



**POLITECNICO**  
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

# Software Engineering 2

## Requirements Analysis and

## Specification Document

Author(s): De Matteis Alessandro - 10845281

Marino Margherita - 10795242

Monaco Giorgio - 10775329

Academic Year: 2024-2025

Version: 1.0

Release date: 21/12/2024



# Contents

<b>Contents</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Purpose . . . . .	1
1.1.1 Goals . . . . .	1
1.2 Scope . . . . .	2
1.2.1 World Phenomena . . . . .	2
1.2.2 Machine Phenomena . . . . .	3
1.2.3 Shared phenomena . . . . .	3
1.3 Definitions, Acronyms, Abbreviations . . . . .	5
1.3.1 Definitions . . . . .	5
1.3.2 Acronyms . . . . .	5
1.3.3 Abbreviations . . . . .	5
1.4 Reference Documents . . . . .	6
1.5 Document Structure . . . . .	6
<b>2 Overall Description</b>	<b>7</b>
2.1 Product perspective . . . . .	7
2.1.1 Scenarios . . . . .	7
2.1.2 Class diagrams . . . . .	10
2.1.3 State diagrams . . . . .	11
2.2 Product functions . . . . .	17
2.3 User characteristic . . . . .	18
2.4 Assumptions, dependencies and constraints . . . . .	19
<b>3 Specific Requirements</b>	<b>21</b>
3.1 External Interface Requirement . . . . .	21
3.1.1 User interfaces . . . . .	21

3.1.2	Hardware interfaces . . . . .	24
3.1.3	Software interfaces . . . . .	24
3.1.4	Communication interfaces . . . . .	25
3.2	Functional requirements . . . . .	25
3.2.1	Requirements . . . . .	25
3.2.2	Use case diagrams . . . . .	28
3.2.3	Use cases . . . . .	29
3.2.4	Mapping on goals . . . . .	57
3.3	Performance requirements . . . . .	62
3.4	Design constraints . . . . .	63
3.4.1	Standard compliance . . . . .	63
3.4.2	Hardware limitations . . . . .	63
3.5	Software system attributes . . . . .	63
3.5.1	Reliability . . . . .	63
3.5.2	Availability . . . . .	63
3.5.3	Security . . . . .	63
3.5.4	Maintainability . . . . .	64
3.5.5	Portability . . . . .	64
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>65</b>
<b>5</b>	<b>Effort Spent</b>	<b>79</b>
<b>6</b>	<b>References</b>	<b>81</b>
6.1	References . . . . .	81
6.2	Used Tools . . . . .	81
<b>List of Figures</b>		<b>83</b>
<b>List of Tables</b>		<b>85</b>

# 1 | Introduction

## 1.1. Purpose

As the demand for skilled graduates grows, bridging the gap between academic learning and practical industry experience has become essential. However, students often face challenges in finding internships that align with their skills and career goals, while companies struggle to identify candidates who truly match their specific needs. Traditional job boards often fail to meet the specific needs of internships, causing mismatches and missed opportunities for students and employers. Students and companies (S&C) exists to address this gap by providing a tailored matchmaking platform. By aligning students' skills, experiences, and aspirations with detailed internship roles, S&C ensures better fit and smoother transitions into the workforce. The platform supports proactive searches, personalized recommendations, and structured selection processes, ultimately helping students gain hands-on experience while allowing companies to find the right talent for their projects.

### 1.1.1. Goals

**[G1]:** University students would like to find internships that better align with their interests and field of study.

**[G2]:** Companies would like to offer internships to students who best match the related figure.

**[G3]:** Improve communication and coordination between students and companies in the selection process.

**[G4]:** Evolve the recommendation system through feedback and suggestions (provided by students and companies) to improve future matches.

**[G5]:** Universities would like to monitor internships and handle any emerging issues promptly, ensuring a smooth and beneficial experience for all parties involved.

## 1.2. Scope

In this section, we are going to identify the S&C platform domain. There are three main users that interact with the system:

- **Students** – University students seeking internships as part of their academic or career development. They can use S&C to proactively browse available internships, receive customized recommendations based on their profiles, and interact with companies to begin the application and selection process.
- **Companies** – Businesses and organizations that offer internships. These companies use S&C to advertise internship roles with detailed information on project tasks, required skills, technologies, and benefits. S&C enables companies to receive suitable student profiles automatically, simplifying their candidate search and supporting the structured interview and selection process.
- **Universities** – Academic institutions that need to monitor and oversee their students' internship experiences. Universities use S&C to track the status and progress of internships, address student concerns, and intervene in issues that may impact the quality of the internship.

According to the World and Machine paradigm we can identify the Machine as the System to be developed and the environment in which S&C will be used as the World. This separation allows us to classify the entire phenomena in three different types.

### 1.2.1. World Phenomena

Events that take place in the real world and that the machine cannot observe:

**[WP1]:** The company creates and defines internship opportunities independently, describing tasks, required skills, and terms.

**[WP2]:** The student creates CVs detailing his skills and attitudes.

**[WP3]:** The company conducts an interview with the student.

**[WP4]:** The student wants to accept/reject the proposal.

**[WP5]:** The student or the company encounters problems during the internship and wants to report it.

**[WP6]:** The university handles the complaint submitted by his student or a related company.

### 1.2.2. Machine Phenomena

Events that take place inside the System and cannot be observed by the real world.

[MP1]: The internal processes analyzes student profiles and internship postings to generate recommendations.

[MP2]: The recommendation system automatically improves using feedback and suggestions.

### 1.2.3. Shared phenomena

- World controlled

Controlled by the world and observed by the machine.

[SP1]: The Guest signs up to the system.

[SP2]: The User logs in to the system.

[SP3]: The C uploads internship terms and details which the system then collects.

[SP4]: The S improves/edits his CV based on the suggestions.

[SP5]: The C improves/edits his internship insertion based on the suggestions.

[SP6]: The S looks for an internship on the platform, by querying a company's name.

[SP7]: The S reviews a C's profile.

[SP8]: The S reviews his calendar.

[SP9]: The S surf through the recommended internships.

[SP10]: The S reviews an internship insertion.

[SP11]: The S or the C submits feedback and suggestions about the matchmaking process.

[SP12]: The S applies for an internship.

[SP13]: The C reviews a candidate's profile.

[SP14]: The C sends a contact request in order to offer an internship to a recommended student.

[SP15]: The S accepts/refuses a contact request by a company.

[SP16]: The C and the S schedule an interview through the chat.

**[SP17]:** The S and the C finalize the selection process after the interview.

**[SP18]:** The C offers an internship to the student after the interview.

**[SP19]:** The S accepts/refuses the proposed internship.

**[SP20]:** The S or the C submits problems, complaints and information about the ongoing internship.

**[SP21]:** The U monitors the ongoing internship.

**[SP22]:** The U interrupts the internship.

- **Machine controlled**

Controlled by the machine and observed by the World.

**[SP23]:** The system analyzes the CV and suggests to the S how to improve it.

**[SP24]:** The system analyzes the Insertion and suggests to the C how to improve it.

**[SP25]:** The system recommends the best internships to the S based on his CV, his attitudes and previously submitted feedback.

**[SP26]:** The system suggests the best candidates to the company based on matching algorithms and keywords.

**[SP27]:** The system notifies the S when new recommended internships become available by sending an email and a notification.

**[SP28]:** The system notifies the C of potential candidates when new CVs become available by sending an email and a notification.

**[SP29]:** The system asks the S and the company to provide feedback or suggestions about the matchmaking process.

**[SP30]:** The system establishes contact between the S and the C when both parties have accepted each other's request.

**[SP31]:** The system adds to the S&C calendar the start and the end date of the internship after the contract agreement.

**[SP32]:** The system adds to the S&C registers in the calendar the submission of information, complaint or problem.

## 1.3. Definitions, Acronyms, Abbreviations

### 1.3.1. Definitions

- **Recommendation:** The process of identifying and suggesting internships to students and suitable candidates to companies using keyword searches, statistical analyses, or other matching algorithms.
- **Selection Process:** A structured process following a contact in which companies interview and assess students to determine fit.
- **Matching:** The automated or semi-automated process of pairing students with internships based on their profiles and internship requirements.
- **Feedback:** Information provided by students and companies on the quality of the matchmaking process or the internship experience, used to improve recommendations.
- **Internship Monitoring:** Ongoing observation on internship status.
- **Users:** referred to logged-in guests: Universities, Companies and Students.
- **Guest:** non logged visitors.
- **S&C Calendar:** a built-in calendar where all events (start and end date of internship, interviews) are visible and scheduled.

### 1.3.2. Acronyms

- **S&C:** Students & Companies.
- **UI:** User Interface.
- **API:** Application Programming Interface.

### 1.3.3. Abbreviations

- **[G\*]:** Goal.
- **[D\*]:** Domain Assumption.
- **[R\*]:** Functional Requirement.
- **[WP\*]:** World Phenomena.

- [MP\*]: Machine Phenomena.
- [SP\*]: Shared Phenomena.
- [UC\*]: Use Case.
- [S]: Student.
- [C]: Company.
- [U]: University.

## 1.4. Reference Documents

The document is based on the following materials:

- The specification of the RASD and DD assignment of the Software Engineering 2 course a.a 2024/2025.
- Slides of the course on WeBeep.

## 1.5. Document Structure

**Introduction** : It aims to give an overview of the project. In particular it's focused on the reasons and goals that are going to be achieved with its development.

**Overall Description** : This section provides a high-level overview of how the S&C platform operates, describing the roles and interactions of the main users: students, companies, and universities. It categorizes the phenomena in the system using the World and Machine paradigm and outlines assumptions and dependencies within the platform's domain.

**Specific Requirements** : Detailed functional and non-functional requirements for achieving the goals. Moreover, it contains more information useful for developers.

**Formal Analysis** : Formal modeling of the key phenomena using Alloy.

**Effort Spent** : Overview of the team's time allocation for each document section.

**References** : Bibliography listing all resources, documentation and software used in preparing this document.

# 2 | Overall Description

## 2.1. Product perspective

### 2.1.1. Scenarios

**Scenario 1: Giovanni Joins S&C to Begin His Internship Search** Giovanni, a university student, is looking to gain some practical experience through an internship. He's aware that securing a good internship is crucial for his future career, but he doesn't know where to begin his search. One evening, while discussing internship opportunities with his friends, Giovanni hears about the S&C platform, which helps match students with internships based on their CVs and company needs. Intrigued, he decides to download the application.

During the sign-up process, Giovanni provides the required information: his email, password, first name, last name and username. Once his account is created, he logs in and completes his profile. With everything set up, Giovanni is ready to explore the various internship opportunities available to him on the platform.

**Scenario 2: Zoogle Joins S&C to Find Talented Interns** Zoogle, a growing tech company, is on the lookout for talented students to join its internship program. The company values fresh ideas and practical skills and believes that collaborating with university students can bring innovation to its projects.

After hearing about the S&C platform, which connects companies with university students seeking internships, Zoogle decides to join. A representative from the company creates an account by providing the necessary information: the company's email, name, VAT number and username. Once the registration is complete, Zoogle gains access to the platform.

**Scenario 3: Zoogle Creates an Internship Posting on S&C** Zoogle is ready to recruit university students for its internship program. The HR team at Zoogle logs into the S&C platform to post a new internship position.

They fill out the internship posting form with all the necessary details: the Internship Title, Description, Required Skills, Duration, Location, and Eligibility Criteria. Once the details are submitted, the position becomes available to thousands of students seeking opportunities. With the new internship posting live, Zoogle is eager to connect with talented candidates like Giovanni.

**Scenario 4: Giovanni Navigates Suggested Companies and Applies for an Internship** Giovanni, eager to find the right internship, navigates through the suggested companies on his S&C homepage. These recommendations are generated by the system based on his experience, skills and interests.

As he explores the suggestions, Giovanni discovers an internship opportunity at Zoogle. He reviews the position and finds that it matches his qualifications perfectly. Excited, Giovanni clicks on the opportunity, begins the application process and submits his application to Zoogle.

**Scenario 5: Giovanni Provides Feedback on a Recommended Internship** After Giovanni reviewed the suggested internship at Zoogle and went back to the Homepage, a pop-up appears asking him for feedback on the recommendation. The system inquires whether the recommendation was useful, and Giovanni is prompted to provide a rating and a comment.

Giovanni, feeling that the suggestion matches his interests and qualifications, selects that the recommendation was useful and provides some positive rating and feedback. His input helps the system's statistical analysis engine, which uses this data to enhance future internship recommendations for other students based on skills, interests, CVs, and other factors.

**Scenario 6: Zoogle Reviews and Selects Student CVs** Zoogle's HR team wants to find the right candidate for their internship position, so they log in to the S&C platform. Upon logging in, they access to his profile page and after clicking on one of their insertion, two candidate's lists appears: the list of students who have applied to their internship and the list of students whose profiles have been recommended to them by the system, based on qualifications, experience, and preferences.

The HR team first goes through the list of students who applied for the internship. After reviewing the CVs, Zoogle's HR representatives select Giovanni's profile from that list, creating a contact. A chat is opened, allowing Giovanni and Zoogle to communicate directly and proceed to the next steps in the hiring process.

**Scenario 7: Giovanni and Zoogle Schedule an Interview** After the contact is established between Giovanni and Zoogle, they begin communicating through the chat feature on the S&C platform. Zoogle's HR team proposes scheduling an interview, and Giovanni agrees. The interview is then scheduled on the S&C calendar, and takes place outside the S&C platform, using a video conferencing tool or in person, depending on the company's preference.

After the interview's scheduling, both parties prepare for the next step in the hiring process, moving closer to a potential internship offer.

**Scenario 8: Zoogle Offers Giovanni an Internship Position** After conducting the interview with Giovanni, Zoogle's HR team is impressed by his skills, enthusiasm, and alignment with the company's values. Giovanni, in turn, felt a strong connection with the team and was excited about the role and responsibilities. Both Giovanni and Zoogle's HR team recognized a great mutual fit.

Given their positive impressions of each other, Zoogle's HR team proposes Giovanni the internship contract he applied for. Giovanni agrees on the contract's conditions and accepts the proposal through the chat.

**Scenario 9: Giovanni Submits a Complaint About His Internship and PoliCt decides to interrupt his internship** Giovanni has started his internship at Zoogle, but he encounters a problem that affects his experience. Unsatisfied with certain aspects of the internship, he decides to raise the issue through the S&C platform.

Giovanni logs into his profile, clicks chat section, enters into the chat and submit a complaint.

PoliCt, Giovanni's university, which monitors the internship program through the S&C platform, reviews Giovanni's complaint. After evaluating the situation, PoliCt decides to act and interrupts Giovanni's internship with Zoogle.

### 2.1.2. Class diagrams

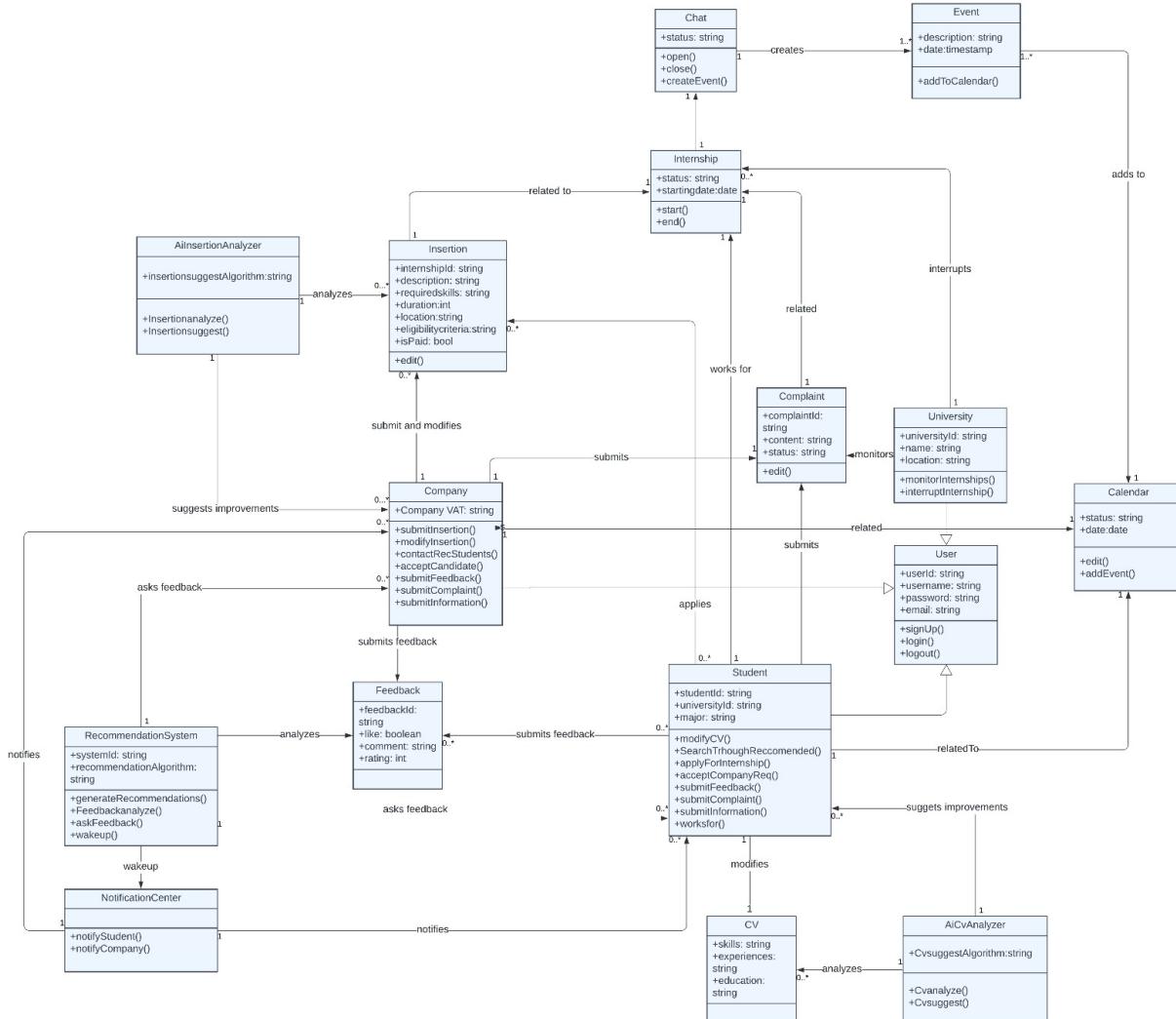


Figure 2.1: UML Class Diagram.

The Student and Company classes extend the abstract class User. Companies can create and modify Insertion instances representing internship opportunities, which are analyzed using the AiInsertionAnalyzer class to provide recommendations.

The RecommendationSystem generates fitted recommendations for students based on their CV details, analyzed by the AiCVAnalyzer. Complaints and feedback are handled through the Complaint and Feedback classes, respectively, and notifications are sent using the NotificationCenter. Additionally, the University class oversees internships, with functionalities to monitor or interrupt them if necessary, while Calendar and Event classes manage scheduling activities like interviews and internship start dates.

### 2.1.3. State diagrams

In this section are presented the State Diagrams of the S&C system representing all the possible operation that a User can perform.

**Sign Up:** The registration process on the S&C platform begins when a new guest decides to create an account. If you represent a university, click on the "Are you a University?" button to begin the tailored registration process. Alternatively, upon selecting the "Sign up" option, the guest is prompted to choose between a student or company account. Depending on their selection, the system tailors the registration flow to gather specific information. After the registration form, the system performs a check on the credentials provided. It ensures that the email or nickname is not already in use by another user and verifies that the password meets the platform's security criteria: it must be at least 8 characters long, contain at least one capital letter, one number, and a special character. If any of these conditions are not met, the user receives an error message prompting them to adjust their input. Once the credentials are validated, the system prompts all users to verify their email addresses by sending a verification link to the email provided during registration. The guest must click this link to confirm their email, ensuring that the contact information is valid. Upon successful email verification, users are directed to complete their profile.

