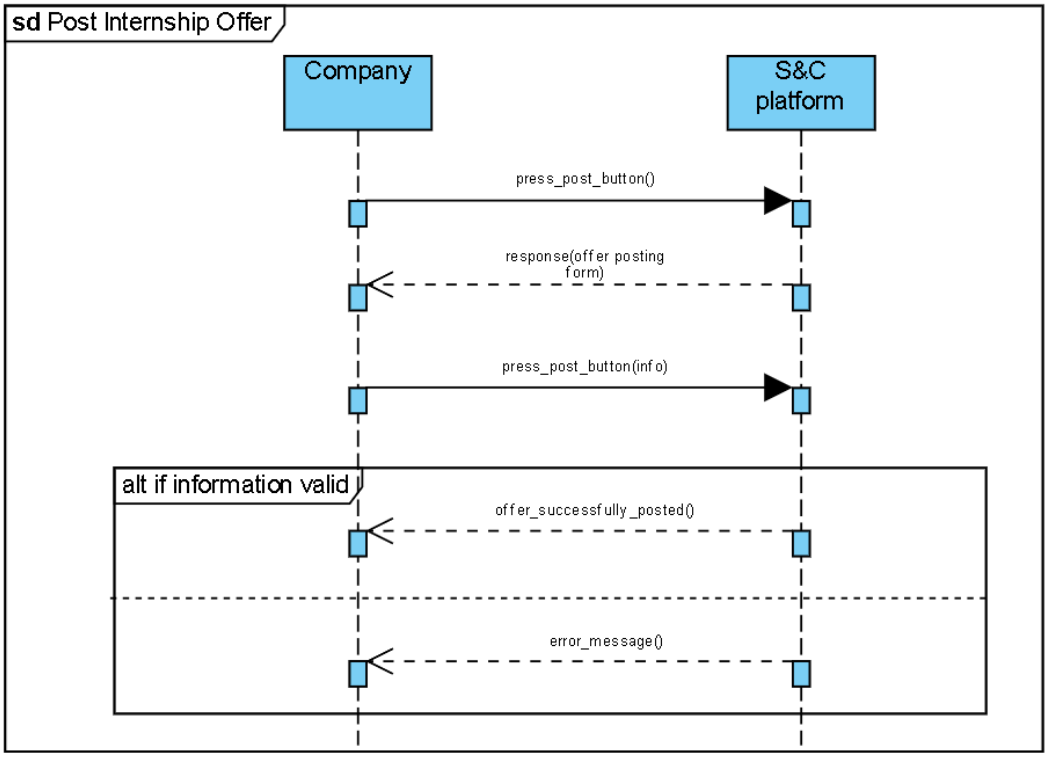
Actors	▪ Company ▪ System
Entry Condition	Company presses the “Candidates” button.
Event Flow	1. Company presses on the “Internship” button to enter the internships section after the end of the interviews. 2. System shows a list of all internship offers the company posted if any. 3. Company selects an internship offer from the list. 4. System redirects company to the internship page. 5. Company presses on the “Candidates” button to enter the candidates section. 6. System shows a list of all candidates applied for the internship if any. 7. Company presses “Reject” button for each candidate to be rejected. 8. System saves the choices and sends a notification (rejection, acceptance) to the candidates (students). 9. System updates the status of each candidate and removes the rejected candidates from the candidates list.
Exit Condition	System successfully saves the choice, notifies the candidates (students), and redirects the company to the same page after the update.
Exception	System didn’t manage to save the choices and send the notifications successfully, so the system notifies the company about the unsuccessful process and redirects to the internship section page.

