POLITECNICO
MILANO 1863

Software Engineering 2 - Computer Science and Engineering

Requirements Analysis and Specification Document

Students&Companies

Authors:
Alessandro Vitobello - 10710941
Jurij Diego Scandola - 10709931
Francesco Raimondi - 10767901

Academic Year:
2024 - 2025

Index

1 Introduction

1.1 Purposes

A university student's journey is typically marked by intensive study, often focused exclusively on theoretical knowledge. This leads many students to pursue internships, allowing them to: step outside the academic environment, gain insights of the working world and alleviate fears of being unprepared for it. In this context, Students&Companies provides a valuable platform, offering support both to students and to companies, facilitating connections by matching students' skills and interests with companies' opportunities and projects. It also helps universities to monitor their students' internships.

1.1.1 Student purposes

Students will use the platform to upload their CV (curriculum vitae) in order to find internships that best suit their interests and strengths.

1.1.2 Companies purposes

Companies will use the platform to promote the internships they offer to the students, in order to enroll them in their own working or training programs.

1.1.3 Universities purposes

Universities are in charge of monitoring internships' situations. They are responsible for handling complaints, especially those that might require the interruption of the internship.

1.2 Goals

[G1] Students can find the most suitable internship for them.

- (G1.1) Students can pro-actively search for internship opportunities.
- (G1.2) Students obtain "recommendations" from the platform, about internships they may like.

[G2] Companies can promote their internships, posting them on the platform.

[G3] Companies can select the most suitable candidates.

[G4] Universities can manage their students' internships.

1.3 Scope

Students&Companies (S&C) is designed to address the specific needs of companies, students, and universities. Through the platform, companies can post internship opportunities, including the project's scope, required tasks, relevant technologies used and specific terms. Meanwhile, students can upload their CVs to showcase their skills, competencies, and experiences. Using this information, S&C matches students with suitable internships, notifying them when relevant opportunities arise and, simultaneously, informing companies of available candidates who align with their requirements (this process is referred as "recommendation"). Anyway, a student could also search for opportunities himself: i.e., searching through the published internships using a keyword. When a promising match is identified and accepted by both parties, S&C facilitates the following steps supporting the selection process by managing interviews and selections' logistics. To further enhance engagement, the platform provides tailored recommendations to both companies and students on how to improve their postings and applications for optimal appeal. Additionally, S&C enables universities to manage any concerns that might necessitate the early termination of an internship, such as complaints regarding the internship conditions or a student's conduct.

1.3.1 Shared Phenomena (Students)

Shared Phenomena	**Description**	**Controlled by**	**Observed by**
S1	Students upload their CV and skills on the platform.	W	M
S2	Students use the platform, proactively searching for internship opportunities.	W	M
S3	The platform sends notifications to students regarding a new recommendation found by the application itself.	M	W
S4	Students accept a recommendation.	W	M

S5	Students are interviewed by the company after the match.	W	M
S6	Students provide feedback about the internship and the accuracy of the recommendation.	W	M
S7	S&C suggest students how to improve their CV submission.	M	W
S8	The platform helps students to keep track of the internship matchmaking.	M	W
S9	Students complain or provide information on their ongoing internship through the platform.	W	M

1.3.2 World Phenomena (Students)

World Phenomena	Description
W1	Students work in a company, gaining experience.

1.3.3 Shared Phenomena (Companies)

Shared Phenomena	Description	Controlled by	Observed by
S1	Companies upload the internships, and the relative terms, they offer.	W	M
S2	Companies use the platform to find students that can do the work they need.	W	M

S3	The platform sends notifications to companies regarding new recommendation found by the application itself.	M	W
S4	Companies accept a recommendation.	W	M
S5	Companies interview students whose recommendation was accepted.	W	M
S6	Companies provide feedback and suggestions about the accuracy of the recommendations.	W	M
S7	S&C suggests companies how to make their internships' advises appealing.	M	W
S8	The platform helps companies to keep track of the internship matchmaking.	M	W
S9	Companies complain or provide information on the accepted students' ongoing internship through the platform	W	M

1.3.4 World Phenomena (Companies)

Shared Phenomena	Description
W1	Companies create a job internship spot.

1.3.5 Shared Phenomena (Universities)

Shared Phenomena	Description	Controlled by	Observed by
S1	S&C helps universities to monitor their students' internships.	M	W
S2	Universities manage their students and companies' complaints, regarding their students' internships.	W	M

1.3.6 Machine Phenomena

Machine Phenomena	Description
M1	The platform's system finds relevant matchings between students and companies' profiles.
M2	The platform collects and analyze relevant data in order to improve the accuracy of its algorithm.

1.4 Definitions, Acronyms and Abbreviations

1.4.1 Definitions

Recommendation: used to refer to the mechanism used by the system to inform students that a possible interesting internship became available and to inform companies about the availability of a student's CV matching their needs.

Internship: a practical training opportunity offered by companies to students who want to get a glimpse of the working world, which may include benefits and/or compensation.

Platform: refers to S&C's online information system that facilitates the connection between students and companies to create internship opportunities.

Student: a user of the S&C platform who is looking for an internship and provides personal information and skills in the form of a resume (CV).

Company: an organization that uses the S&C platform to post internship opportunities and evaluate candidates. The company account is managed by a representative.

Match: the result of the recommendation process in which a potential connection between a student and an internship is identified.

Selection process: the selection procedure that begins after the initial contact, in which the company interviews the student to assess their suitability for the internship.

Feedback: qualitative or quantitative information provided by students and companies regarding their experience with the platform, the selection processes, or the internship.