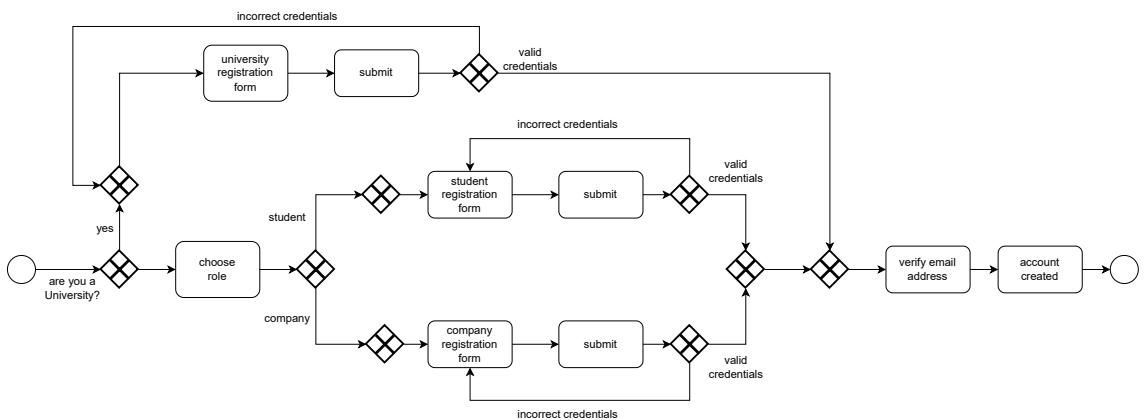


Figure 2.2: Sign Up State Diagram.

**Log In:** The login process begins when an existing user opens the web page. The guest is prompted to enter their registered email or username, along with their password. If the credentials are invalid, the system shows an error message and the user is asked to re-enter the information. If the credentials are successfully verified, the system checks if the user's email has been previously verified. If the email verification step was skipped, the guest is prompted to verify their email before proceeding. Upon successful verification of all login details, the user is granted access to their dashboard.

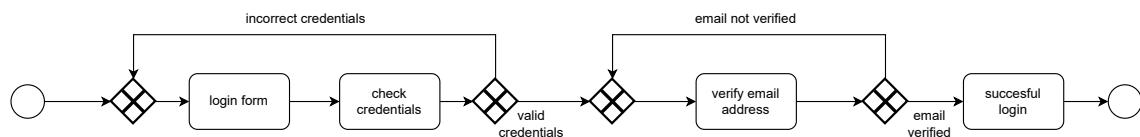


Figure 2.3: Log In State Diagram.

**Enhance your CV:** The process begins when a student opens his profile menu and selects the "Enhance" button near the CV. Upon clicking the button, the platform initiates the enhancement process by analyzing the student's existing CV using AI algorithms. It identifies areas for improvement, including highlighting key skills, rephrasing descriptions and adding meaningful action verbs. Once the analysis is complete, the AI generates a set of suggestions and provides a preview of the enhanced CV. The student can review the suggested changes and manually modify his CV. After finalizing the edits, the student uploads the enhanced CV to their profile.

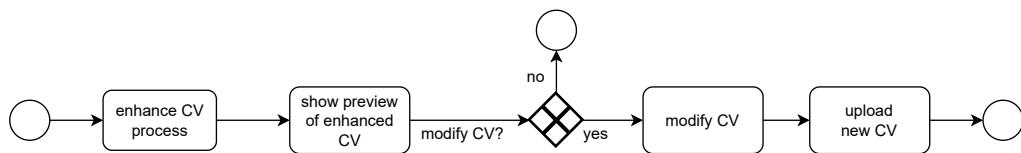


Figure 2.4: Enhance CV State Diagram.

**Apply for an Internship:** The application process for an internship begins when a student clicks on an internship announcement that catches their interest. Upon selecting the internship, the student is redirected to a detailed view of the internship posting, where they can review all relevant information about the job description. If the student decides to proceed, they click the "Apply" button. This action inserts the S's profile into the candidates list of the C for the internship that he applied for.

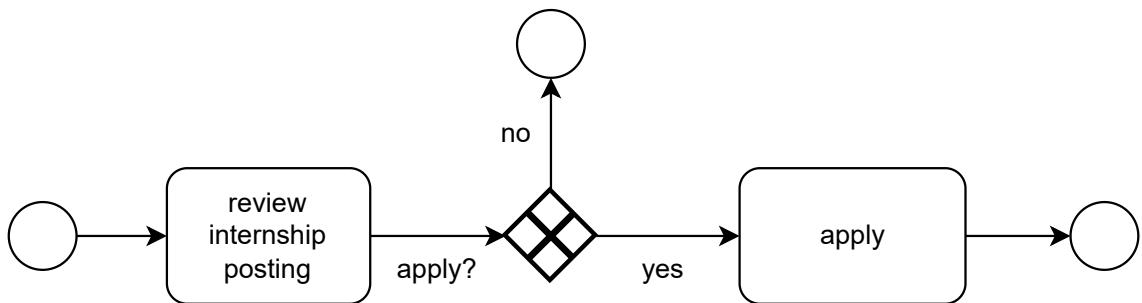


Figure 2.5: Apply for an Internship State Diagram.

**Require Feedback:** After a user (either a student or a company) closes one of the recommended profiles, a pop-up appears prompting the user to provide feedback and rating about the recommendation. The pop-up includes the question, "Was this recommendation useful?" along with an option to give a rating out of 5 stars and a comment section. This feedback helps the platform improve its recommendation engine, ensuring more accurate matches in the future.

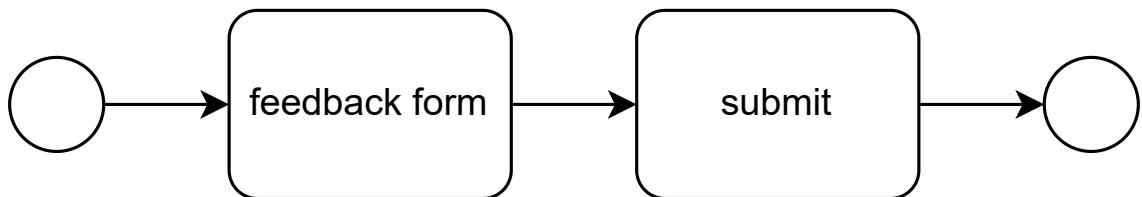


Figure 2.6: Require Feedback State Diagram.

**Insert a new internship and enhance the internship description:** After logging in, the company representative selects the "Post" button on his dashboard to begin the creation process. The system shows a form to input all necessary details about the internship. After compiling all the details the representative can also click on the button "Enhance". Upon clicking the button, the platform initiates the enhancement process by analyzing the existing internship description using AI algorithms. The AI evaluates various aspects of the description, such as clarity, structure, and attractiveness to potential candidates. It identifies areas for improvement, such as refining the job title, rephrasing the responsibilities and requirements for better readability, and highlighting the benefits of the internship. Once the analysis is complete, the AI generates a set of suggestions. The company representative can review the suggested changes and manually modify the insertion. Once the representative is satisfied with the details, they submit the internship posting. The system performs validation checks to ensure the following:

- All mandatory fields are completed.
- Data provided is in the correct format (e.g., valid dates for duration, proper email for contact).

After successful validation, the internship is published on the platform and becomes visible to students.

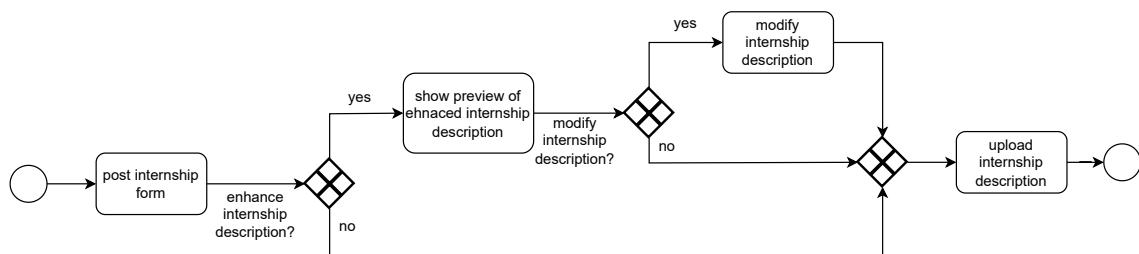


Figure 2.7: Insert and Enhance Internship State Diagram.

**Selection Process and schedule of interview:** The selection process begins when a contact is established between a student and a company on the S&C platform. Once the contact is created, a chat opens between the student and the company, allowing them to communicate directly. Through the chat, they schedule an interview at a mutually convenient time. The company proposes a date and time until the student accepts. The event is now added to both the company and the student S&C calendar. The interview is conducted separately from the S&C platform, either in person or using an external virtual meeting tool.

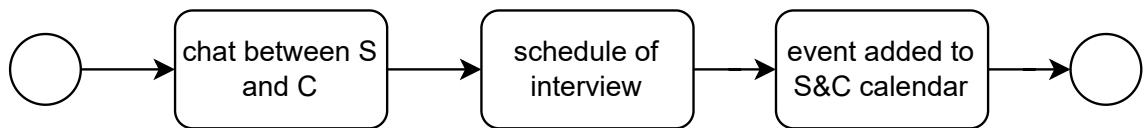


Figure 2.8: Selection and Schedule Interview State Diagram.

**Finalizing Process:** The interview was successful and the company proposes an internship offer through chat, specifying start and end date. Now the student has to make a decision, he can accept or refuse through the chat. If the student accepts the event is added to the S&C calendar.

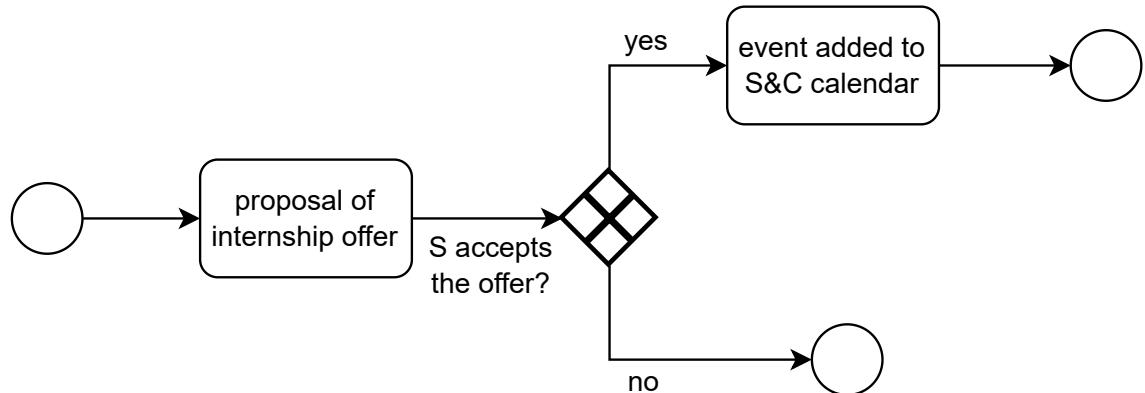


Figure 2.9: Finalizing Process State Diagram.

**Report a Problem:** During the internship, either the student or the company may encounter an issue that needs to be addressed. To report a problem, they can log into the S&C platform and open the chat related to the internship. Here they can click the 'Help' button and then "Report a Problem". This action opens a form where the user can describe the nature of the issue in detail, providing all necessary information. Once the complaint is submitted, the event is added to the S&C calendar, and an email is automatically sent to the student's university, alerting them to the problem.

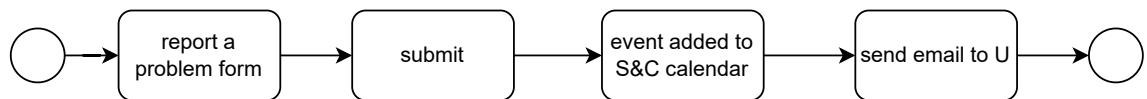


Figure 2.10: Report Problem State Diagram.

**Interrupt Internship:** If the university receives an email about a problem reported during an internship, they can log into the S&C platform to review the issue in detail. After evaluating the complaint, if the university determines that the situation warrants action, they can click on the "Interrupt Internship" button. This action terminates the internship between the student and the company, ensuring the well-being of the involved parties and addressing serious concerns effectively.

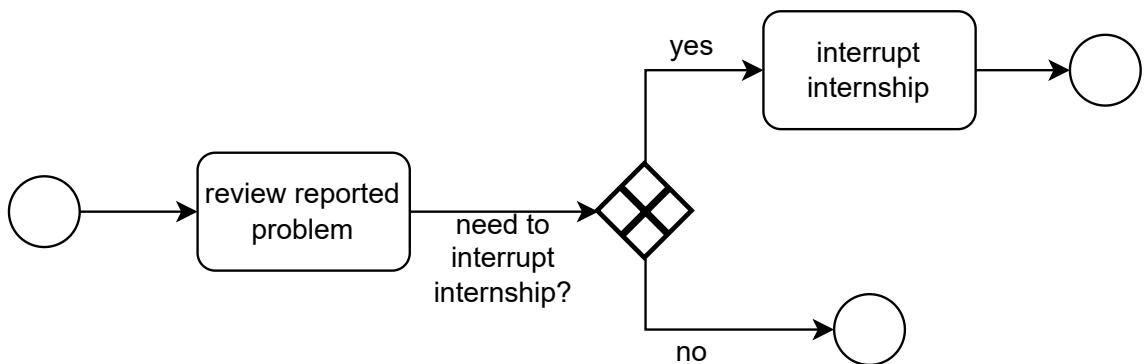


Figure 2.11: Interrupt Internship State Diagram.

## 2.2. Product functions

Here is a summary of the main functions of the S&C system:

**Sign Up:** The guest begins by selecting the correct category then by clicking the "Sign Up" button, a corresponding form is generated. Then fills out the form by providing the necessary information. A confirmation mail is required to complete the process.

**Login:** The guest, independently by his category, signs in after typing his email/username and password in the login form and clicking the "Login" button.

**Modify your profile:** The S, C can modify his personal profile information, such as personal data, profile image, password, email, username.

**Enhance the CV:** The S can improve his CV on his profile's page by clicking the "Enhance" button thanks to suggestions given by the system.

**Recommend internships:** The S can look for an internship in the recommendation's section made by the system.

**Surf through available internships:** The S can surf through the search bar on the main page and look for the most pertinent insertion.

**Ask feedback/suggestions/rating:** The S, C submits feedback, suggestions and rating about the recommended proposal, when requested.

**Notify about new internship:** The S is notified when an internship, considered relevant for him, is found by the system.

**Notify about new possible candidates:** The C, after submitting an insertion, is notified when the system finds a relevant match.

**Submit an internship's insertion:** The C can submit an internship insertion clicking on "Post" button, by filling in all the details needed.

**Improve insertion:** The C can enhance his insertion thanks to suggestions given by the system, clicking on "Enhance" on the insertion's page.

**Apply for an internship:** The S can apply for a preferred internship, by clicking on "Apply" button on the insertion page.

**Accept a candidate:** The C can accept one of the candidates that made an application.

**Contact a recommended student:** The C can contact one of the proposed candidates by the recommendation system.

**Accept a company:** The S can accept a company after being contacted

**Create a chat/manage interviews:** After a successful match between the C and the S, a chat is established in order to manage the next steps.

**Sending a contract proposal:** After a successful interview, the C sends a contract proposal through the chat.

**Accept/refuse a contract proposal:** After receiving a contract proposal, the S can decide to accept or refuse the contract through the chat.

**Submit Information:** The S,C can submit information about ongoing internships through the chat.

**Submit Complaint:** The S, C can submit complaints/problems during the internship period through the chat.

**Monitor Internship:** The U can monitor complaints/information submitted by both parties in its dashboard.

**Interrupt Internship:** The U can interrupt an internship if necessary.

**Review S&C calendar:** The C and S can review their personal events through the S&C calendar.

### 2.3. User characteristic

There are two main users S,C and one side U. Here we have a summary of users characteristics and needs:

- The **Student**, would like to look for a suitable internship opportunity that aligns with his interests. He needs guidance in identifying the most relevant offers and support throughout the application process. Furthermore, he would like to communicate easily with the company and with the university, in order to address any potential issues.
- The **Company** seeks to efficiently recruit qualified students for new positions, monitor candidates during hiring, and track internship progress. It aims to address student requests and maintain clear communication with both students and their universities to resolve any issues promptly.
- The **University** would like to manage its students during the internship period and to intervene promptly when necessary.

## 2.4. Assumptions, dependencies and constraints

The following assumptions are made for the domain. They represent foundational expectations and conditions that the S&C platform will take for granted. These assumptions define the operational environment and set the context for the system's design and functionality. They must be validated to ensure correct platform behavior and a common understanding of the factors influencing the platform's success.

[D1]: S must be enrolled in a U.

[D2]: The related U should be already registered when the S signs up.

[D3]: S should already have a valid CV to be uploaded.

[D4]: S' CV should be a PDF document.

[D5]: The User can easily access his eMail provider.

[D6]: S,C provide accurate and truthful information in their profiles, CVs, and internship description.

[D7]: U are willing and able to use the S&C platform to monitor internship statuses, address complaints, and respond to student or company feedback as needed.

[D8]: C on the platform are assumed to be legitimate and offer genuine internships, operating legally and ethically without verification by S&C.

[D9]: The S&C recommendation engine achieves high accuracy when S and C provide detailed, complete, and accurate information in their CVs and job descriptions.

[D10]: The AI system integrated into the S&C platform effectively enhances students' CVs and companies' internship postings, providing accurate and useful suggestions to improve the matching process.

[D11]: The eMail provider functions reliably, ensuring that all emails, including verification and notification messages, are successfully delivered to the intended recipients.

[D12]: Students and companies are committed to the internship process and will engage with features such as scheduling interviews and responding to feedback.

[D13]: Companies must clearly communicate interview requirements (e.g., time, format, platform) to students during scheduling.

[D14]: Interviews must be conducted either in person or via secure external tools, with students and companies responsible for ensuring their availability and functionality.



# 3 | Specific Requirements

This section is devoted to a specific description of every kind of requirement the system has to deal with in order to achieve all the functionalities described.

## 3.1. External Interface Requirement

### 3.1.1. User interfaces

The S&C's user interface will be a website, developed in order to be used by all the possible users. The following images provide a high-level overview of the interfaces, offering a peek into their structure and functionality. A detailed analysis of these interfaces will be presented in the Design Document.

**Access Page:** When a user navigates to the website for the first time, the access page (**Figure 3.1**) serves as the initial interface. Here, users are prompted to log in using their credentials. If the user does not already have an account, they are provided with an option to register and create a new one. Additionally, universities can also access this page to proceed with their specific login or registration process, ensuring seamless on-boarding for both individual users and institutional entities.

**Student Homepage:** When a student enters the homepage (**Figure 3.2**), they are greeted with an organized layout. On the left side, an overview of their profile is displayed, along with a section highlighting past and upcoming events. On the right side, a toolbar is provided for searching companies, complemented by an overview of recommended companies. At the top of the page, a small toolbar houses icons for easy access to the homepage, chat, notifications, help, settings, and the student's personal profile.

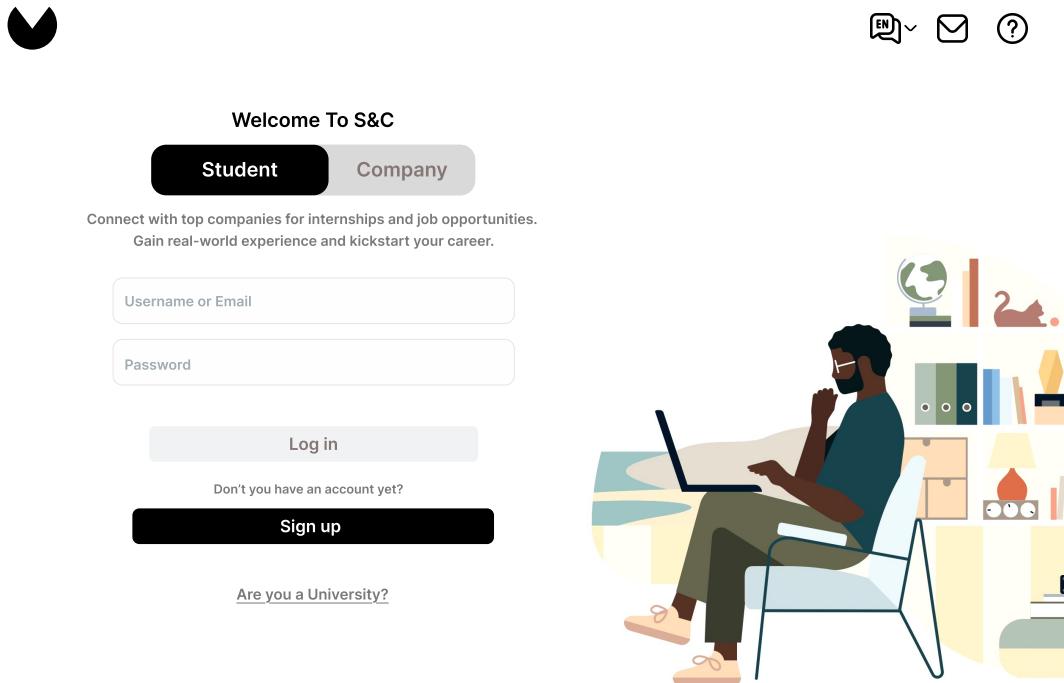


Figure 3.1: Access Page UI.