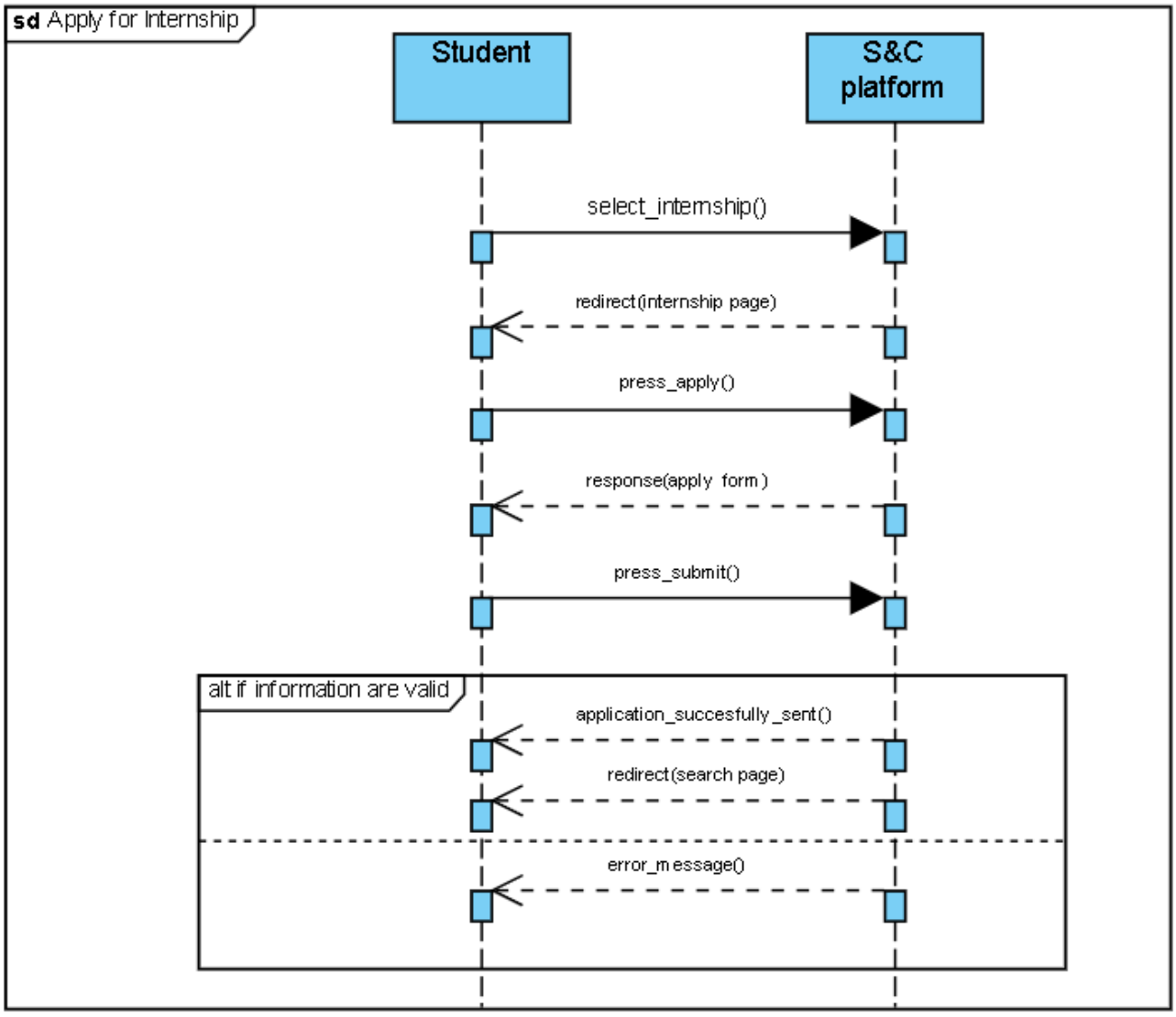
3.4. Sequence Diagrams

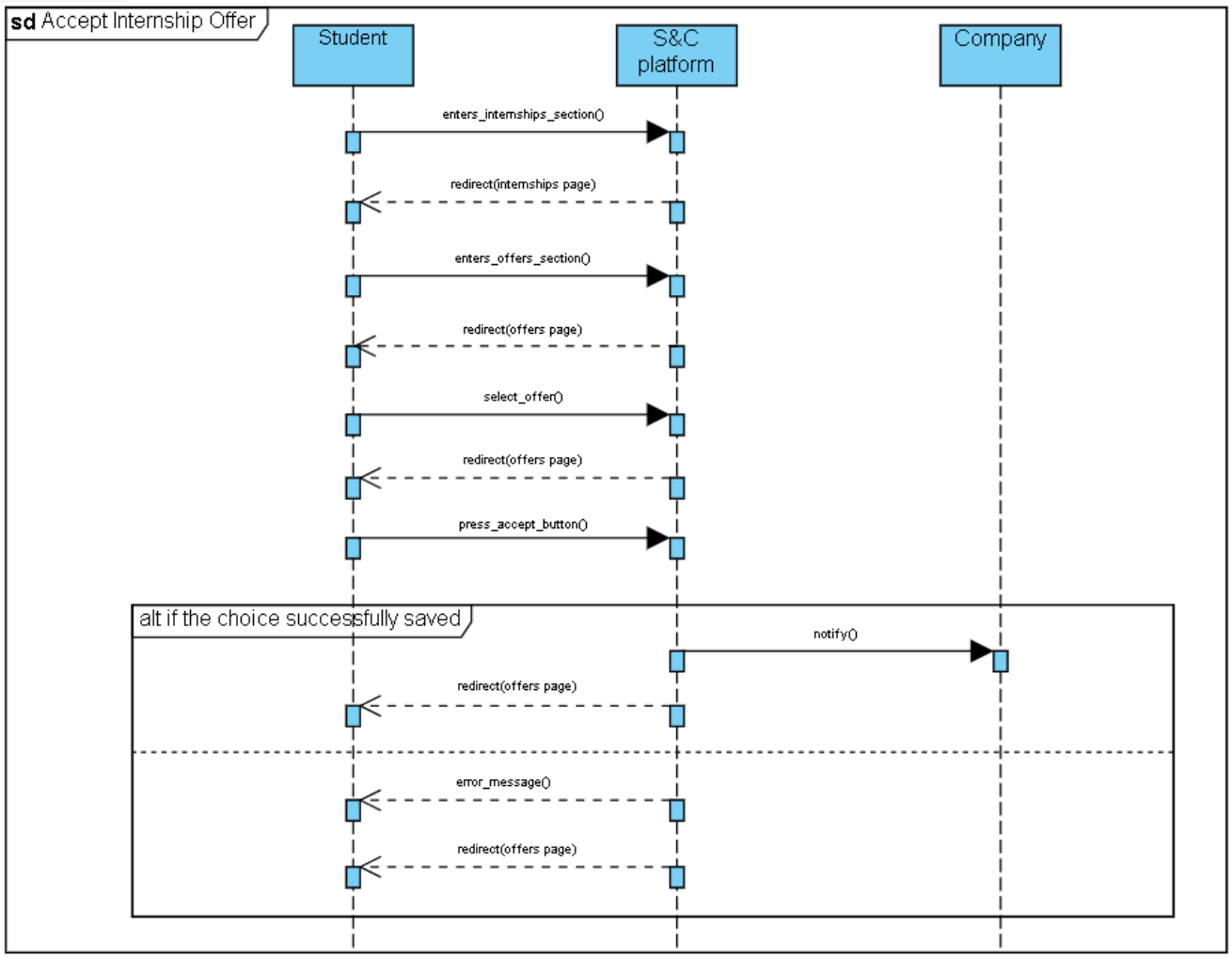


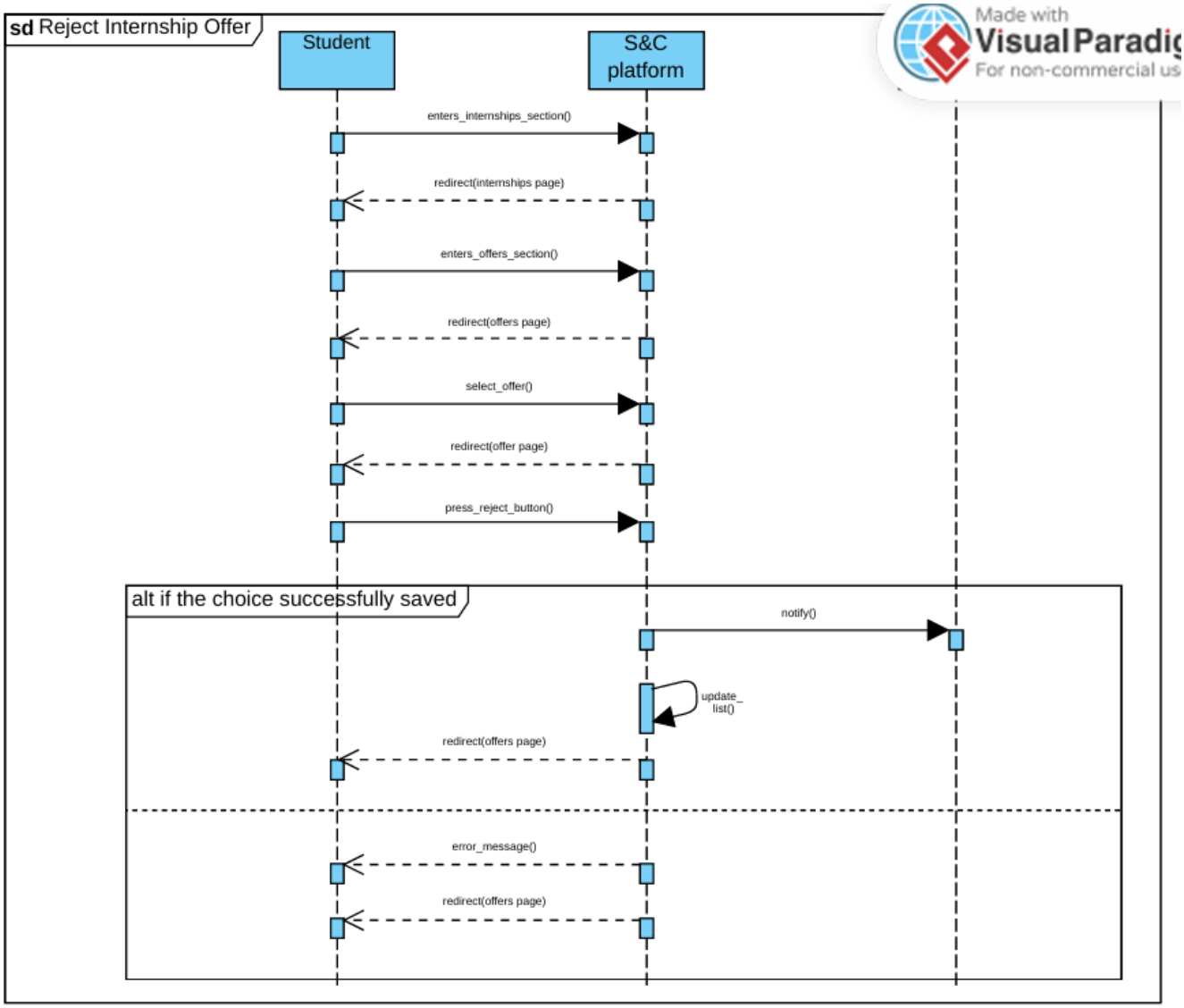


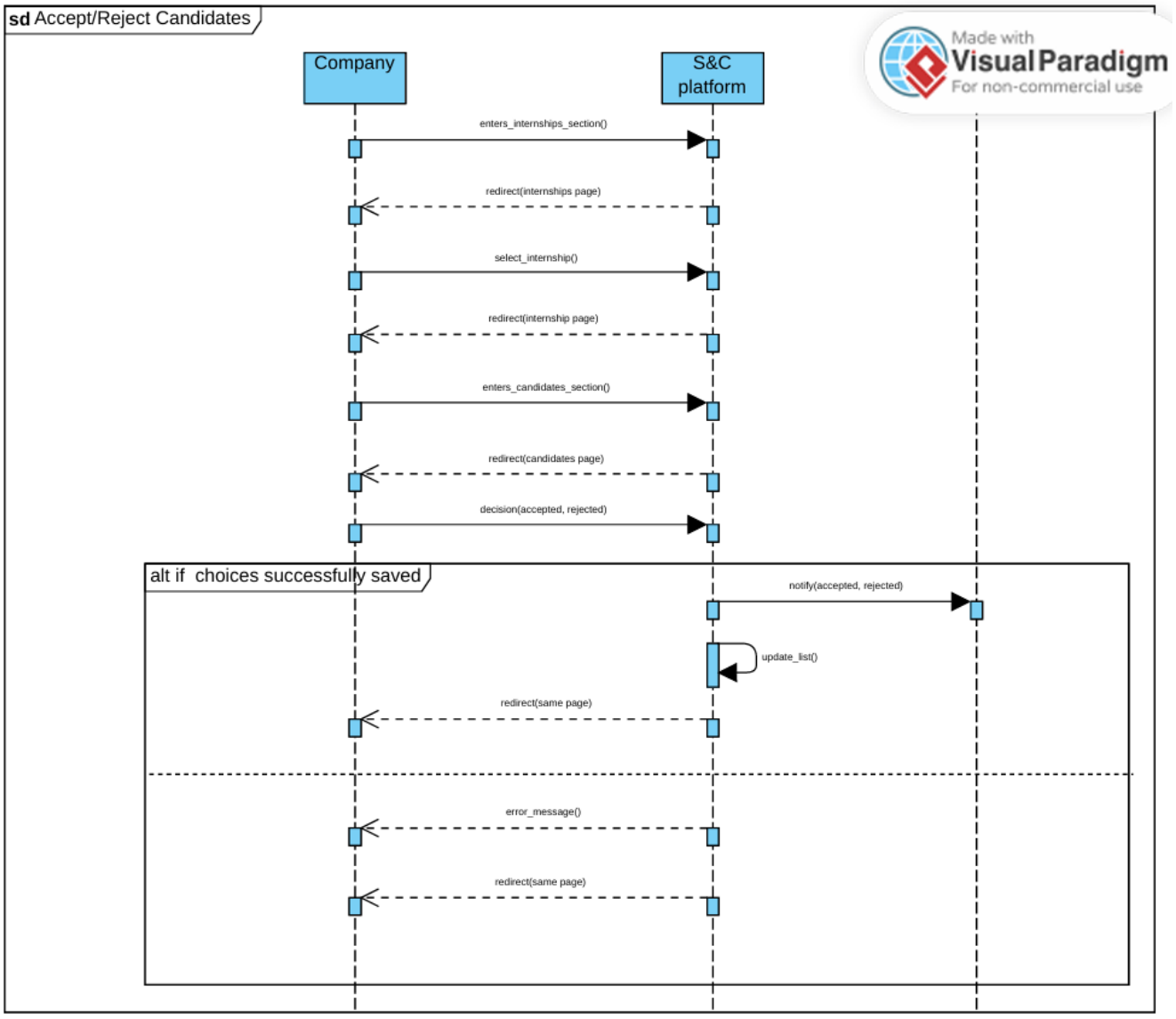

sd Search for Internship
Student
S&C platform
search(search words)
press_search_button()
alt if search words valid
response(search result)
error_message()


sd Post Internship Offer
Company
S&C platform
press_post_button()
response(offer posting form)
press_post_button(info)
alt if information valid
offer_successfully_posted()
error_message()


sd Apply for Internship
Student
S&C platform
select_internship()
redirect(internship page)
press_apply()
response(apply form)
press_submit()
alt if information are valid
application_succesfully_sent()
redirect(search page)
error_message()


sd Accept Internship Offer
Student
S&C platform
Company
enters_internships_section()
redirect(internships page)
enters_offers_section()
redirect(offers page)
select_offer()