The image shows the student homepage. At the top right are icons for settings, help, and user profile. On the left, there's a profile card for 'Giovanni Rossi STUDENT' (MSc Computer Science Engineering @POLIMI). Below it, under 'Internship', there are sections for 'Ongoing' (Zoogle Inc., End: 30-03-2025), 'Previous' (Facebook Inc., Ended: 30-09-2024), and 'Next' (Amazon Inc., Start: 01-05-2025). Under 'Highlights', there's a card for 'IPM' (Interview DEC, IPM, 20 interviews). On the right, the main content area is titled 'Find your next job' with a search bar. It features a section 'Recommended for you' with three job cards: 'Zoogle Inc.' (Product Manager, Data Platform, San Francisco, California), 'Facebook Inc.' (Software Engineer, Developer Platform, San Francisco, California), and 'Amazon Inc.' (Product Designer, San Francisco, California). Each job card has a '...' button to its right.

Figure 3.2: Student Homepage UI.

**Chat page:** When a user navigates to the chat page (**Figure 3.3**), they are presented with a clear layout. On the left side, a list of open chats is displayed, allowing quick access to previous conversations. On the center-right side, the selected chat appears, featuring a text input bar, a send button, and a "+" button for additional operations, such as adding information, submitting a complaint, and other actions only for companies.

**Notifications page:** When the user accesses the notifications page (**Figure 3.4**), they are presented with a list of notifications they have received, ordered from the most recent to the least. This provides an easy way to stay up-to-date with important alerts and messages.

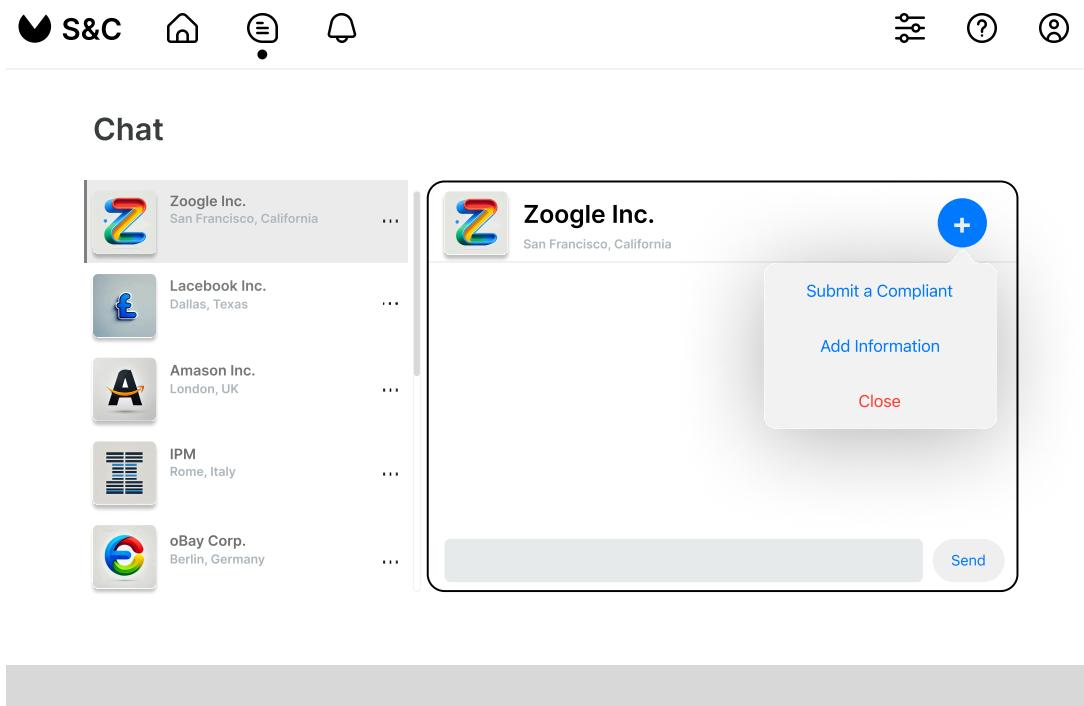


Figure 3.3: Chat Page UI.

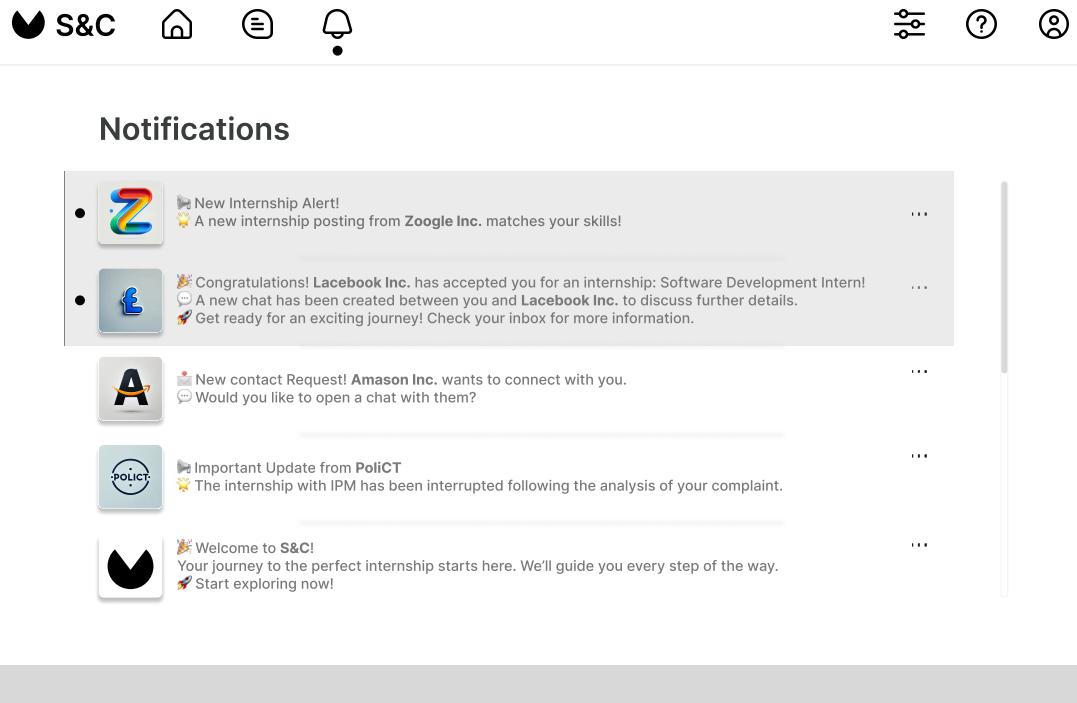


Figure 3.4: Notifications Page UI.

### 3.1.2. Hardware interfaces

The system will be accessible from any device equipped with an internet browser and a stable internet connection. Users can choose their preferred device, such as a computer, tablet, or smartphone. However, it is recommended to use a computer for a more seamless experience when navigating through the internship postings.

### 3.1.3. Software interfaces

The S&C platform integrates key software tools for effective functionality:

- **Email Provider:** Sends confirmation emails for user registration and notifications for matches, updates, and reminders.
- **Notification System:** Delivers real-time updates, such as new applications or interview schedules, directly on the platform.
- **Data Analytics Services:** Offers data-driven insights to enhance recommendations and feedback.
- **AI Tools:** Improve user CVs and internship descriptions to optimize matches and user experience.

### 3.1.4. Communication interfaces

The communication interfaces required by the S&C platform are as follows:

- **HTTPS Protocol:** The platform will use HTTPS to ensure secure communication between users and the system, encrypting all data exchanged to protect user privacy and information.
- **SMTP (Simple Mail Transfer Protocol):** The platform will utilize SMTP for sending emails, including confirmation emails during registration, notifications about internship opportunities, application statuses, interview schedules, and other important updates to users.

## 3.2. Functional requirements

### 3.2.1. Requirements

Here is the list of functional requirements organized by user classes:

- **Guest**

[R1]: The system should allow an unregistered guest to sign up.

- **Student**

[R2]: The system should allow registered students to log in.

[R3]: The system should allow students to insert their CVs and manage their profile information (e.g. personal data, skills and profile photo).

[R4]: The system should provide personalized suggestions to students for enhancing their CVs to improve their chances of being selected.

[R5]: The system should notify students when internships matching their skills, experiences, and interests become available.

[R6]: The system should allow students to browse available internship postings.

[R7]: The system allows S to visualize the profile of other C.

[R8]: The system should allow students to apply only for open internships.

[R9]: The system should allow S to provide feedback, suggestions and rating to improve the recommendation process.

**[R10]:** The system allows S to establish a chat with C when mutual interest is identified.

**[R11]:** The system should assist S in scheduling interviews.

**[R12]:** The system should allow S to track the status of their applications and final selections.

**[R13]:** The system should not allow a student already engaged in an internship to apply for other internships.

**[R14]:** The system should provide S with a mechanism to report issues or complaints related to the ongoing internship they are working on.

**[R15]:** The system allows a maximum of one unresolved complaint per S during an ongoing internship.

**[R16]:** The system should allow S to review the S&C Calendar.

**[R17]:** The system should insert the S's events on the S&C Calendar.

- **Company**

**[R18]:** The system should allow registered companies to log in.

**[R19]:** The system should allow C to manage their organization profile.

**[R20]:** The system should allow C to post new internship opportunities, specifying details such as required skills, tasks, and benefits.

**[R21]:** The system should provide suggestions to C to improve their internship postings, making them more appealing to S.

**[R22]:** The system allows C to visualize the profile of other Users.

**[R23]:** The system should inform C about the availability of S CVs corresponding to their needs.

**[R24]:** The system should allow C to provide feedback and suggestions to improve the recommendation process.

**[R25]:** The system allows C to establish a chat with S when mutual interest is identified.

**[R26]:** The system should allow only one worker per internship.

**[R27]:** The system should enable C to track the progress of their internship selection processes, including applications and decisions.

**[R28]:** The system should allow C to submit complaints or information related to their ongoing internships.

**[R29]:** The system allows a maximum of one unresolved complaint per C during an ongoing internship.

**[R30]:** The system should not allow applications for C's internships that are currently ongoing.

**[R31]:** The system should allow C to review the S&C Calendar.

**[R32]:** The system should insert the C's events on the S&C Calendar.

- **University**

**[R33]:** The system should allow U to log in.

**[R34]:** The system should allow U to monitor internships of their students.

**[R35]:** The system should allow U to interrupt ongoing internships of their S when a complaint was submitted.

### 3.2.2. Use case diagrams

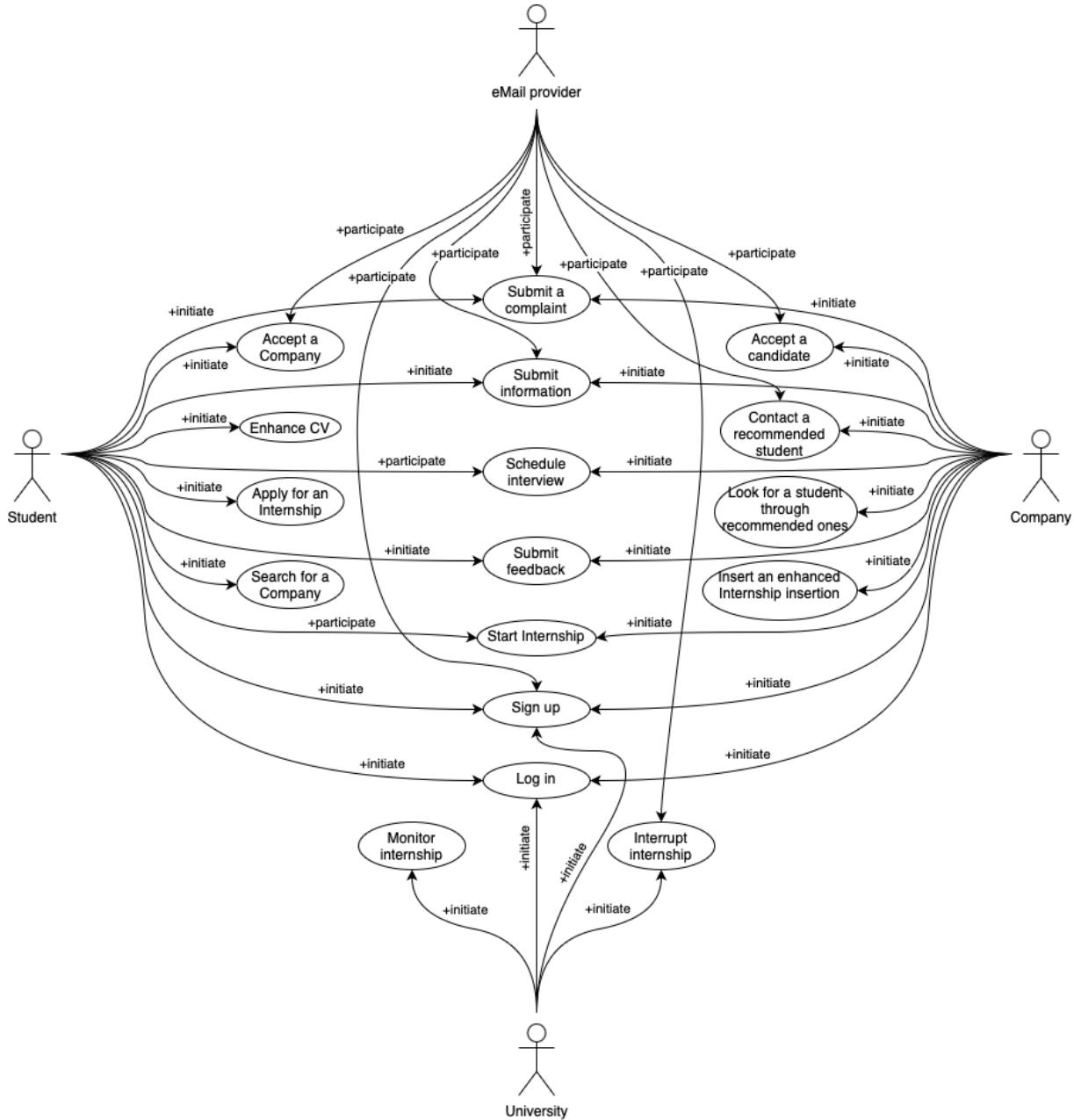


Figure 3.5: Use Cases Diagram for all the Users.

### 3.2.3. Use cases

This chapter presents core use cases, showcasing the application's real-world impact and versatility.

#### UC1. Sign Up

<b>Name</b>	Sign Up
<b>Actor</b>	Guest, eMail provider
<b>Entry conditions</b>	The Guest wants to create an account and knows the S&C URL
<b>Event Flow</b>	<p>1 - The Guest enters the URL on the search bar of his browser.</p> <p>2 - The Guest clicks on the “Sign Up” button entering in the registration page.</p> <p>3 - The Guest selects the correct registration form by selecting the right option between Student, Company or University.</p> <p>4 - The Guest enters his personal information: eMail, password, name and username for the S; eMail, password, name, VAT Name and username for the C; eMail, password, name, PEC, Name and username for the U.</p> <p>5 - The Guest clicks on the “Accept &amp; Join” button.</p> <p>6 - The system checks the credentials.</p> <p>7 - If the check is successful the system sends a confirmation eMail to the User through the eMail provider.</p>
<b>Exit condition</b>	The system lets the User enter the website.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The username is already in use, so the system shows an error message and the registration form is shown again.</li> <li>• The eMail is already in use, so the system shows an error message and the registration form is shown again.</li> <li>• The password is not valid: it's less than 8 characters long and/or it doesn't contain any uppercase letter and/or it doesn't contain any lowercase letter and/or it doesn't contain any number and/or it doesn't contain any special character (!,#,&amp;\$,?,+,-). The system shows an error message and the registration form is shown again.</li> </ul>

Table 3.1: Sign Up use case.

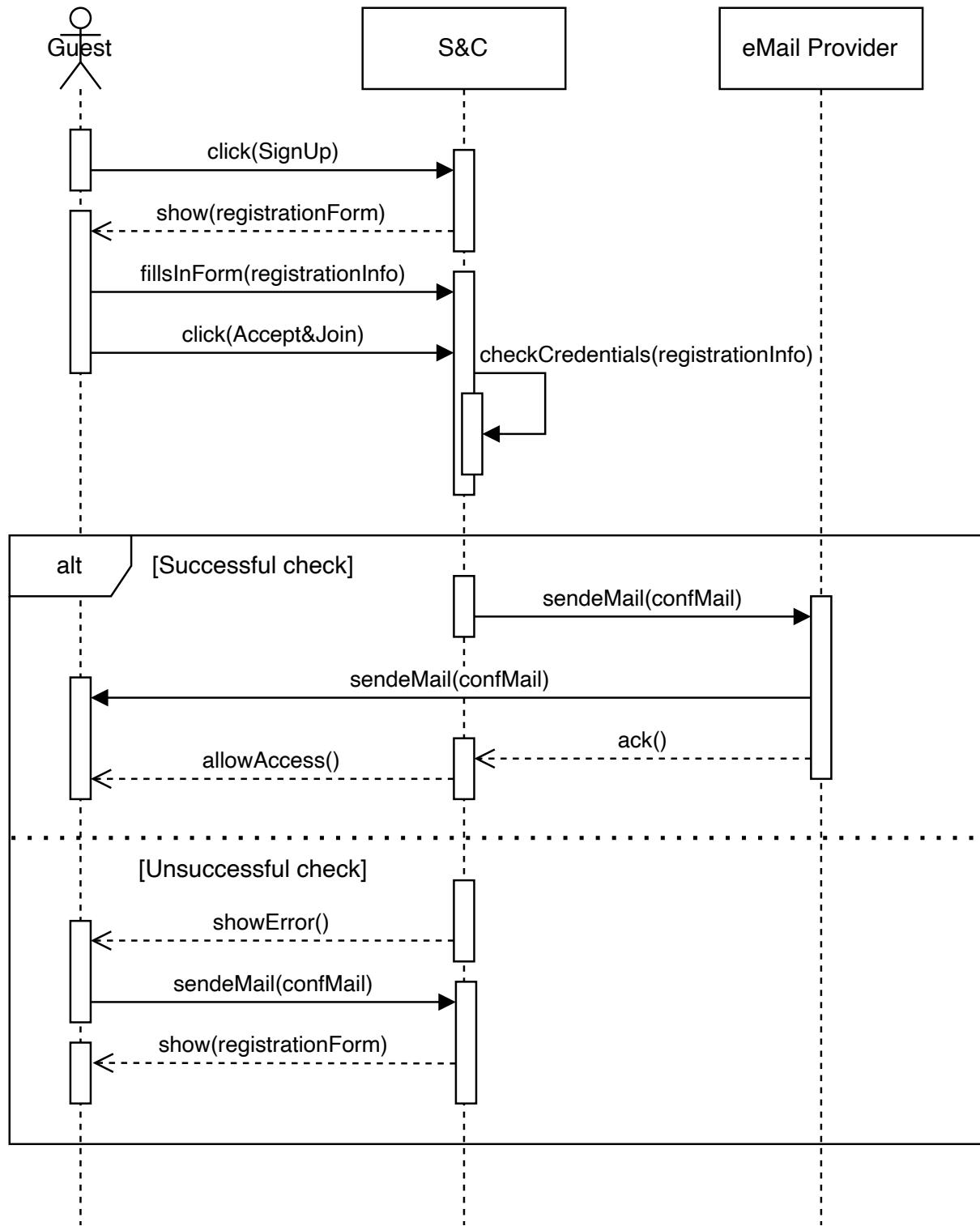


Figure 3.6: Sign Up sequence diagram.

## UC2. Login

<b>Name</b>	Login
<b>Actor</b>	Users
<b>Entry conditions</b>	The Guest wants to access his account
<b>Event Flow</b>	<p>1 - The Guest enters the URL of S&amp;C on the search bar of his browser.</p> <p>2 - The Guest selects the right option between Student and Company button or in the case of a University clicks on the “Are you a University?” button.</p> <p>3 - The Guest enters the email and password in the correct form.</p> <p>4 - The Guest clicks on the “Login” button.</p> <p>5 - The system checks the credentials. If the check is successful it redirects the user to the right main page, according to his role.</p>
<b>Exit condition</b>	The User is logged in and can surf into the website.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Incorrect eMail or password. An error message is shown and the User is redirected back to the Login page.</li> </ul>

Table 3.2: Login use case.

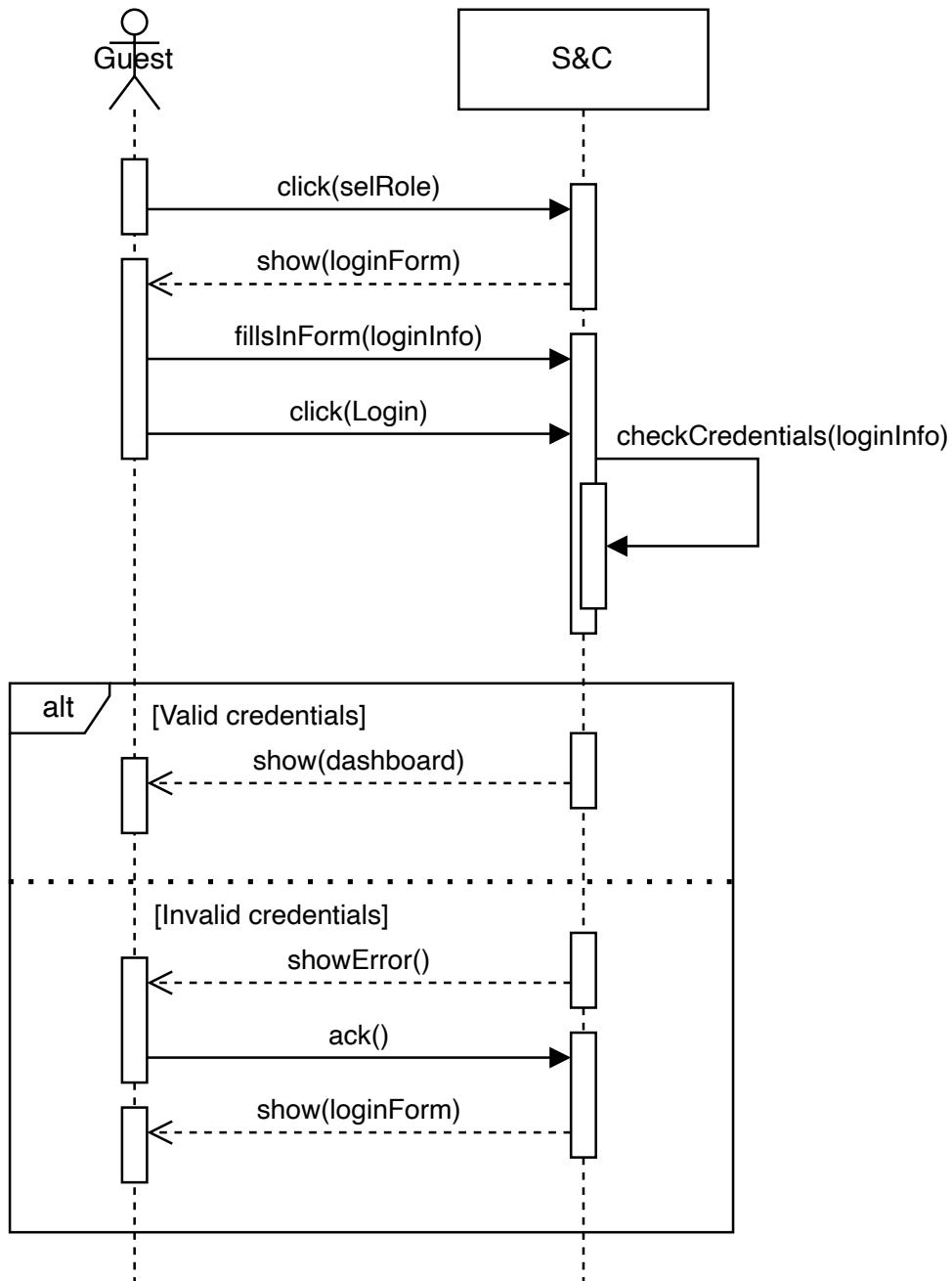


Figure 3.7: Login sequence diagram.

### UC3. Insert an enhanced Internship Insertion

<b>Name</b>	Insert an enhanced Internship Insertion
<b>Actor</b>	C
<b>Entry conditions</b>	The C is logged in and wants to post a new insertion.
<b>Event Flow</b>	<p>1 - The C clicks on “Post” at the bottom-right of his homepage.</p> <p>2 - The C fills in the forms with all the information needed about his position.</p> <p>3 - The C clicks on the “Enhance” button.</p> <p>4 - The system analyzes the insertion and gives suggestions on how to improve project description.</p> <p>5 - The C reviews the suggestions given by the system and modifies the insertion if needed.</p> <p>6 - The C clicks on the “Post” button.</p>
<b>Exit condition</b>	The system publishes the C's insertion and shows to the C his homepage.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The insertion is very well-done and there are no suggestions available, so the system shows an error message.</li> </ul>

Table 3.3: Insert an enhanced Internship Insertion use case.

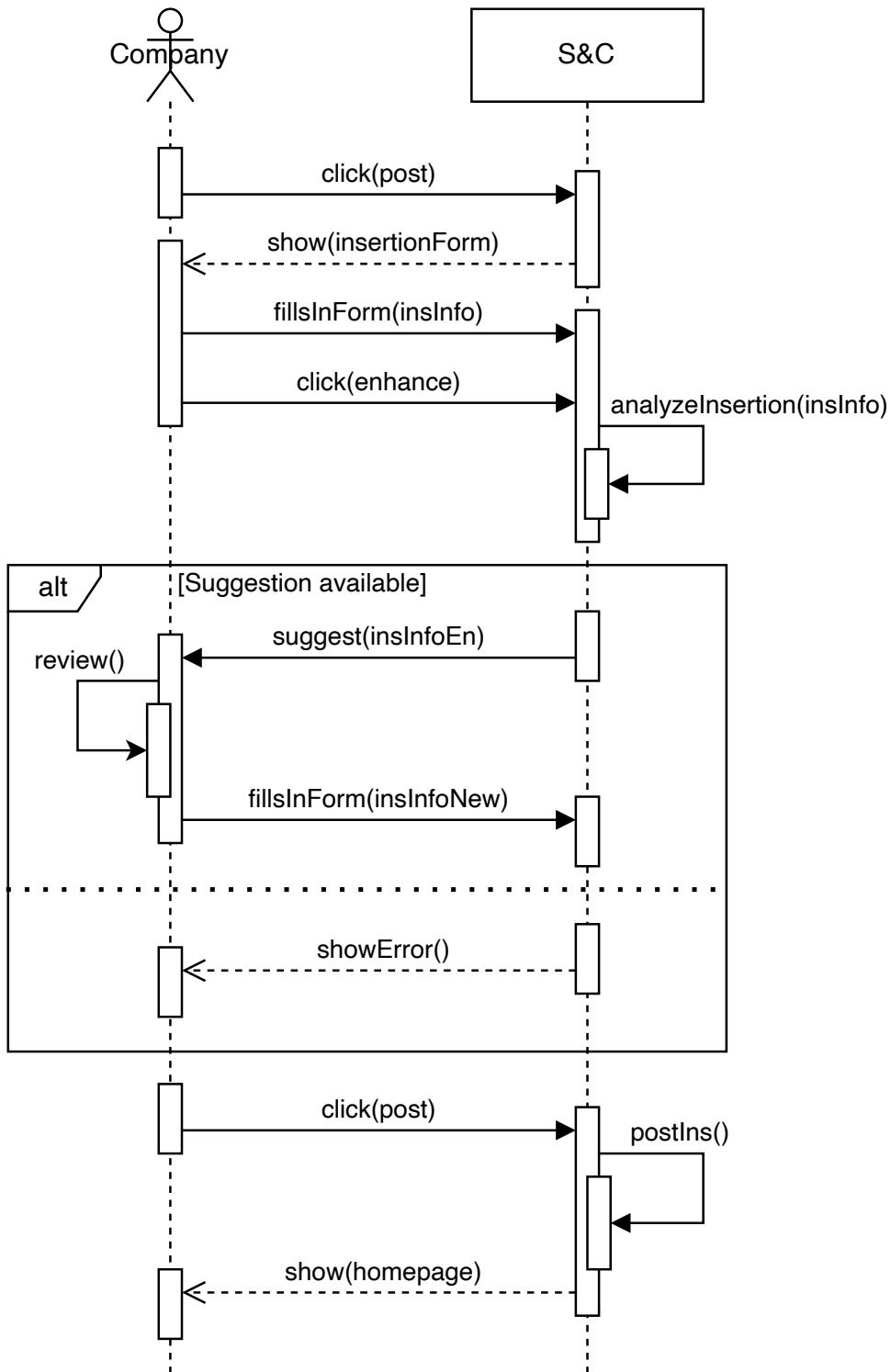


Figure 3.8: Insert an enhanced Internship Insertion sequence diagram.

## UC4. Enhance CV

<b>Name</b>	Enhance CV
<b>Actor</b>	S
<b>Entry conditions</b>	The S is logged in, has already uploaded a CV and wants to improve it.
<b>Event Flow</b>	<p>1 - The S opens his personal profile page.</p> <p>2 - The S clicks on the “Enhance” button near the CV section.</p> <p>3 - The system analyzes the CV and gives suggestions on how to improve it.</p> <p>4 - The S reviews suggestions given by the system.</p> <p>5 - The S makes some suggested changes on the file of the CV.</p> <p>6 - The S reuploads a new version of CV.</p>
<b>Exit condition</b>	The system shows the S profile page.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The CV is very well-done and there are no suggestions available, so the system shows an error message.</li> </ul>

Table 3.4: Enhance CV use case.

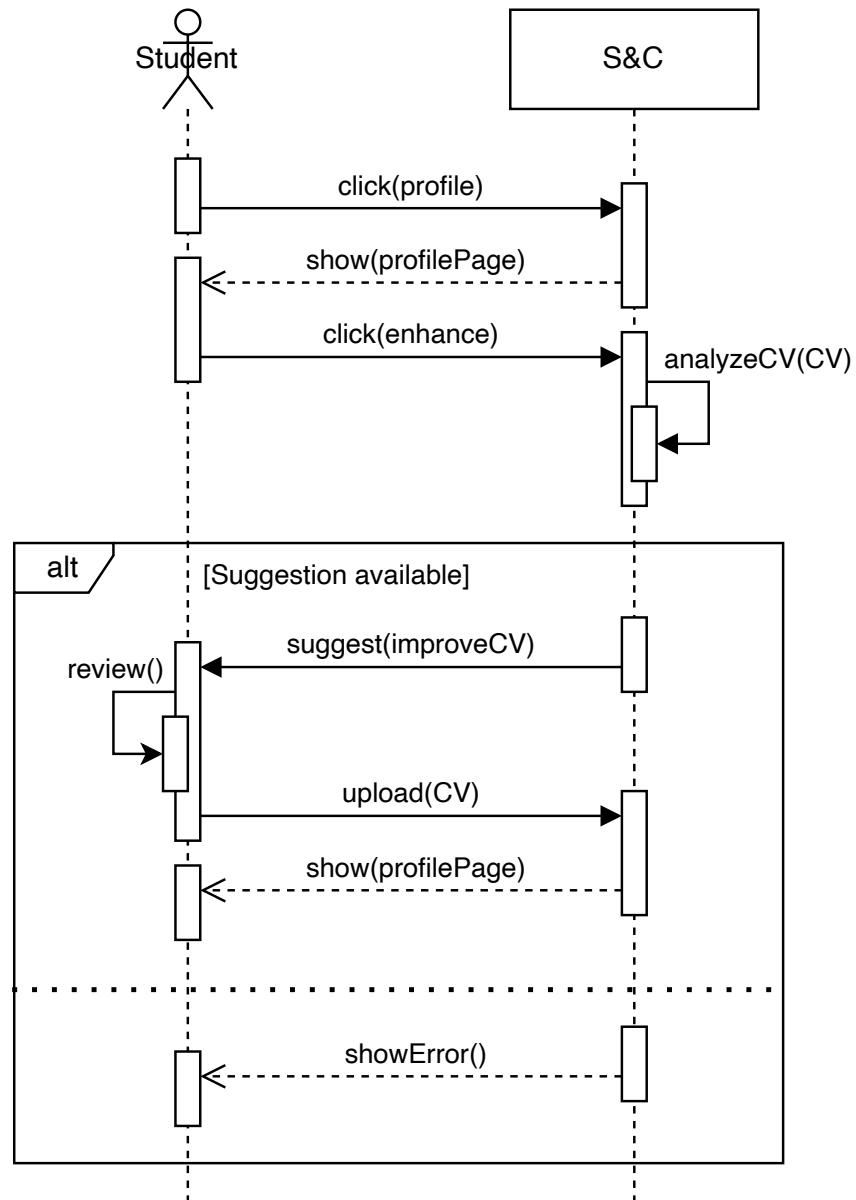


Figure 3.9: Enhance CV sequence diagram.

## UC5. Submit a feedback

<b>Name</b>	Submit a feedback
<b>Actor</b>	S, C
<b>Entry conditions</b>	The S (C) wants to look for a recommended internship (student) and is on the homepage.
<b>Event Flow</b>	<p>1 - The S (C) scrolls through the recommended insertions (students).</p> <p>2 - The S (C) clicks on a company's insertion (student's profile).</p> <p>3 - The system shows the review of the insertion (student).</p> <p>4 - The S (C) closes the window.</p> <p>5 - The system shows a pop-up asking the S (C) to submit a feedback, including rating and some comments on the received recommendations.</p> <p>6 - The S (C) submits his feedback.</p>
<b>Exit condition</b>	The system saves the feedback to feed the recommendation system and shows the homepage.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The system does not find any recommended C (S) available for the S (C).</li> </ul>

Table 3.5: Submit a feedback use case.

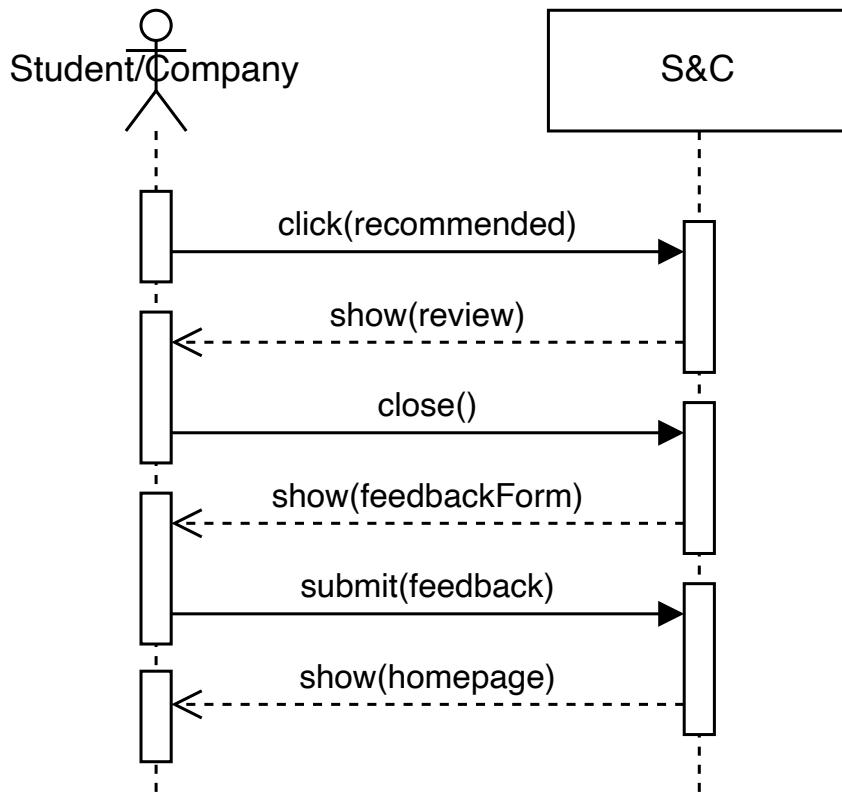


Figure 3.10: Submit a feedback sequence diagram.

## UC6. Search for a company

<b>Name</b>	Search for a company
<b>Actor</b>	S
<b>Entry conditions</b>	The S is on the homepage and wants to search for a specific C.
<b>Event Flow</b>	<p>1 - The S clicks on the search bar.</p> <p>2 - The S writes the C's name and clicks 'Enter' on the keyboard or the button 'Search'.</p> <p>3 - The system shows a list of the matching results.</p> <p>4 - The S clicks on the C profile.</p>
<b>Exit condition</b>	The system shows the C profile.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>The system doesn't find any matching result.</li> </ul>

Table 3.6: Search for a company use case.

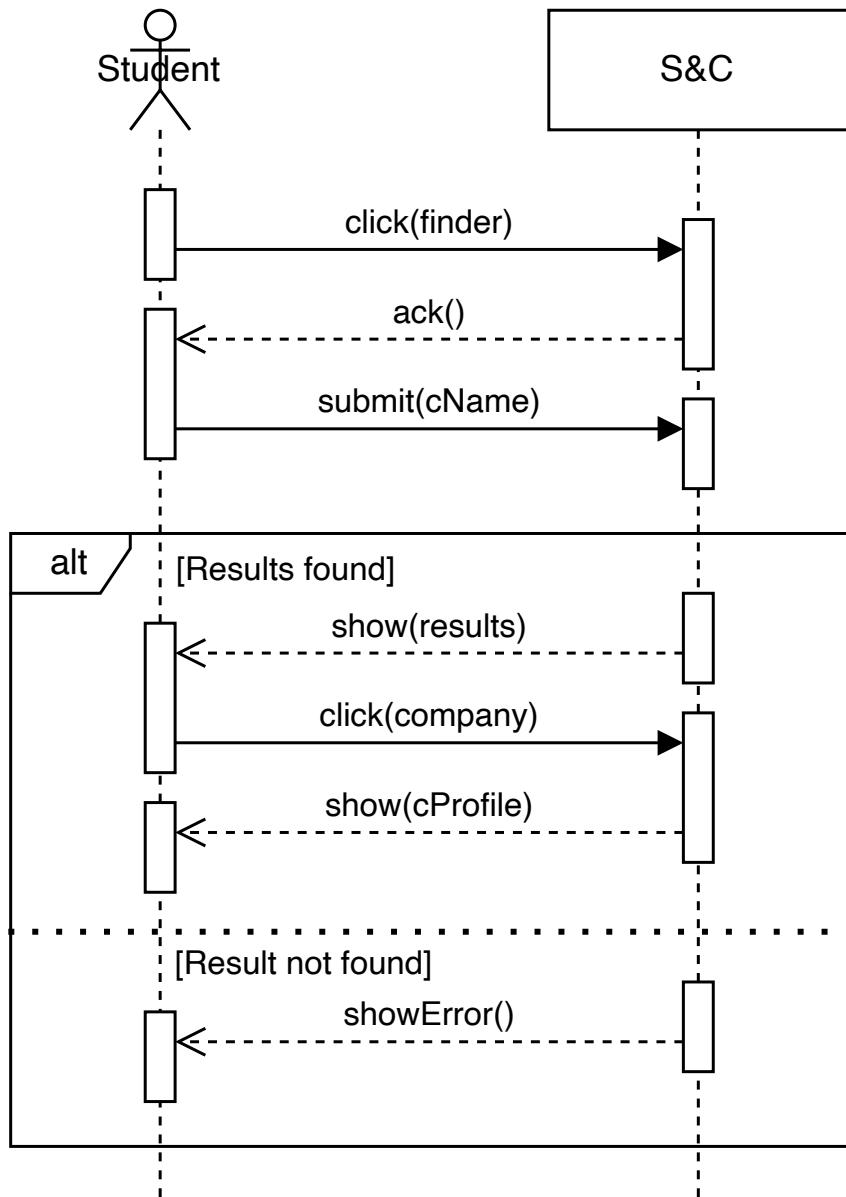


Figure 3.11: Search for a company sequence diagram.