redirect(offers page)
press_accept_button()
alt if the choice successfully saved
notify()
redirect(offers page)
error_message()
redirect(offers page)


sd Reject Internship Offer
Student
S&C platform
enters_internships_section()
redirect(internships page)
enters_offers_section()
redirect(offers page)
select_offer()
redirect(offer page)
press_reject_button()
alt if the choice successfully saved
notify()
update_ list()
redirect(offers page)
error_message()
redirect(offers page)


sd Accept/Reject Candidates
Company
S&C platform
enters_internships_section()
redirect(internships page)
select_internship()
redirect(internship page)
enters_candidates_section()
redirect(candidates page)
decision(accepted, rejected)
alt if choices successfully saved
notify(accepted, rejected)
update_list()
redirect(same page)
error_message()
redirect(same page)
Made with VisualParadigm
For non-commercial use

3.5. Design Constraints

3.5.1. Hardware Limitations

Here is presented a summary of the hardware features that user should have to use the platform properly:

- The user must have a device with good internet connection (should be compatible with at least one of these standards 3G, 4G, 5G, IEEE 802.11 and IEEE 802.3). Both for the wired and wireless communications it must be connected to a device able to guarantee an internet connection such as a modem or an access point and so on.
- The user can have a device with normal hardware features such as (smartphone, laptop, tablet).

3.5.2. Software System Attributes

Here we will explain some software attributes that the system should provide.

3.5.3. Reliability

The system must be reliable because it will have to run continuously for a long period of time. To ensure this feature the platform must have some sort of replication and consistency policy to avoid system crash. Moreover, as best practice, it is important to have offline backups of the system for recovering information after data loss.

3.5.4. Availability

This is the most important attribute the system must provide. The system should be available most of the time. Some replication policies must be implemented, it must be prepared for a possibly large number of submissions when a new offer is posted.

3.5.5. Security

The system will store the user's personal data so the security aspect must be carefully considered.

- Passwords stored in the central database must be encrypted.
- Stored data must be protected with all possible security measures to avoid internal and external attacks.
- The platform must ensure integrity, consistency and confidentiality by using appropriate cyber-risk avoidance policies.

3.5.6. Maintainability

The system must guarantee a good level of maintainability. The code must be well documented. A testing routine must be provided, and it has to cover at least 75 % of the code.