## UC7. Apply for an internship

<b>Name</b>	Apply for an internship
<b>Actor</b>	S
<b>Entry conditions</b>	The S is logged in and wants to apply for an internship.
<b>Event Flow</b>	1 - The S finds an interesting insertion. 2 - The S clicks on the relative insertion. 3 - The system shows the review of the insertion post. 4 - The S applies for it, by clicking “Apply”.
<b>Exit condition</b>	The system inserts S into the insertion’s candidates of the C.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>No specific exceptions for this use case.</li> </ul>

Table 3.7: Apply for an internship use case.

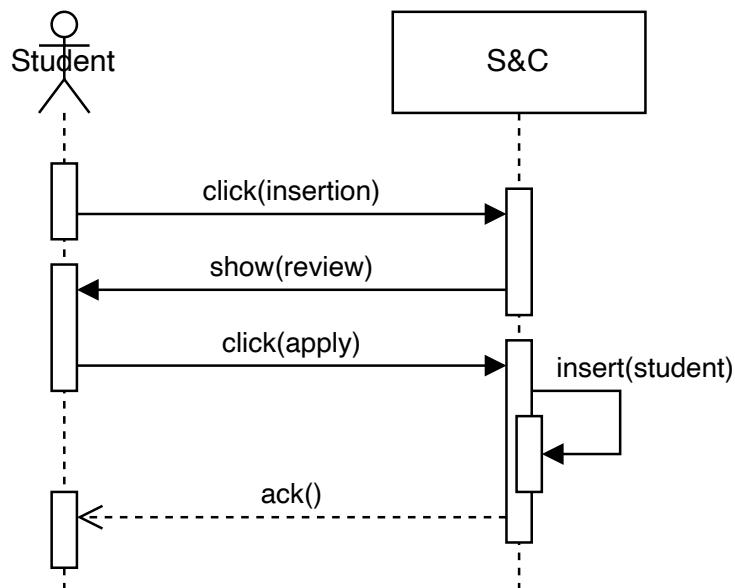


Figure 3.12: Apply for an internship sequence diagram.

## UC8. Accept a candidate

<b>Name</b>	Accept a candidate
<b>Actor</b>	C, S, eMail provider
<b>Entry conditions</b>	The C is logged in and wants to find a new candidate from the ones that applied for the internship.
<b>Event Flow</b>	<p>1 - The C goes on his profile page.</p> <p>2 - The C clicks on "Insertion".</p> <p>3 - The C clicks on the selected insertion post.</p> <p>4 - The system shows the review of the insertion.</p> <p>5 - The C opens the candidates' section.</p> <p>6 - The C scrolls through all the possible candidates.</p> <p>7 - The C finds a S's profile.</p> <p>8 - The C clicks on the S's profile.</p> <p>9 - The system shows the review of the S profile.</p> <p>10 - The C decides to accept the S's profile by clicking on the "Accept" button.</p> <p>11 - The system sends an eMail to the S notifying him of the decision of the C through the eMail provider.</p>
<b>Exit condition</b>	The system creates a chat between the S and the C.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• There are no candidates available for the internship.</li> </ul>

Table 3.8: Accept a candidate use case.

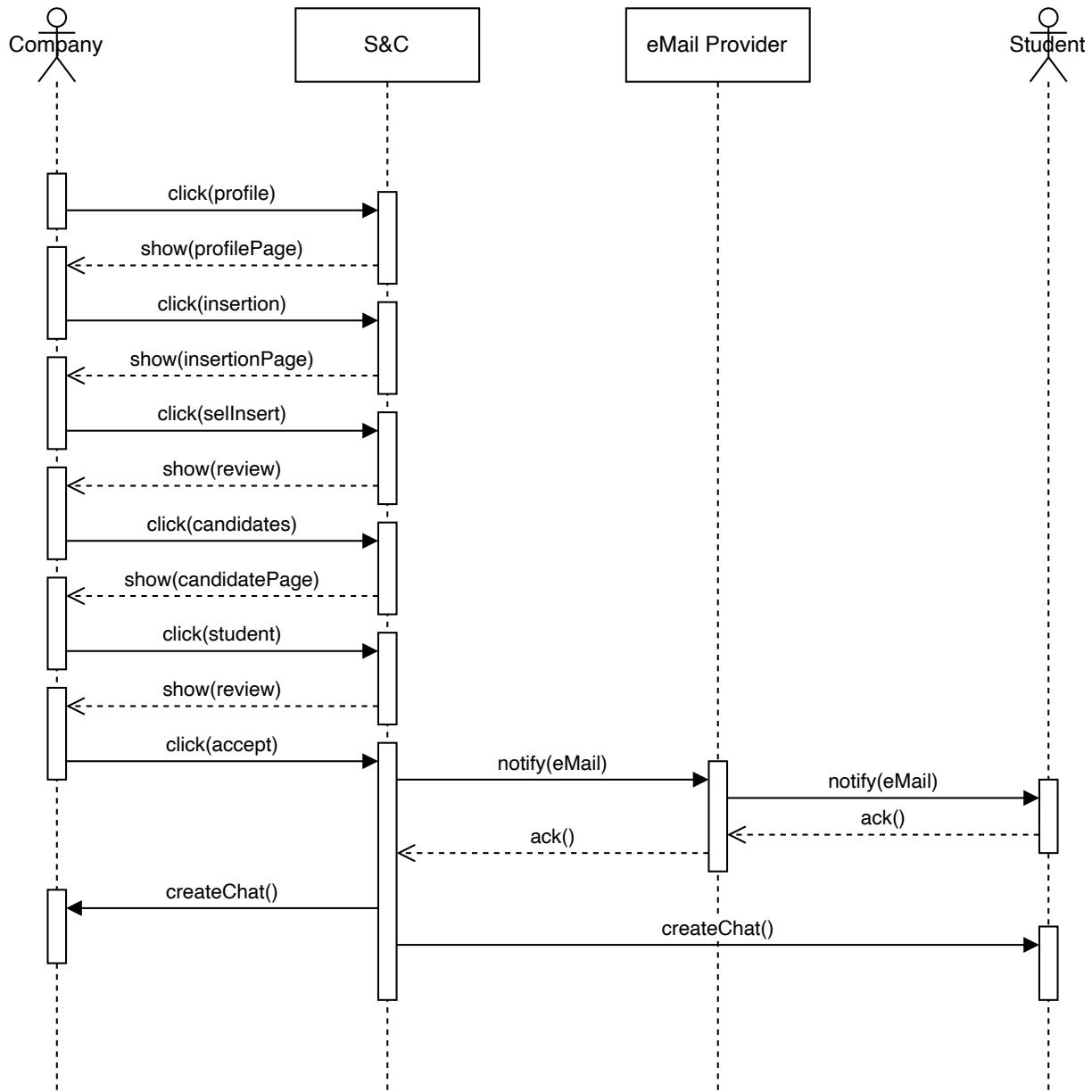


Figure 3.13: Accept a candidate sequence diagram.

## UC9. Submit a Complaint

<b>Name</b>	Submit a Complaint
<b>Actor</b>	S, C, eMail provider
<b>Entry conditions</b>	The S (C) has encountered a problem during the internship period.
<b>Event Flow</b>	<p>1 - The S (C) goes to the chat page.</p> <p>2 - The S (C) clicks on the chat with the selected C (S).</p> <p>3 - The S (C) clicks on the “+” button.</p> <p>4 - The S (C) clicks on “Submit a Complaint”.</p> <p>5 - The system shows the user a form.</p> <p>6 - The S (C) describes the problem in the form.</p> <p>7 - The S (C) clicks on the “Submit” button.</p> <p>8 - The system checks if the description is not empty.</p> <p>9 - If the problem insertion is not empty the system sends a notification and an eMail through the eMail provider to the S's U.</p>
<b>Exit condition</b>	The complaint is submitted.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The user tried to submit an empty form. An error message is shown, inviting the user to describe the problem.</li> </ul>

Table 3.9: Submit a Complaint use case.

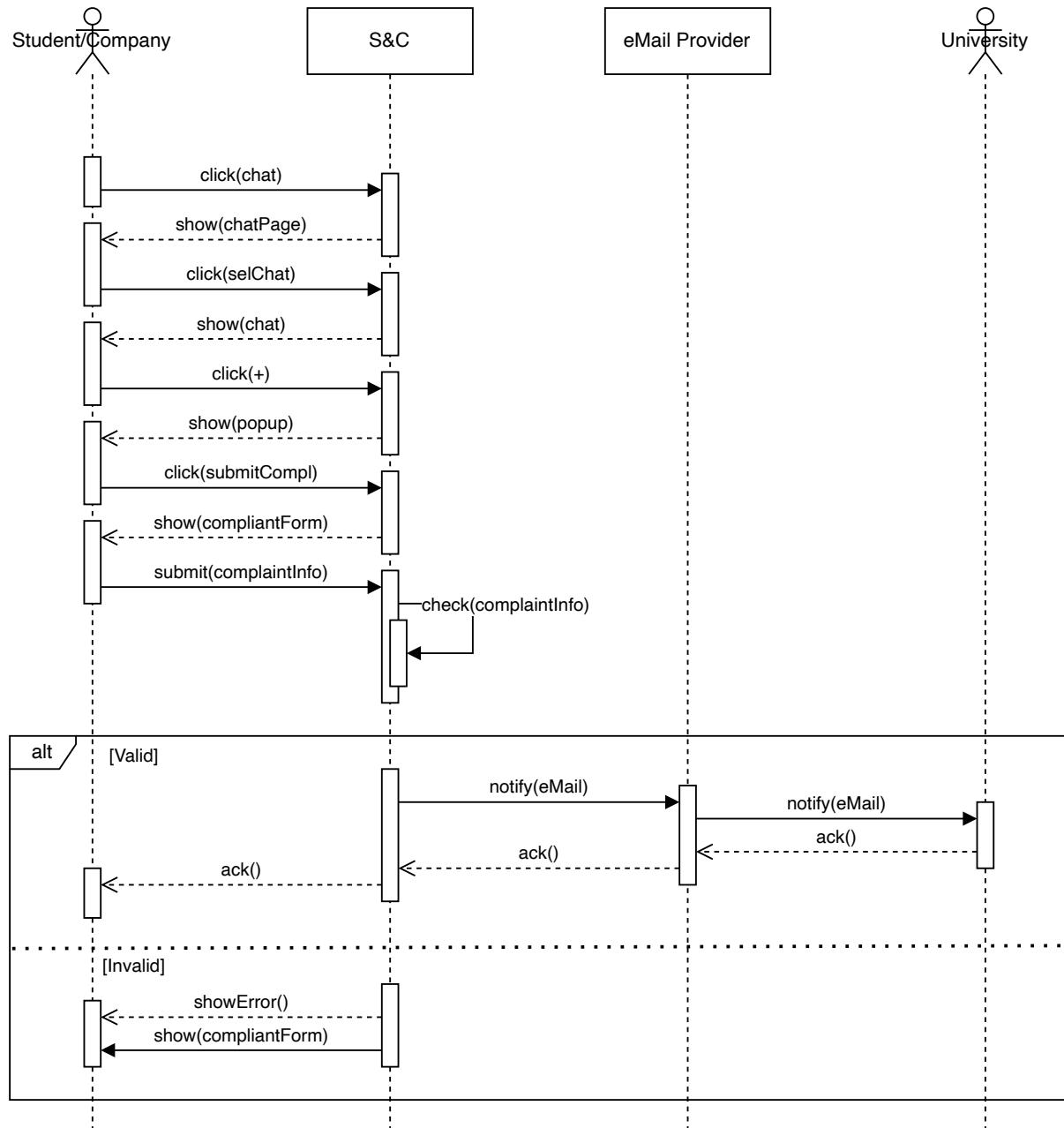


Figure 3.14: Submit a Complaint sequence diagram.

## UC10. Submit Information

<b>Name</b>	Submit Information
<b>Actor</b>	S, C, eMail provider
<b>Entry conditions</b>	The S (C) has some information to report regarding the ongoing internship.
<b>Event Flow</b>	<p>1 - The S (C) goes to the chat page.</p> <p>2 - The S (C) clicks on the chat with the selected C (S).</p> <p>3 - The S (C) clicks on the “+” button.</p> <p>4 - The S (C) clicks on “Submit an Information”.</p> <p>5 - The system shows the user a form.</p> <p>6 - The S (C) fills in the form.</p> <p>7 - The S (C) clicks on the “Submit” button.</p> <p>8 - The system checks if the description is not empty.</p> <p>9 - If the information insertion is not empty, the system sends a notification and an eMail through the eMail provider to the S's U.</p>
<b>Exit condition</b>	The information is submitted.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The user tried to submit an empty form. An error message is shown, inviting the user to add the information in the form.</li> </ul>

Table 3.10: Submit Information use case.

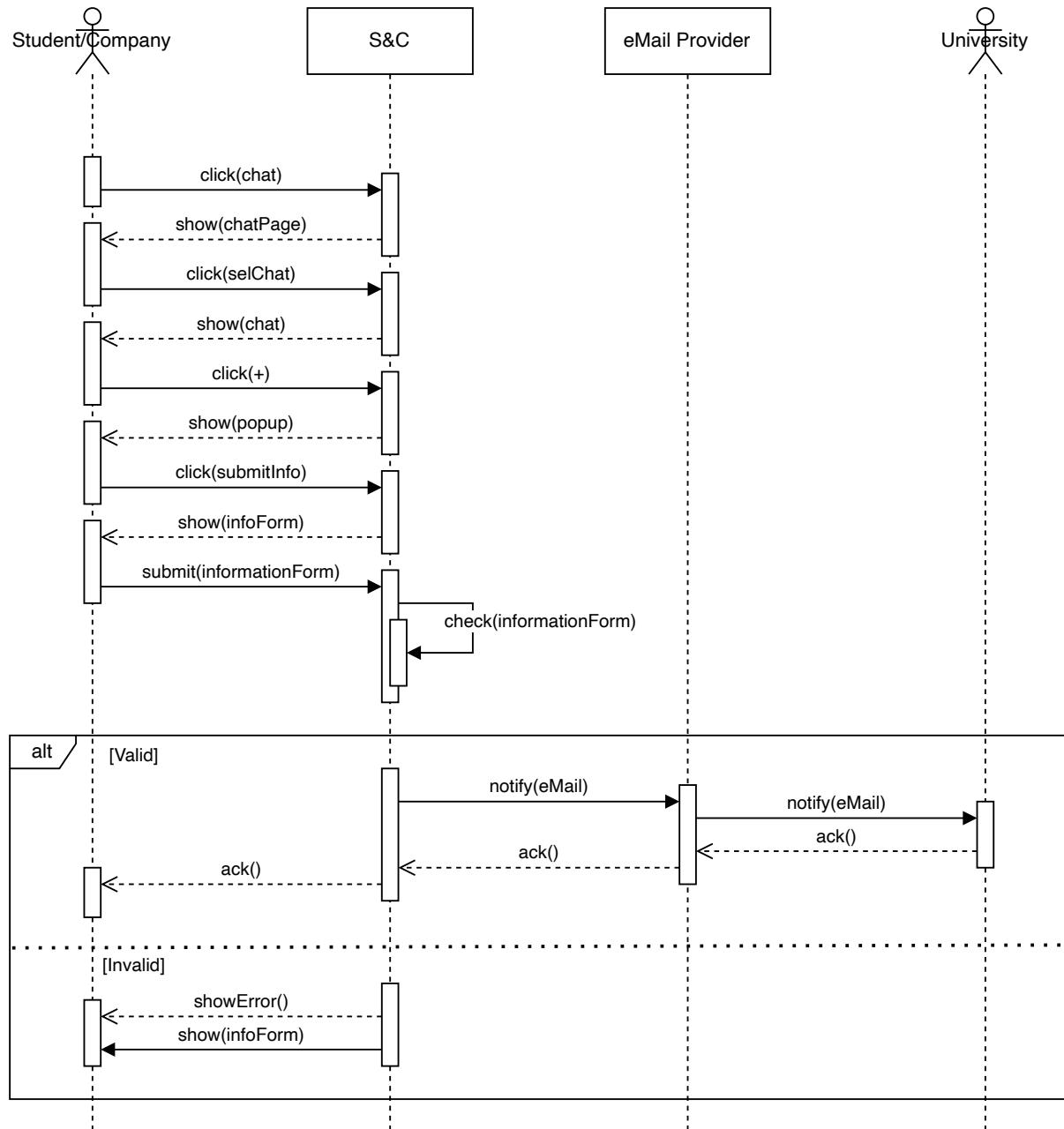


Figure 3.15: Submit Information sequence diagram.

### UC11. Look for a student through recommended ones

<b>Name</b>	Look for a student through recommended ones
<b>Actor</b>	C
<b>Entry conditions</b>	The C wants to look for a S and has an open position for an internship.
<b>Event Flow</b>	<p>1 - The C clicks on the profile page.</p> <p>2 - The C clicks on "Insertion".</p> <p>3 - The C selects the chosen internship posting.</p> <p>4 - The system shows the review of the insertion.</p> <p>5 - The C opens the candidates' section.</p> <p>6 - The system shows the recommended students.</p>
<b>Exit condition</b>	The C can look through all the recommended students for that internship.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• There are no recommended S for that internship posting.</li> </ul>

Table 3.11: Look for a student through recommended ones use case.

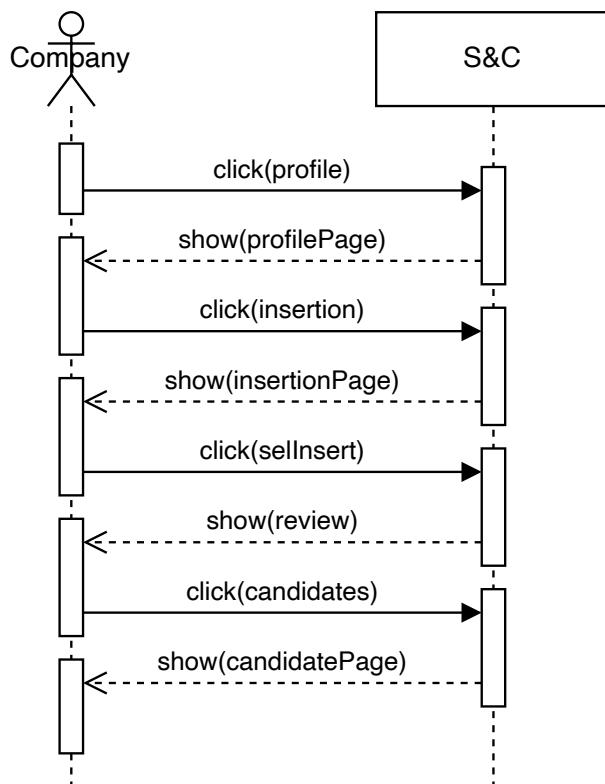


Figure 3.16: Look for a student through recommended ones sequence diagram.

## UC12. Contact a recommended student

<b>Name</b>	Contact a recommended student
<b>Actor</b>	C, S, eMail provider
<b>Entry conditions</b>	The C wants to send the proposal for an internship to a recommended student.
<b>Event Flow</b>	<p>1 - The C clicks on the profile page.</p> <p>2 - The C selects the internship posting for which it wants to find a candidate.</p> <p>3 - The C starts scrolling through the recommended student's profiles.</p> <p>4 - The C clicks on the profile of a recommended S.</p> <p>5 - The C reviews the S's CV.</p> <p>6 - The C contacts the S, by clicking "Contact".</p> <p>7 - The system sends a notification and an eMail through the eMail provider to the S.</p>
<b>Exit condition</b>	The S receives a contact request from the C.
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• There are no recommended students for the internship.</li> </ul>

Table 3.12: Contact a recommended student use case.

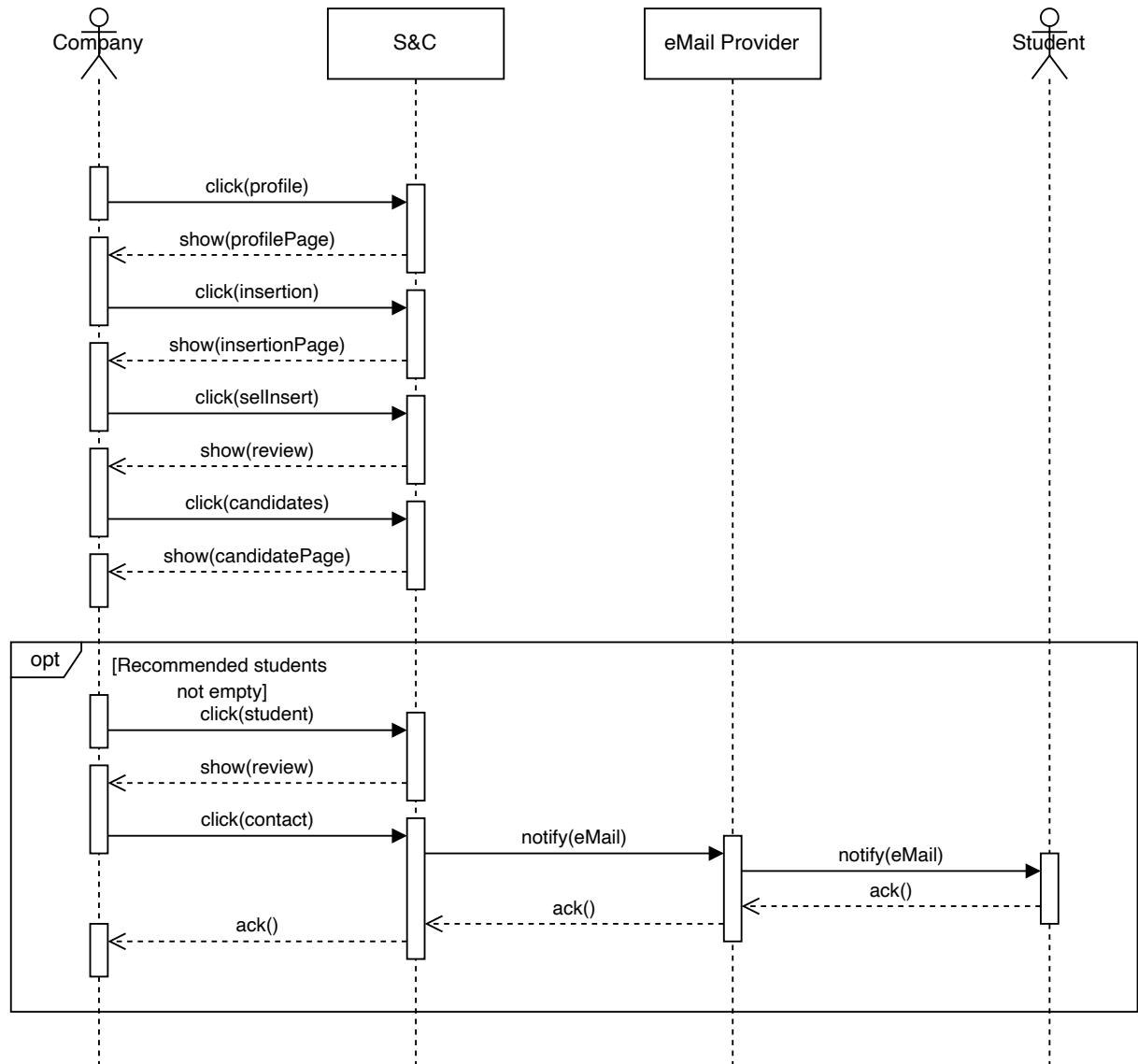


Figure 3.17: Contact a recommended student sequence diagram.

### UC13. Accept a company

<b>Name</b>	Accept a company
<b>Actor</b>	S, C, eMail provider
<b>Entry conditions</b>	The C has contacted the S proposing him an internship position and the S wants to accept it.
<b>Event Flow</b>	<p>1 - The S goes on his notification page.</p> <p>2 - The S scrolls through the list of C that contacted him.</p> <p>3 - The S clicks on the C's profile and reviews the internship description.</p> <p>4 - The S decides to accept the internship by clicking on the "Accept" button.</p> <p>5 - An email is sent to the C through the eMail provider, notifying them that the student accepted their request.</p>
<b>Exit condition</b>	The system creates a chat between the S and the C.

Table 3.13: Accept a company use case.

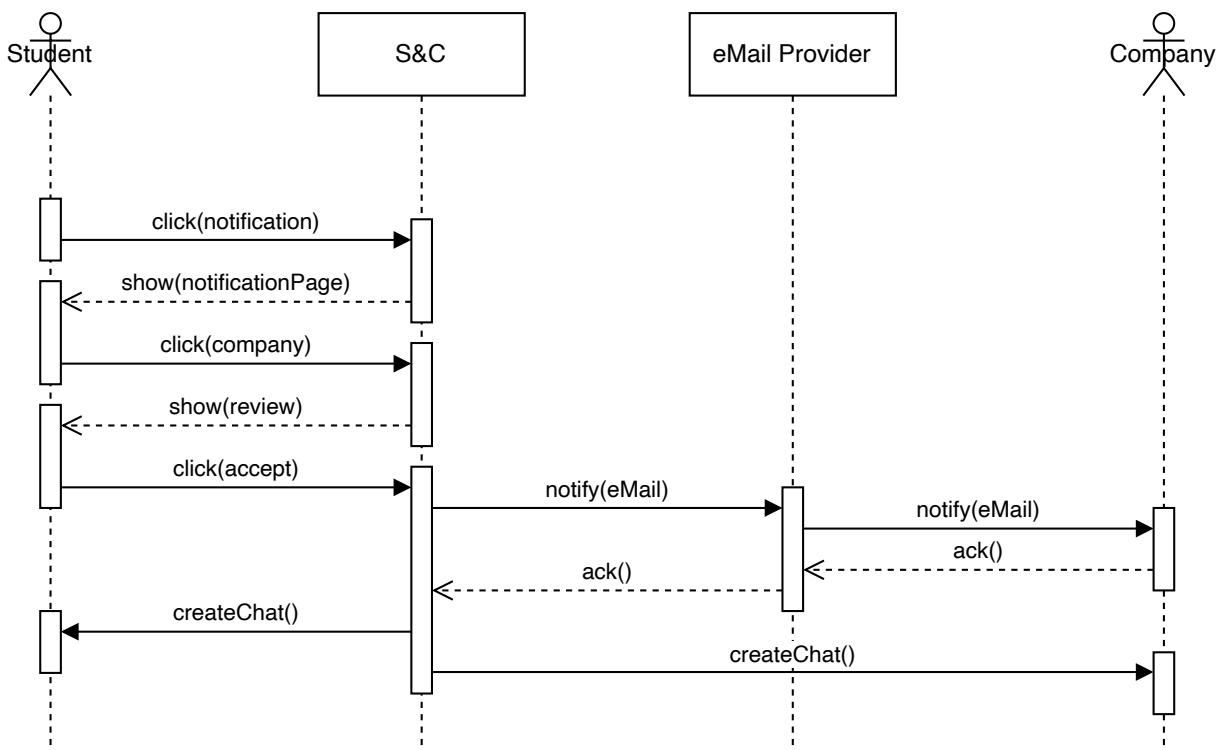


Figure 3.18: Accept a company sequence diagram.

### UC14. Schedule an Interview

<b>Name</b>	Schedule an Interview
<b>Actor</b>	S, C
<b>Entry conditions</b>	The S and the C have accepted each other's request and a chat is established between them.
<b>Event Flow</b>	<p>1 - The C clicks on the “+” button.</p> <p>2 - The C selects “schedule interview”.</p> <p>3 - The C compiles a form selecting a date and time for the interview.</p> <p>4 - The C submits the form and sends it to the S.</p> <p>5 - The S can either accept or decline the proposal by selecting the appropriate option.</p> <p>6 - If the S accepts, the event is added to the S&amp;C calendar for both the S and the C.</p>
<b>Exit condition</b>	An interview event is scheduled and added to both the S's and the C's S&C calendars.
<b>Exceptions</b>	The S declines the interview proposal from the C, and the system cancel the proposal.

Table 3.14: Schedule an Interview use case.

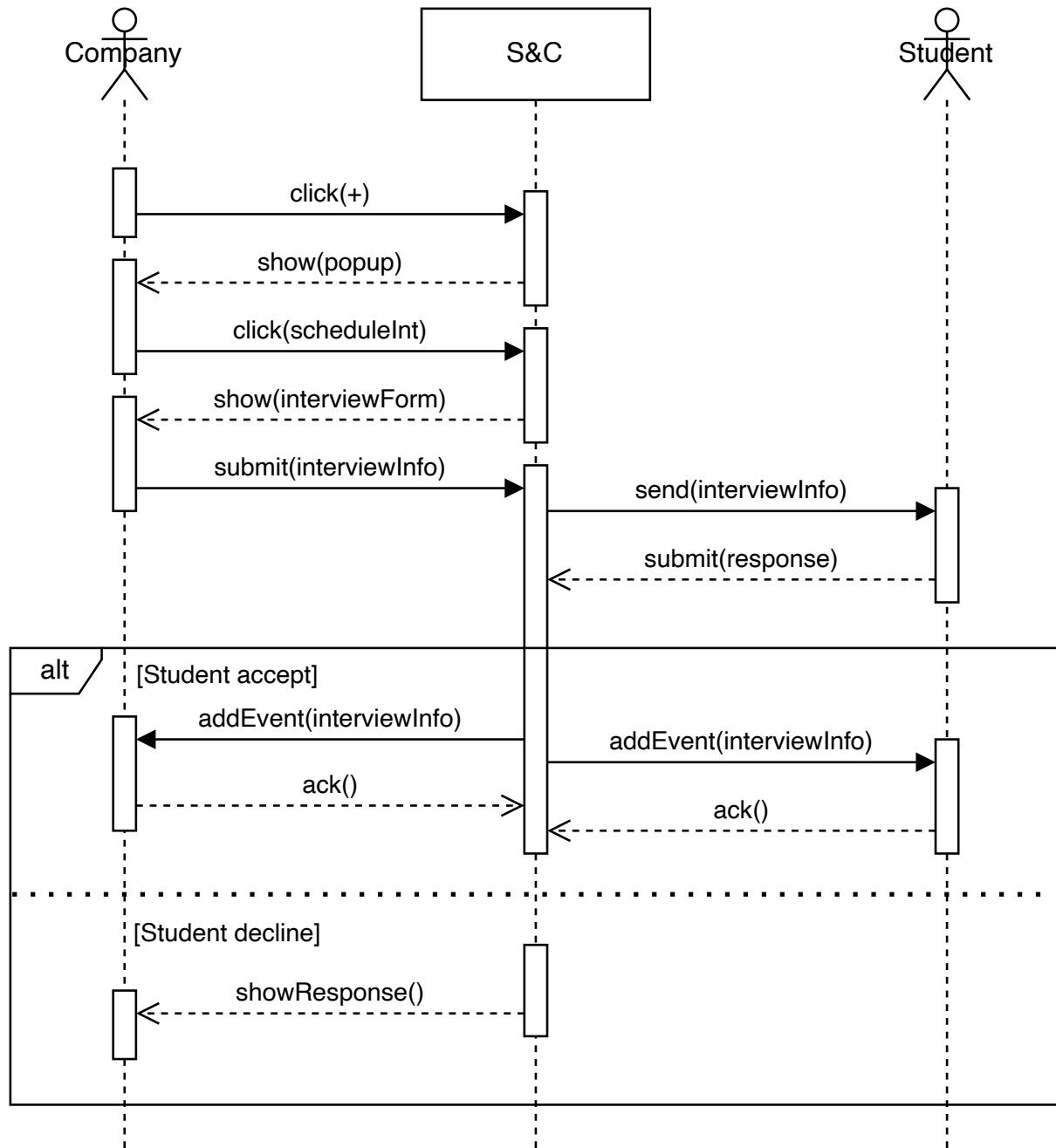


Figure 3.19: Schedule an Interview sequence diagram.

## UC15. Start an Internship

<b>Name</b>	Start an Internship
<b>Actor</b>	S, C
<b>Entry conditions</b>	The S and the C have already completed an interview, and the C is impressed with the S.
<b>Event Flow</b>	<p>1 - The C clicks on the “+” button.</p> <p>2 - The C selects “propose internship”.</p> <p>3 - The C compiles a form selecting the dates of starting and ending of the internship.</p> <p>4 - The C submits the form and sends it to the S.</p> <p>5 - The S accepts the proposal by selecting the appropriate option.</p>
<b>Exit condition</b>	The system adds the event to both the S's and the C's S&C calendars.
<b>Exceptions</b>	The S declines the offer and the chat between S and C is closed.

Table 3.15: Start an Internship use case.

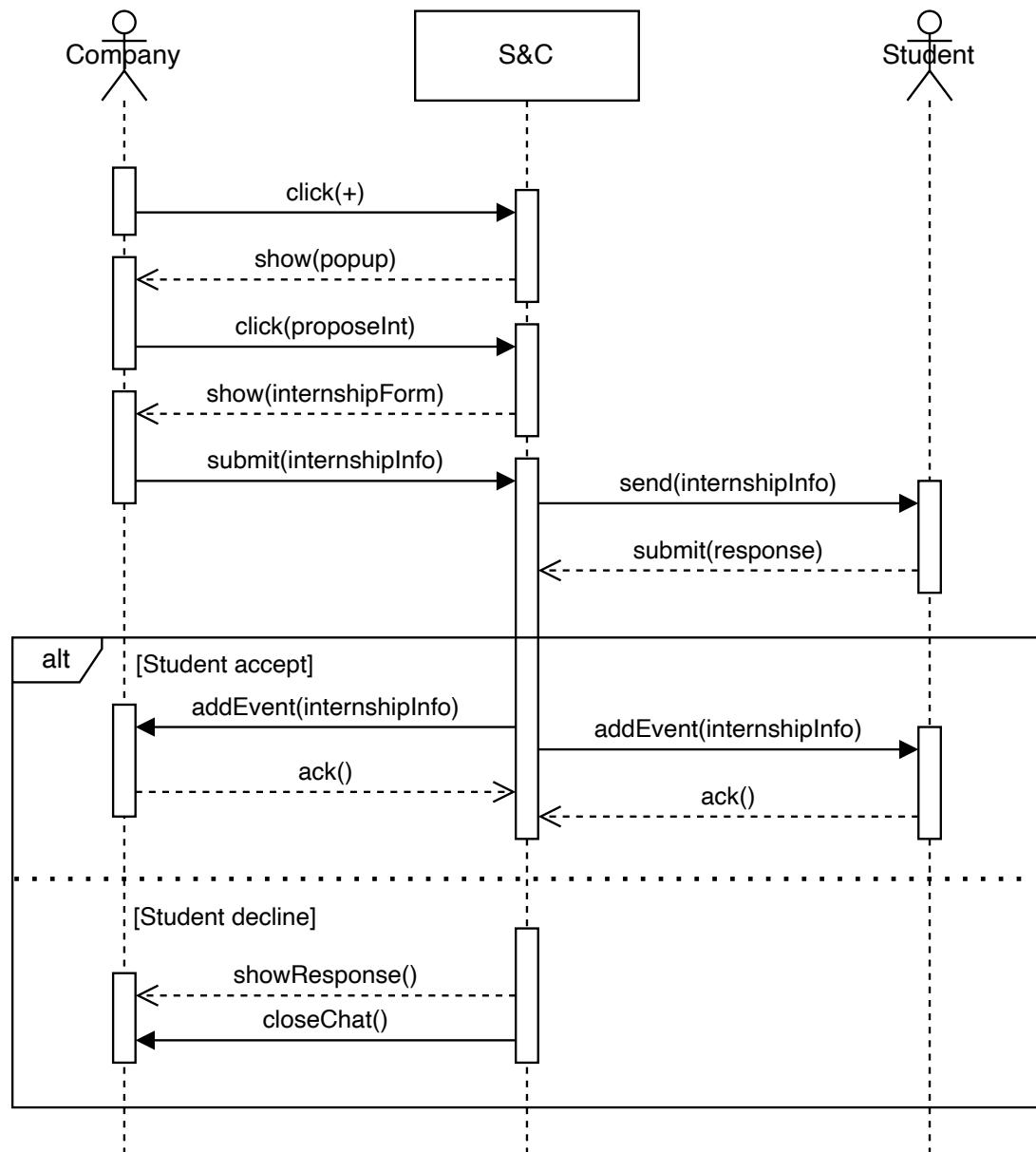


Figure 3.20: Start an Internship sequence diagram.

### UC16. Monitor internship

<b>Name</b>	Monitor internship
<b>Actor</b>	U
<b>Entry conditions</b>	The U monitors the internship between the C and the S.
<b>Event Flow</b>	1 - The U goes on the main page. 2 - The U clicks on the chat regarding the S and the C. 3 - The system shows the chat and the related complaints or information.
<b>Exit condition</b>	The U can monitor the information and issues reported by the other two parties during the internship.

Table 3.16: Monitor Internship use case.

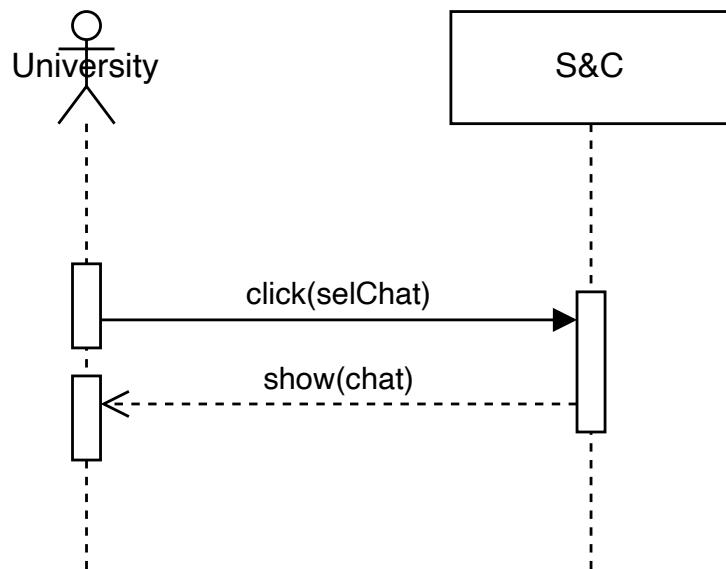


Figure 3.21: Monitor Internship sequence diagram.

## UC17. Interrupt internship

<b>Name</b>	Interrupt internship
<b>Actor</b>	U, S, C, eMail provider
<b>Entry conditions</b>	The U monitors the internship and reads a complaint.
<b>Event Flow</b>	1 - The U goes on the main page. 2 - The U clicks on the related chat. 3 - The U reviews the complaint. 4 - If the U deems it necessary, clicks on "Interrupt Internship". 5 - The system sends an eMail through the eMail provider and notifies the S and C about the interruption of the internship.
<b>Exit condition</b>	The internship is interrupted and the chat closed.

Table 3.17: Interrupt internship use case.