3.5.7. Portability

The system is a web application so it must be compatible with different web browsers such as (Firefox, Google Chrome, ...), and devices such as(smartphones, computers, ...).

4. Formal Analysis Using ALLOY

4.1. Code

```
abstract sig Agent {}

sig Student extends Agent{
    cv: one CV,                     // Associated CV
     interest : some Interest  //Interest field
}

sig CV {
    skills: set Skill,              // Skills possessed by the student
    experience: set Experience,     // Experiences of the student
    attitude: set Attitude,         // Attitudes/traits of the student
    futuregoals: set FutureGoals  // goals planned to achieve in the future

}

sig Company extends Agent {
    internships: set Internship     // Internships offered by the company
}

sig Internship {
    project: one Project,           // Details of the project
        (tasks, technologies, etc.)
    terms: one Terms,               // Terms offered (paid, mentorship, etc.)
    field : some Interest // internship field
}

abstract sig Status {}
one sig ACCEPTED, REJECTED, PENDING extends Status {}

sig Recommendation {
    student: one Student,           // Recommended student
    internship: one Internship,     // Recommended internship
   status: one Status,              // Status of the recommendation
        (e.g., accepted, rejected)
application: lone Application, //Selection process
feedback : set Feedback //Feedback
}
{status != ACCEPTED implies #(application)=0
    && status = ACCEPTED implies #(application)=1 }
//if status is accepted then there is a selection process
    otherwise there is no a selection process


sig University {
    students: set Student,          // Students belonging to this university
```

```
}

sig Issue {
    agent: one Agent,          // Student or company raising the complaint
    internship: one Internship,    // Internship associated with the complaint
    description: some String       // Complaint description
}

sig Interview {
    time: one Time,          // Time of interview
    place: one Place,    // physical place or virtual meeting link
}

sig Application {
    student: one Student,          // Student of selection process
internship: one Internship,    // Internship of selection process
status: one Status,          // Application status
    interview: set Interview,    // Interview of selection process
    questionnaires: lone Questionnaires,    // Questionnaires
}



sig Interest {}
sig Skill {}
sig Experience {}
sig Attitude {}
sig FutureGoals {}
sig Project {}
sig Terms {}
sig Time {}
sig Place {}
sig Questionnaires  {}
sig Feedback{}


// Constraints and Facts

fact StudentCV  {
    //each student has one and only one CV
    all cv1 : CV | (one s: Student| cv1= s.cv)
}
fact StuUni  {
    //each student has one and only one CV
    all s : Student | (one u: University| s in u.students)
}

fact CompanyInternship  {
    //each student has one and only one CV
```

```
    all i : Internship | (one c: Company| i in c.internships)
}

fact IssueAgent{
all c : Issue |( (one com :Company| c.agent = com
    && c.internship in com.internships )
|| (one s : Student | c.agent = s && (one a: Application| a.student = s
    && a.internship = c.internship && a.status = ACCEPTED)))
}

fact ValidStuIntRacc {
    // Recommendations must link valid students and internships
    all r: Recommendation |
        r.student in Student and r.internship in Internship
}

fact ValidStuIntApp {
    // Selection process must link valid students and internships
    all r: Application |
        r.student in Student and r.internship in Internship
}

fact StudentRacSelPro  {
    all r : Recommendation | (r.status = ACCEPTED implies
        (one a: Application | r.student= a.student))
}
//When the racommendation is accepted the student linked by reccondation must
    the same of the student linked by selection process

fact DifRecommendation {
    all r1, r2 : Recommendation | r1!=r2 implies (r1.student != r2.student) ||
        (r1.internship != r2.internship)}

// Given two different raccomandation, the student or the internship
    must be different


fact DifSelPro {
all r1, r2 : Application | r1!=r2 implies (r1.student != r2.student) ||
    (r1.internship != r2.internship)}