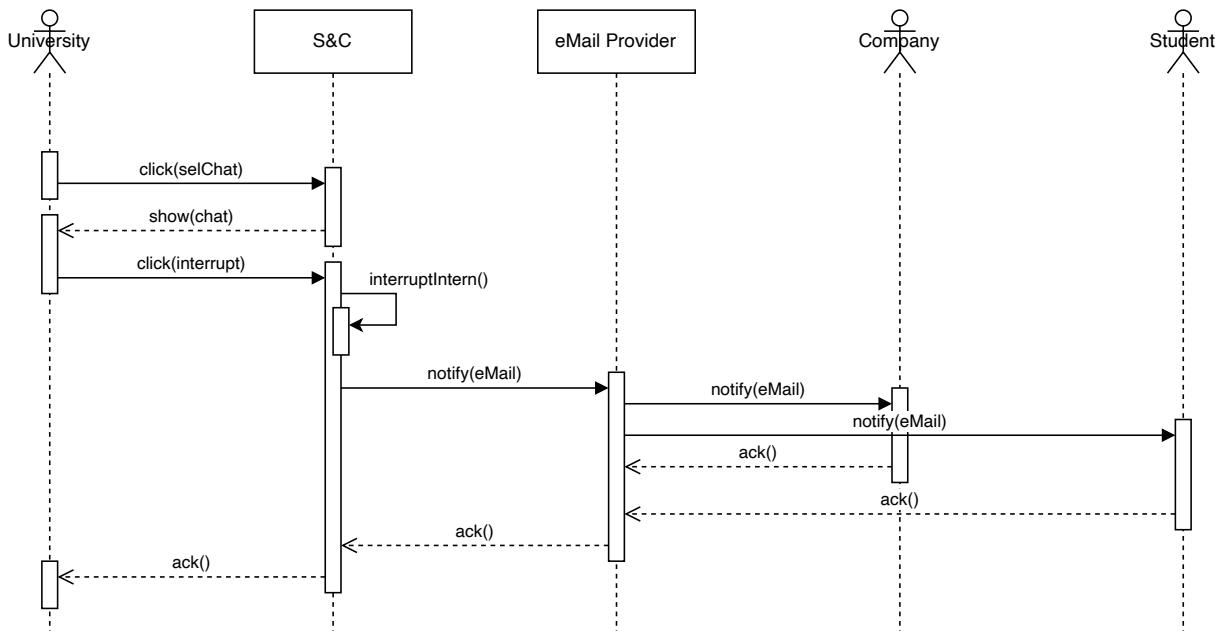


Figure 3.22: Interrupt Internship sequence diagram.

### 3.2.4. Mapping on goals

The mapping links the platform's goals to specific requirements and domain assumptions, ensuring each objective is addressed effectively.

- [G1]: University students would like to seek internships that better align with their interests and field of study.
  - [R1]: The system should allow an unregistered guest to sign up.
  - [R2]: The system should allow registered students to log in.
  - [R3]: The system should allow students to insert their CVs and manage their profile information (e.g. personal data, skills and profile photo).
  - [R4]: The system should provide personalized suggestions to students for enhancing their CVs to improve their chances of being selected.
  - [R5]: The system should notify students when internships matching their skills, experiences, and interests become available.
  - [R6]: The system should allow students to browse available internship postings.
  - [R7]: The system allows S to visualize the profile of other C.
  - [R8]: The system should allow students to apply only for open internships.
  - [R30]: The system should not allow applications for C's internships that are currently ongoing.
  - [D1]: S must be enrolled in a U.
  - [D3]: S should already have a valid Cv to be uploaded.
  - [D4]: S's Cv should be a PDF document.
  - [D5]: The User can easily access his eMail provider.
  - [D6]: S,C provide accurate and truthful information in their profiles, CVs, and internship description.
  - [D9]: The S&C recommendation engine achieves high accuracy when S and C provide detailed, complete, and accurate information in their CVs and job descriptions.

- [D10]: The AI system integrated into the S&C platform effectively enhances students' CVs and companies' internship postings, providing accurate and useful suggestions to improve the matching process.
  - [D11]: The eMail provider functions reliably, ensuring that all emails, including verification and notification messages, are successfully delivered to the intended recipients.
- [G2]: Companies would like to offer internships to students who best match the related figure.
    - [R1]: The system should allow an unregistered guest to sign up.
    - [R18]: The system should allow registered companies to log in.
    - [R19]: The system should allow C to manage their organization profile.
    - [R20]: The system should allow C to post new internship opportunities, specifying details such as required skills, tasks, and benefits.
    - [R21]: The system should provide suggestions to C to improve their internship postings, making them more appealing to S.
    - [R22]: The system allows C to visualize the profile of other Users.
    - [R23]: The system should inform C about the availability of S CVs corresponding to their needs.
    - [R25]: The system allows C to establish a chat with S when mutual interest is identified.
    - [R26]: The system should allow only one worker per internship.
    - [D5]: The User can easily access his eMail provider.
    - [D6]: S,C provide accurate and truthful information in their profiles, CVs, and internship description.
    - [D8]: C on the platform are assumed to be legitimate and offer genuine internships, operating legally and ethically without verification by S&C.
    - [D9]: The S&C recommendation engine achieves high accuracy when S and C provide detailed, complete, and accurate information in their CVs and job descriptions.

- [D10]: The AI system integrated into the S&C platform effectively enhances students' CVs and companies' internship postings, providing accurate and useful suggestions to improve the matching process.
  - [D11]: The eMail provider functions reliably, ensuring that all emails, including verification and notification messages, are successfully delivered to the intended recipients.
- **[G3]: Enhance communication and coordination between students and companies in the selection process.**
    - [R10]: The system allows S to establish a chat with C when mutual interest is identified.
    - [R11]: The system should assist S in scheduling interviews.
    - [R12]: The system should allow S to track the status of their applications and final selections.
    - [R13]: The system should not allow a student already engaged in an internship to apply for other internships.
    - [R16]: The system should allow S to review the S&C Calendar.
    - [R17]: The system should insert the S's events on the S&C Calendar.
    - [R25]: The system allows C to establish a chat with S when mutual interest is identified.
    - [R27]: The system should enable C to track the progress of their internship selection processes, including applications and decisions.
    - [R31]: The system should allow C to review the S&C Calendar.
    - [R32]: The system should insert the C's events on the S&C Calendar.
    - [D12]: Students and companies are committed to the internship process and will engage with features such as scheduling interviews and responding to feedback.
    - [D13]: Companies must clearly communicate interview requirements (e.g., time, format, platform) to students during scheduling.
    - [D14]: Interviews must be conducted either in person or via secure external tools, with students and companies responsible for ensuring their availability and functionality.

- [G4]: Evolve the recommendation system through feedback and suggestions (provided by students and companies) to improve future matches.
  - [R5]: The system should notify students when internships matching their skills, experiences, and interests become available.
  - [R7]: The system allows S to visualize the profile of other C.
  - [R9]: The system should allow S to provide feedback, suggestions and rating to improve the recommendation process.
  - [R22]: The system allows C to visualize the profile of other Users.
  - [R23]: The system should inform C about the availability of S CVs corresponding to their needs.
  - [R24]: The system should allow C to provide feedback and suggestions to improve the recommendation process.
  - [D6]: S,C provide accurate and truthful information in their profiles, CVs, and internship description.
  - [D9]: The S&C recommendation engine achieves high accuracy when S and C provide detailed, complete, and accurate information in their CVs and job descriptions.
  - [D10]: The AI system integrated into the S&C platform effectively enhances students' CVs and companies' internship postings, providing accurate and useful suggestions to improve the matching process.
  - [D12]: Students and companies are committed to the internship process and will engage with features such as scheduling interviews and responding to feedback.
- [G5]: Universities would like to monitor internships and handle any emerging issues promptly, ensuring a smooth and beneficial experience for all parties involved.
  - [R1]: The system should allow an unregistered guest to sign up.
  - [R14]: The system should provide S with a mechanism to report issues or complaints related to ongoing internship they are working on.
  - [R15]: The system allows a maximum of one unresolved complaint per S during an ongoing internship.

- [R28]: The system should allow C to submit complaints or information related to their ongoing internships.
- [R29]: The system allows a maximum of one unresolved complaint per C during an ongoing internship.
- [R33]: The system should allow U to log in.
- [R34]: The system should allow U to monitor internships of their students.
- [R35]: The system should allow U to interrupt ongoing internships of their S, when a compliant was submitted.
- [D2]: The related U should be already registered when the S signs up.
- [D5]: The User can easily access his eMail provider.
- [D7]: U are willing and able to use the S&C platform to monitor internship statuses, address complaints, and respond to student or company feedback as needed.
- [D11]: The eMail provider functions reliably, ensuring that all emails, including verification and notification messages, are successfully delivered to the intended recipients.

Row ID	Goal ID	Req ID	DA ID	Use Case ID
1	G1	R1, R2, R3, R4, R5, R6, R7, R8, R30	D1, D3, D4, D5, D6, D9, D10, D11	UC1, UC2, UC4, UC6, UC7, UC13, UC15
2	G2	R1, R18, R19, R20, R21, R22, R23, R25, R26	D5, D6, D8, D9, D10, D11	UC1, UC2, UC3, UC8, UC11, UC12, UC15
3	G3	R10, R11, R12, R13, R16, R17, R25, R27, R31, R32	D12, D13, D14	UC9, UC10, UC12, UC14
4	G4	R5, R7, R9, R22, R23, R24	D6, D9, D10, D12	UC5
5	G5	R1, R14, R15, R28, R29, R33, R34, R35	D2, D5, D7, D11	UC9, UC16, UC17

Table 3.18: Traceability Matrix

### 3.3. Performance requirements

**Number of Concurrent Users:** The S&C platform must be designed to handle a significant number of concurrent users, ensuring smooth performance even during peak activity periods, such as internship application deadlines. The platform should support a substantial portion of its active user base accessing the system simultaneously.

**Data Storage:** The platform must store and manage all relevant information about students, including CVs, profiles, and application statuses, as well as companies' internship postings, feedback, and selection processes. Historical data, such as completed internships and feedback, should remain accessible to support recommendations and user references.

**Response Time:** Operations directly executed by the S&C platform, such as registration, login, posting internships, applying for internships, and scheduling interviews, must have a response time of under 500 milliseconds. For operations involving external systems, such as email notifications or third-party scheduling tools, the platform will strive to minimize delays but cannot guarantee their performance.

## 3.4. Design constraints

### 3.4.1. Standard compliance

The S&C platform must comply with relevant data protection and privacy regulations, including the EU's General Data Protection Regulation (GDPR). This regulation ensures that the personal data of users, including students, companies, and universities, is collected, stored, and processed securely and transparently. The platform must implement appropriate measures to protect user privacy and ensure data security.

### 3.4.2. Hardware limitations

The primary hardware requirement for accessing the S&C platform is a reliable internet connection and a device with a modern web browser.

## 3.5. Software system attributes

### 3.5.1. Reliability

The S&C platform must be fault-tolerant to ensure continuous service and minimize the impact of potential errors. This will prevent system failures from propagating, allowing users to continue their tasks without significant interruptions.

### 3.5.2. Availability

The S&C platform must maintain a high level of availability. This ensures that the platform is accessible to users at all times.

### 3.5.3. Security

The platform must implement robust security measures to protect user data and ensure secure access. This includes both authentication and authorization:

- Authentication will verify the identity of users attempting to log in, ensuring only legitimate users can access the platform.
- Authorization will manage the permissions of authenticated users, restricting access to specific features based on user roles.

To protect sensitive information, the system will adopt best practices for data security, including:

- Encryption of passwords and personal data stored in the database.
- Protection against query injections and other common security vulnerabilities to prevent unauthorized data access.

### 3.5.4. Maintainability

The S&C platform must be designed using scalable and modular components to allow easy updates and additions of new features with minimal effort. The architecture should support future expansions, such as new functionalities or integrations with third-party services, without major changes to the core system. Regular maintenance should be scheduled during off-peak hours, preferably at night.

### 3.5.5. Portability

The system must be accessible to users from any device or platform that supports a modern web browser. On the server side, there are no specific portability requirements, as the platform will rely on standard web technologies and services to ensure compatibility across different environments.

# 4 | Formal Analysis Using Alloy

In this section is presented the detailed alloy specification for the system, as outlined and described in the RASD.

---

//////////////////*SIGNATURES*//////////////////

```
abstract sig User{
    var submit: lone Complaint
}

sig Internship{
    var candidates: set Student,
    var status: InternshipStatus,
    var interview:set Student
}

enum InternshipStatus {notUploaded,free, ongoing, interrupted}

sig Company extends User{
    var upload: set Internship
}

sig Student extends User{
    university: one University,
    var internship: lone Internship
}

sig University{
    var interrupt:set Internship
}
```

```

sig Complaint{
var referredTo: lone Internship,
var status: ComStatus
}

enum ComStatus {notSubmitted,notHandled,Handled}

-----
/////////////////PREDICATES///////////////
-----

//One possible World
pred show{ }

//In this world we can observe the creation of the internship by a company
pred uploadInternship[c: Company, i: Internship]{
    //pre
    i.status=notUploaded
    //post
    c.upload'=c.upload+i'
    i.status' = free
#Complaint=0
}

//In this world we can observe a student applying for an internship
pred apply[s:Student,i:Internship]{
    //pre
    s not in i.candidates
    s.internship=none
    i.status=free
    //post
    i.^upload'=i.^upload
    i.candidates'=i.candidates+s'
    i.status'=free
    s.internship'=none
#Complaint=0
}

```

```

//In this world we can observe a company scheduling an interview with a candidate
pred scheduleInterview[s:Student, i:Internship]{
  //pre
  s not in i.interview
  s in i.candidates
  s.internship=none
  i.status=free
  //post
  i.~upload'=i.~upload
  i.status'=i.status
  i.interview'=i.interview+s,
  s.internship'=s.internship
  i.candidates'=i.candidates
  #Complaint=0
}

//In this world we can observe a company accepting a candidate after an interview
pred acceptCandidate[s:Student, i:Internship]{
  //pre
  s in i.candidates
  s in i.interview
  //post
  i.~upload'=i.~upload
  i.status'=ongoing
  s.internship'=i
  i.candidates'=none
  i.interview'=none
  #Complaint=0
}

//In this world we can observe a user submitting a complaint about his ongoing
internship
pred submitComplaint[com:Complaint, u:User, i:Internship]{
  //pre
  com.status=notSubmitted
  i in u.internship or i in u.upload
  i.status=ongoing
}

```

```

//post
i.status'=i.status
com.referredTo'=i
com.status'=notHandled
u.submit'=u.submit+com
#Complaint=1
}

//In this world we can observe an university interrupting an internship of one of his students after
//receiving a complaint by one of the interested parts
pred interruptInternship[i:Internship]{
//pre
i.^referredTo!=none
some com:Complaint|i in com.referredTo and com.status=notHandled
i.status=ongoing
//post
i.status'=interrupted
i.^internship.university.interrupt'=i.^internship.university.interrupt+i'
}

//In this world we can observe the evolution of a complaint's status
pred internshipStatusChange[i:Internship]{
i.status=notUploaded; i.status=free; i.status=ongoing; i.status=interrupted
#Internship=1
}

//In this world we can observe the evolution of a internship's status
pred complaintStatusChange[com:Complaint]{
com.status=notSubmitted; com.status=notHandled; com.status=Handled
#Complaint=1
}

```

```

run show
run uploadInternship
run apply
run scheduleInterview
run acceptCandidate
run submitComplaint
run interruptInternship
run internshipStatusChange
run complaintStatusChange

-----FACTS-----
-----INTERNSHIP STATUS-----
//Conditions to be satisfied to be an ongoing internship
fact OnGoing{
  always(all i:Internship|
    i.status=ongoing iff
      i.^internship!=none and
        i.candidates=none and
          i.interview=none)
}

//Conditions to be satisfied to be a NotUploaded internship
fact NotUploaded{
  always(all i:Internship|
    i.status=notUploaded implies
      i.candidates=none
        and i.interview=none
        and i.^internship=none
        and i.^upload=none)
}

//Conditions to be satisfied to be a free internship
fact Free{
  always(all i:Internship|

```

```

        i.status=free implies
            i.^internship=none)
}

//Conditions to be satisfied to be an interrupted internship
fact Interrupted{
always(all i:Internship|
        i.status=interrupted implies
            i.^internship=none and
            i.candidates=none and
            i.interview=none and
            i.^referredTo.status=Handled)
}

-----COMPLAINT STATUS-----
//Conditions to be satisfied to be a NotSubmitted complaint
fact NotSubmitted{
always(all com:Complaint|
        com.status=notSubmitted implies
            com.referredTo=none and
            com.^submit=none)
}

//Conditions to be satisfied to be a NotHandled complaint
fact notHandled{
always(all com:Complaint|
        com.status=notHandled implies
            com.referredTo!=none and
            (com.referredTo in com.^submit.internship or
            com.referredTo in com.^submit.upload) and
            com.^submit!=none
            and
            com.referredTo.status=ongoing)
}

//Conditions to be satisfied to be a Handled complaint
fact Handled{
always(all com:Complaint|
        com.status=Handled implies

```

```

com.referredTo!=none and
  (com.referredTo in com.^submit.internship or
   com.referredTo in com.^submit.upload) and
   com.^submit!=none and
   (com.referredTo.status=ongoing
    or
    com.referredTo.status=interrupted))
}

```

----- GENERAL FACTS -----

//Correlation between upload and internship status

```

fact NoStrangeInternship{
  always(no i:Internship|
    i.status!=notUploaded and
    i.^upload=none)
}

```

//Imposing one complaint per student

```

fact OneComplaintPerStudent{
  always( all s:Student| #(s.submit)<2)
}

```

//A complaint should have at most one submitter

```

fact OnlyOneSubmitter{
  always(all com:Complaint| lone u:User| com in u.submit)
}

```

//An internship should have at most one worker

```

fact OneWorkerPerInternship{
  always(all i:Internship| lone s:Student| i in s.internship)
}

```

//Correlation between internship status and complaint status

```

fact InterruptOnlyHandled{
  always(all i:Internship, com:Complaint |
    i.status=interrupted and
    i in com.referredTo implies
      com.status=Handled)
}

```

```
}
```

```
//If an internship is interrupted so once interruptInternship happened
fact InterruptedOnceInterrupt{
always(all i:Internship|(i in University.interrupt or
                    i.status=interrupted) implies
          once
          interruptInternship[i])
}
```

```
//Correlation between universities and internship status
fact UniInterrupt{
always(all i:Internship|i.^interrupt!=none iff i.status=interrupted)
}
```

```
//An internship can be uploaded by only one company
fact OneCompanyPerInternship{
always(all disj c1,c2: Company| #(c1.upload & c2.upload)<1)
}
```

```
//A student,in order to be interviewed , should be a candidate for that internship
fact ConditionToInterview{
always(all i:Internship| i.interview in i.candidates)
}
```

```
//Correlation between the evolution of internship status and university's interruption
fact Interruption{
always(all i:Internship|
      i.status=ongoing and
      i.status'=interrupted implies
      i' in i.^internship.university.interrupt')
}
```

```
//Condition to satisfy by a company in order to submit
fact ConditionToSubmitC{
always(all com: Complaint, i:Internship, c:Company|
      i in com.referredTo and
      com in c.submit implies
      i in c.upload)
```

```
}
```

```
//Condition to satisfy by a student in order to submit
fact ConditionToSubmitS{
  always(all com: Complaint, i:Internship, s:Student|
    i in com.referredTo and
    com in s.submit implies
    s.internship=i)
}
```

```
//A worker can't be a candidate
```

```
fact WorkerNoCandidate{
  always(all s:Student| s.internship!=none implies s.^candidates=none)
}
```

-----INTERNSHIP STATUS TRANSITION-----

```
fact InternshipStateChange1{
  always(no i:Internship|i.status=notUploaded and
    (i.status'=ongoing or i.status'=interrupted))
}
```

```
fact InternshipStateChange2{
  always(no i:Internship|i.status=free and
    (i.status'=notUploaded or i.status'=interrupted))
}
```

```
fact InternshipStateChange3{
  always(no i:Internship|i.status=ongoing and
    (i.status'=free or i.status'=notUploaded))
}
```

```
fact InternshipStateChange4{
  always(no i:Internship|i.status=interrupted and
    (i.status'=free or i.status'=notUploaded or i.status'=ongoing))
}
```

-----*COMPLAINT STATUS TRANSITION*-----

```

fact ComplaintStateChange1{
  always(no com:Complaint|
    com.status=notSubmitted and
    com.status'=Handled)
}

fact ComplaintStateChange2{
  always(no com:Complaint|
    com.status=notHandled and
    com.status'=notSubmitted)
}

fact ComplaintStateChange2{
  always(no com:Complaint|
    com.status=Handled and
    (com.status'=notHandled or com.status'=notSubmitted))
}