// Given two different selection process, the student or the internship
    must be different

fact SelProIntern  {
all i : Internship | #{a : Application | a.internship =i
    && a.status = ACCEPTED} <= 1
}
//there are more than 1 selection process linked to the internship,
    but only one application can be accepted
```

```
fact DifAppDiffInterQues {
all r1, r2 : Application | r1!=r2 implies (r1.interview != r2.interview)
    && (r1. questionnaires!= r2. questionnaires)}
// Given two different application, the interview and questionnaires
    must be different

fact InteApp {
all i : Interview |(one a : Application | i in a.interview)
}
// An interview belongs on to one application

fact AppRecc {
all a : Application |(lone r : Recommendation | a in r.application)
}
// An application belongs on to zero or one recommendation

fact QuesApp {
all i : Questionnaires |(one a : Application | i in a.questionnaires)
}
// An questionaires belongs on to one application

fact SkillCV {
all i : Skill |(some cv : CV | i in cv.skills)
}
// Skill must belongs on to some CV


fact ExpCV {
all i : Experience |(one cv : CV | i in cv. experience)
}
// Experience must belongs on to one CV


fact AttCV {
all i : Attitude |(some cv : CV | i in cv. attitude)
}
// Attitudes must belongs on to some CV

fact FGCV {
all i : FutureGoals  |(some cv : CV | i in cv. futuregoals)
}
// Futuregoals must belongs on to some CV

fact ProInt {
all p : Project  |(some i : Internship | p = i.project)
}
// Project must belongs on to some internship
```

```
fact TermInt {
all p : Terms  |(some i : Internship | p = i.terms)
}
// Term must belongs on to some internship

fact TimInt {
all p : Time  |(some i : Interview | p = i.time)
}
// Time must belongs on to some interview

fact PlaInt {
all p : Place  |(some i : Interview | p = i.place)
}
// Place must belongs on to some interview

fact FeedRec {
all p : Feedback  |(one i : Recommendation | p = i.feedback)
}
// Feedback must belongs on to one Recommendation


pred AcceptedAppStu(s ,s': Application) {
//precondition
s.status = PENDING
//postcondition
        s'.status = ACCEPTED && s.student = s'.student
            && s.internship=s'.internship && s.interview=s'.interview
            && s.questionnaires = s'.questionnaires
}
//Application accepted

pred RejectedAppStu(s ,s': Application) {
//precondition
s.status = PENDING
//postcondition
        s'.status = REJECTED && s.student = s'.student
            && s.internship=s'.internship && s.interview=s'.interview
            && s.questionnaires = s'.questionnaires
}
//Application rejected

pred AddInterview(s ,s': Application, i :Interview) {
//precondition
s.status = PENDING
not i in s.interview
//postcondition
s'.interview = s.interview + i && s.student = s'.student
    && s.internship=s'.internship && s.status=s'.status
    && s.questionnaires = s'.questionnaires
```

```
}
//Add interview in application

pred AddQuest(s ,s': Application, i :Questionnaires) {
//precondition
s.status = PENDING
not i in s.questionnaires
//postcondition
s'. questionnaires = s. questionnaires + i && s.student = s'.student
    && s.internship=s'.internship && s'.status=ACCEPTED
    && s.interview = s'.interview
}
//Add questionnaires in application

pred AcceptedRecStu(s ,s': Recommendation) {
//precondition
s.status = PENDING
//postcondition
        s'.status = ACCEPTED && s.student = s'.student
            && s.internship=s'.internship && s.application =s'.application
            && s.feedback = s'.feedback
}
//Recommendation accepted

pred RejectedAppStu(s ,s': Application) {
//precondition
s.status = PENDING
//postcondition
        s'.status = REJECTED && s.student = s'.student

            && s.internship=s'.internship && s.interview= s'.interview
            && s.questionnaires=s'.questionnaires
}
//Recommendation rejected

pred AddFeed(s ,s': Recommendation, f :Feedback) {
//precondition
s.status = PENDING
not f in s.feedback
//postcondition
        s'.feedback = s.feedback + f && s.student = s'.student
            && s.internship=s'.internship && s.application =s'. application
            && s.status != REJECTED
}
//Add feedback in recommendation

pred World {
#Student=3
#Application=3
```

```
#Recommendation=3
}
run World for 5
```

4.1.1. Examples

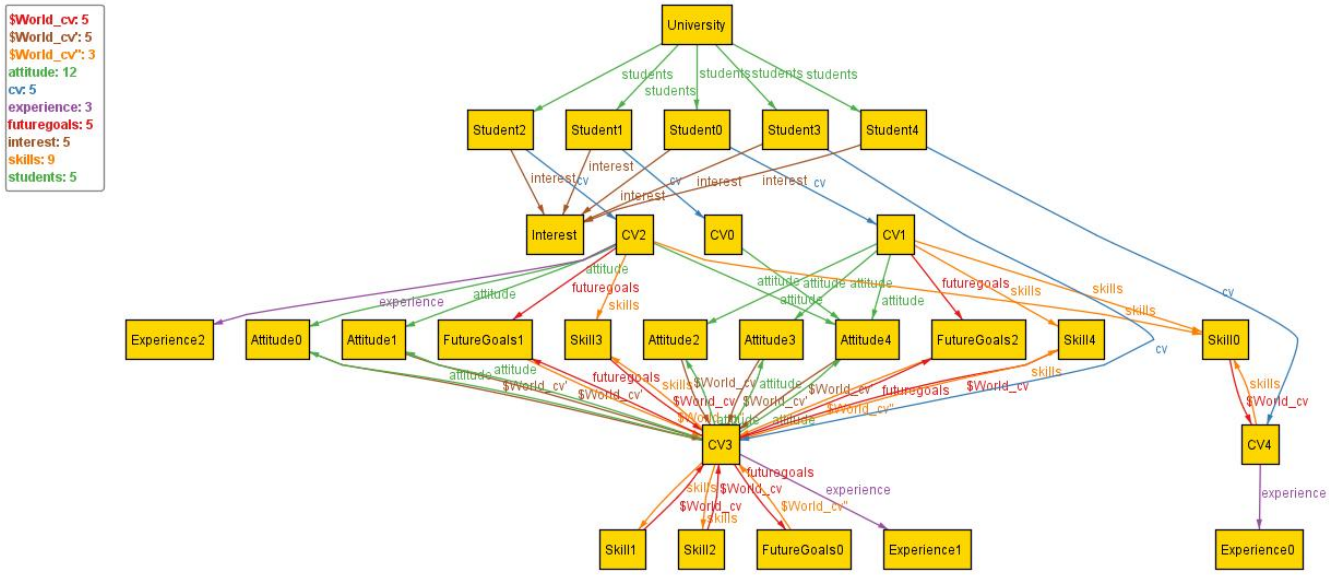


Figura 9: Caption

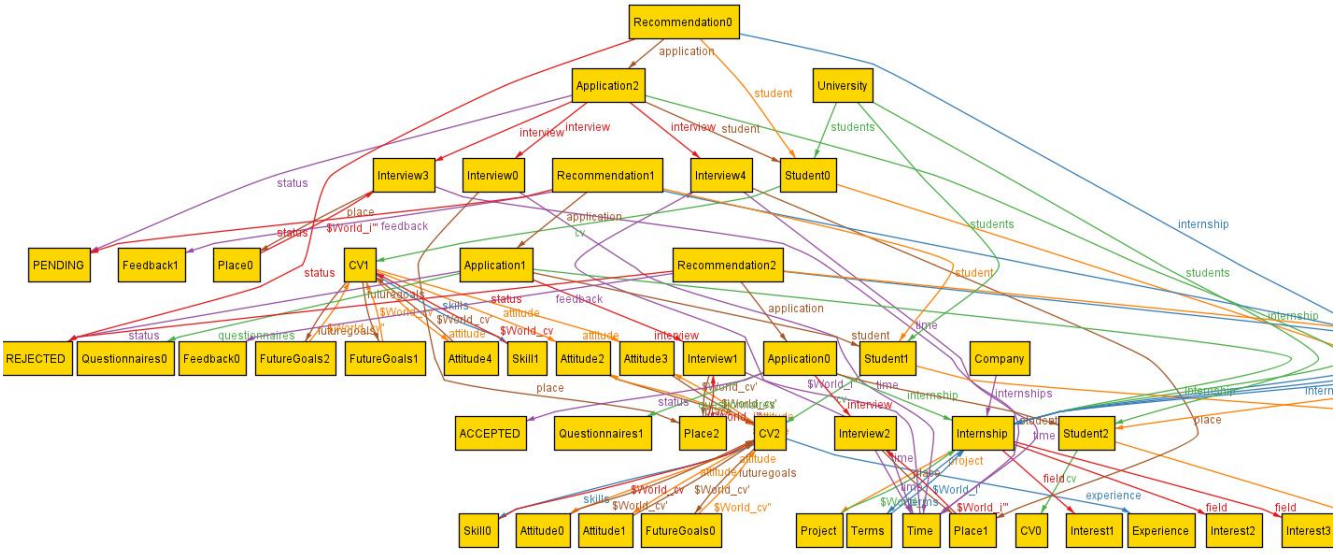


Figura 10: Caption

5. Effort Spent

- **Guillermo:** 30h
- **Mohamed:** 30h
- **Yichen:** 30h

6. References

[1] Matteo Giovanni Rossi, Software engieenering 2. Politecnico di Milano, lessons 2024.