```

-----*PERMANENT CONDITIONS*-----

```

fact SubmitterAlwaysSubmitter{
  all u:User, com:Complaint|
    com in u.submit implies
      always(com in u.submit)
}

fact UploaderAlwaysUploader{
  all c:Company, i:Internship|
    i in c.upload implies
      always(i in c.upload)
}

fact ReferredAlwaysReferred{
  all com:Complaint, i:Internship|
    com.referredTo=i implies
}

```

```
        always(com.referredTo=i)
    }

fact InterruptedAlwaysInterrupted{
    always(all u:University| u.interrupt in u.interrupt')
}

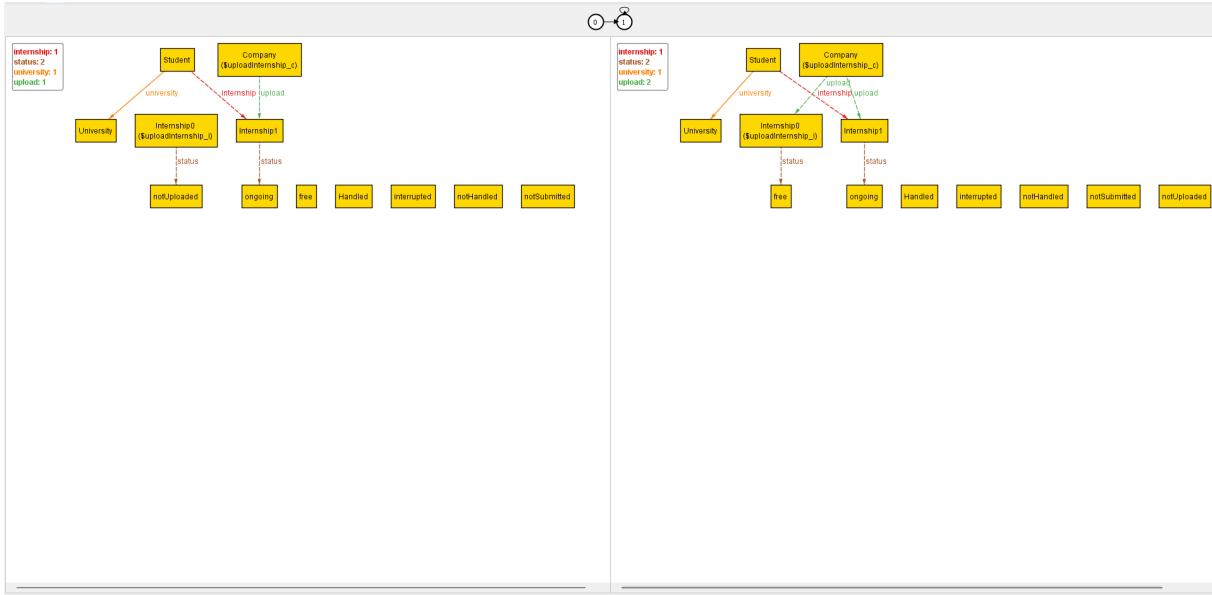
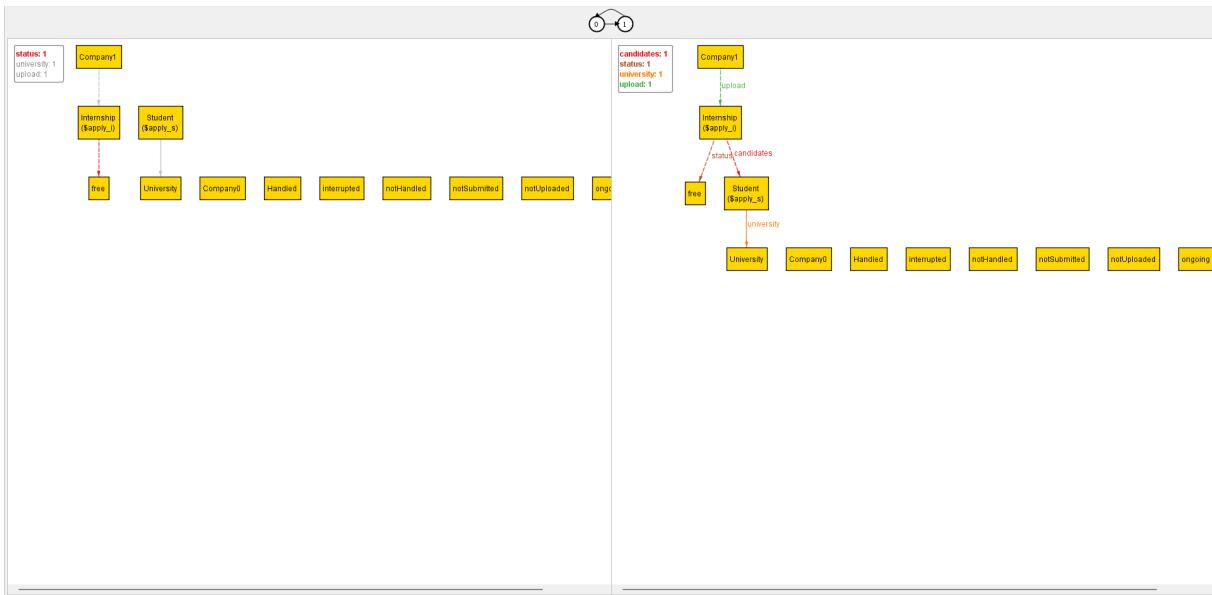
-----
////////////////ASSERTIONS/////////////
-----

assert noComplaintReferredToWrong{
    always(all com:Complaint, u:User|
        (u in com.referredTo.~upload and u in Student implies
            u.internship in com.referredTo) or
        (u in com.referredTo.~upload and u in Company implies
            u.upload in com.referredTo))
    )
}

check noComplaintReferredToWrong

assert OneComplaintOneSubmitter{
    always( no disj u1,u2:User| #(u1.submit & u2.submit)>1)
}
check OneComplaintOneSubmitter

assert OneInternshipOneSubmitter{
    always( no disj c1,c2:Company| #(c1.upload & c2.upload)>1)
}
check OneInternshipOneSubmitter
```

Figure 4.1: Example of the predicate `uploadInternship`.Figure 4.2: Example of the predicate `apply`.

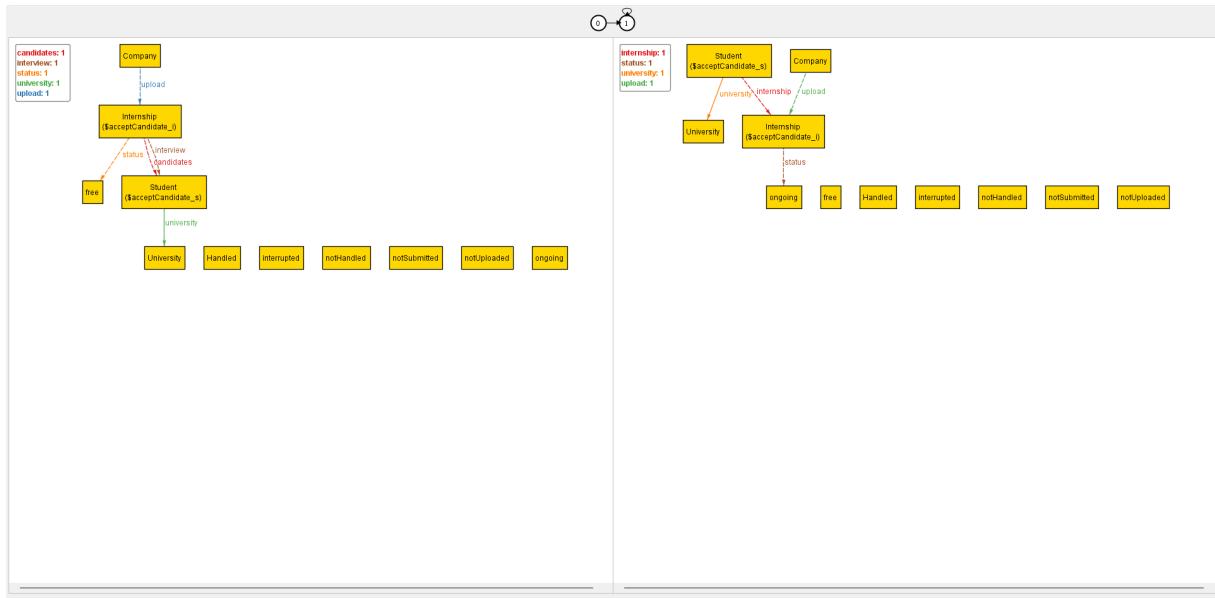


Figure 4.3: Example of the predicate acceptCandidate.

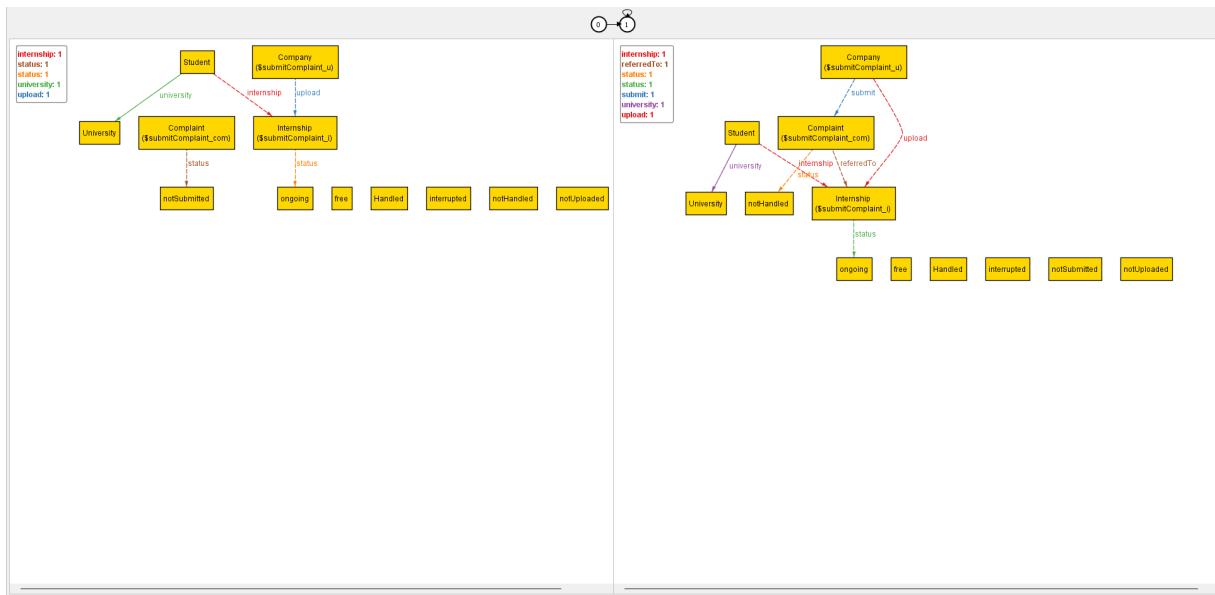


Figure 4.4: Example of the predicate submitComplaint.

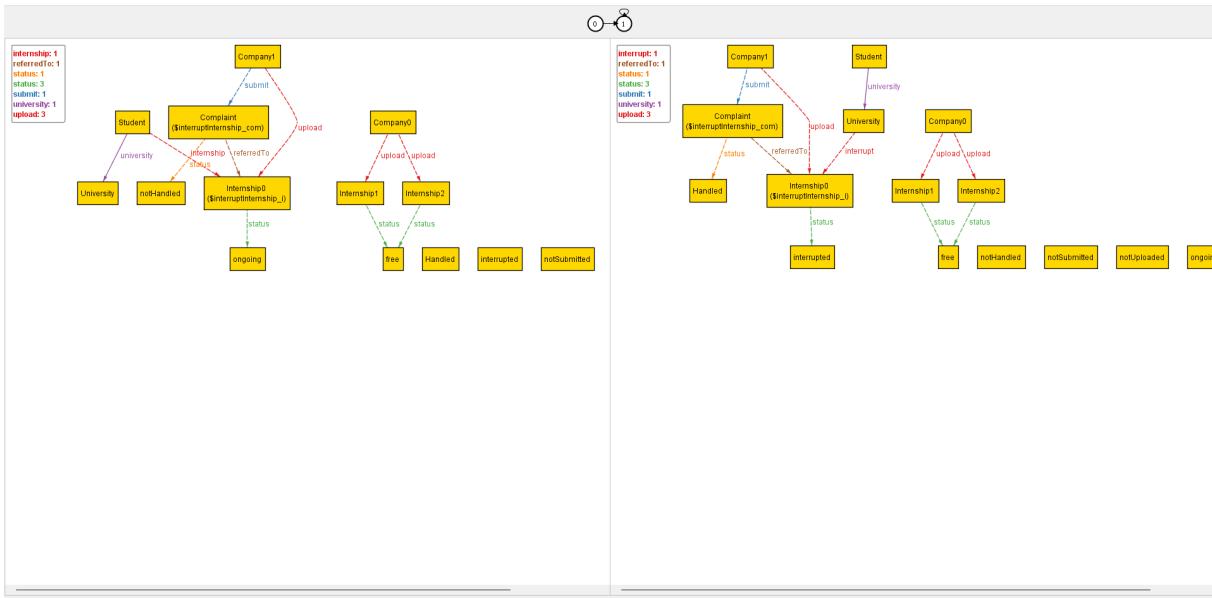


Figure 4.5: Example of the predicate interruptInternship.

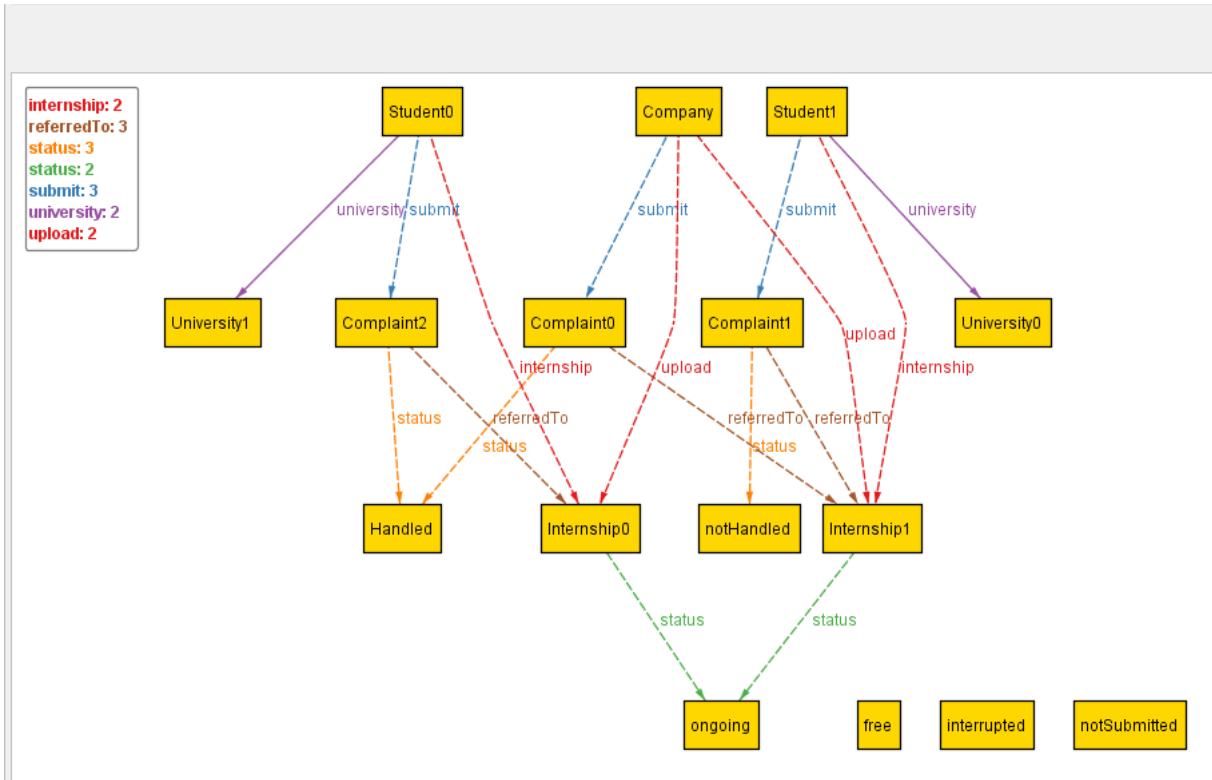


Figure 4.6: Example of a possible world by running predicate Show.

# 5 | Effort Spent

Member of group	Effort spent	
De Matteis Alessandro	Introduction	3h
	Overall description	13h
	Specific requirements	10h
	Formal analysis	20h
	Logic Development	4h
Marino Margherita	Introduction	3h
	Overall description	16h
	Specific requirements	8h
	Formal analysis	20h
	Logic Development	3h
Monaco Giorgio	Introduction	3h
	Overall description	10h
	Specific requirements	25h
	Formal analysis	2h
	Logic Development	10h

Table 5.1: Effort spent by each member of the group.



# 6 | References

## 6.1. References

- 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Requirements engineering.
- A.Y. 2024-2025 Software Engineering 2 Requirement Engineering and Design Project.

## 6.2. Used Tools

- *GitHub* for project versioning and sharing.
- *L<sup>A</sup>T<sub>E</sub>X* and *Overleaf* as editor for writing this document.
- *draw.io* and *Lucidchart* for all the diagrams' design.
- *figma* for the UI's design.
- *Alloy Analyzer 6.1.0* for formal analysis.
- *Google Documents* and *Notion* for collaborative writing, notes and reasoning.



# List of Figures

2.1	UML Class Diagram.	10
2.2	Sign Up State Diagram.	11
2.3	Log In State Diagram.	12
2.4	Enhance CV State Diagram.	12
2.5	Apply for an Internship State Diagram.	13
2.6	Require Feedback State Diagram.	13
2.7	Insert and Enhance Internship State Diagram.	14
2.8	Selection and Schedule Interview State Diagram.	15
2.9	Finalizing Process State Diagram.	15
2.10	Report Problem State Diagram.	16
2.11	Interrupt Internship State Diagram.	16
3.1	Access Page UI.	22
3.2	Student Homepage UI.	22
3.3	Chat Page UI.	23
3.4	Notifications Page UI.	24
3.5	Use Cases Diagram for all the Users.	28
3.6	Sign Up sequence diagram.	30
3.7	Login sequence diagram.	32
3.8	Insert an enhanced Internship Insertion sequence diagram.	34
3.9	Enhance CV sequence diagram.	36
3.10	Submit a feedback sequence diagram.	38
3.11	Search for a company sequence diagram.	39
3.12	Apply for an internship sequence diagram.	40
3.13	Accept a candidate sequence diagram.	42
3.14	Submit a Complaint sequence diagram.	44
3.15	Submit Information sequence diagram.	46
3.16	Look for a student through recommended ones sequence diagram.	47
3.17	Contact a recommended student sequence diagram.	49
3.18	Accept a company sequence diagram.	50

3.19 Schedule an Interview sequence diagram. . . . .	52
3.20 Start an Internship sequence diagram. . . . .	54
3.21 Monitor Internship sequence diagram. . . . .	55
3.22 Interrupt Internship sequence diagram. . . . .	56
4.1 Example of the predicate uploadInternship. . . . .	76
4.2 Example of the predicate apply. . . . .	76
4.3 Example of the predicate acceptCandidate. . . . .	77
4.4 Example of the predicate submitComplaint. . . . .	77
4.5 Example of the predicate interruptInternship. . . . .	78
4.6 Example of a possible world by running predicate Show. . . . .	78

# List of Tables

3.1	Sign Up use case.	29
3.2	Login use case.	31
3.3	Insert an enhanced Internship Insertion use case.	33
3.4	Enhance CV use case.	35
3.5	Submit a feedback use case.	37
3.6	Search for a company use case.	38
3.7	Apply for an internship use case.	40
3.8	Accept a candidate use case.	41
3.9	Submit a Complaint use case.	43
3.10	Submit Information use case.	45
3.11	Look for a student through recommended ones use case.	47
3.12	Contact a recommended student use case.	48
3.13	Accept a company use case.	50
3.14	Schedule an Interview use case.	51
3.15	Start an Internship use case.	53
3.16	Monitor Internship use case.	55
3.17	Interrupt internship use case.	56
3.18	Traceability Matrix	62
5.1	Effort spent by each member of the group.	79