University: an educational institution that uses S&C to monitor the internship progress of its students and handle any complaints or issues. The university account is managed by a representative.

Monitoring mechanisms: platform functionalities that allow users to track the status and progress of selection processes, internships, and interactions between the involved parties.

User: we refer to user as any of the S&C actors between Student, University and Company.

1.4.2 Acronyms

S&C: stands for Student&Companies, the platform's name.

CV: curriculum vitae; a written document where all the educational and professional experiences of an individual and his skills are listed (sometimes including also his personal interests).

HR: human resources.

UI: user interface.

DB: database.

1.5 Revision history

- Version 1.0: 22/12/2024
- Version 2.0: 07/01/2025 (added 1 Company requirement, fixed Student UC diagram, fixed last part of Alloy model)
- Version 3.0: 07/01/2025

1.6 Referenced Documents

- CreatingRASD.pdf (WeBeep.polimi.it)

1.7 Document Structure

This document is composed of a total of six sections, detailed below.
In the first section the project purposes are introduced and made clear through the goals' section. Additionally the scope of the project is specified along with the various phenomena occurring extracted thanks to a world analysis. Lastly, some useful information about the abbreviations and acronyms used in the document are presented.
Section two contains an overall description of the system. At first, it is possible to observe some scenarios to help the reader understand how the application will work. After, there is the domain class diagram with product functions description and process charts. Lastly the domain assumption we made are stated after the user characteristics descriptions.
In section three, the first part include the requirements of the system: these includes external interfaces requirements, functional requirements and, in the end, non-functional requirements. The most rich part is the one on functional requirements that include: the use cases with their accompanying sequence diagrams; it also contains mappings of functional requirements to the system's goals.
Section four contains a formal analysis of our model made using Alloy.
In section five a table with the team efforts is presented.
Section six contains the references used in the document.

2 Overall Description

2.1 Product perspective

2.1.1 Student registers on the S&C platform

Andrea is a student who's looking for an internship to put in practice what he has been learning at university. While he is browsing on his university's website, he finds out S&C announcement and decides to register on its platform through the registration form. The platform requests some mandatory information such as email, username and password. After registering, he decides to upload his CV in order to view more tailored results through the specific button in his dashboard.

2.1.2 Company registers on S&C

BlueCon, a tech start-up, is about to launch a new project and it needs interns to work on it. The son of the CFO attends the university and has heard of S&C platform: during a dinner, the CFO talks about the new upcoming project to his son and the difficulties to advertise it. As the son hears the problem, he quickly reminds about S&C and shares what he read about it. After the dinner, the CFO browse for S&C website, and once he finds out how powerful the platform is, he decides to register the company creating a company's account providing contact's information (such as the e-mail), a username and a password.

2.1.3 University registers on S&C

Politecnico di Milano, a top-tier Italian university, has been offering its students the possibility to take an internship experience for years, in order to help them to get closer to the working world. After receiving the proposal to collaborate from a S&C manager, the Rector decide to stipulate a contract in order to make things easier for both the university and his students in their internships. A Polimi's delegate can now register the university's official account in order to follow the situation of all its students and help them in case of any problem.

2.1.4 Company uploads an internship spot

ChipCorp, a growing software company, wants to hire an intern for their development team. A company's representative logs into the S&C platform and navigates to the "Post an Internship" section. They fill out a form describing the internship position, including details such as required skills, project tasks, duration, and compensation. Once completed, they submit the listing, and it becomes available for students to be visualized.

2.1.5 Student searches for internships

Mark, a computer science student, is looking for an internship in software engineering. He logs into S&C, opens the "Search Internships" page, and filters options based on "Software Development," "Remote," and "Paid." The platform displays a list of relevant positions, allowing Mark to look or apply to opportunities that match his criteria.

2.1.6 Student receives internship recommendations

Since uploading his CV, Leo receives emails about personalized internship recommendations. He logs into S&C and finds these recommendations on his dashboard: the platform suggests three internships that align closely with Leo's skills and interests in data analysis and machine learning. He clicks on each recommendation to learn more about it.

2.1.7 Student applies for an internship

Giulia has found a project management internship with a company that interests her, searching through the platform. She clicks "Apply" on the internship listing: since she already uploaded her CV, the application is submitted directly to the company's S&C dashboard for review.

2.1.8 Company receives student recommendations

GreenBuild&Co. posts an internship announcement, looking for an environmental engineering student. Shortly after, they receive notifications with recommended student profiles based on their listing criteria. The list includes students with relevant skills and experience in the environmental field. GreenBuild Co. can review these profiles to decide who they might want match, trying to invite them for an interview.

2.1.9 Student receives response on a submitted application

Sam recently applied for an internship at HealthTech Inc. After a few days, he receives a notification from S&C informing him that HealthTech Inc. reviewed his application and answered him. Sam opens the message and sees that, although they were impressed with his experience, they chose another candidate but encouraged him to apply again in the future.

2.1.10 Company manages applications

EcoEnergy Ltd. has received several applications for their renewable energy internship. The hiring manager logs into S&C, where applications are listed by

date and percentage of matches (calculated by the platform algorithm). Using the platform's tools they can "Accept" or "Reject" the different applications: once the manager clicked on one of these 2 buttons, a notification is sent to the applying student.

2.1.11 Company initiates the selection process

After reviewing the applications, CityInnovate selects four candidates for the next stage. They use S&C to send interview invitations to each candidate, coordinate schedules, and provide details on the interview format. Candidates receive email notifications and have the possibility to confirm their availability through the platform.

2.1.12 Student and company conduct an interview

Jordan has been invited to an interview with SoftSolutions for a UX design internship he applied 2 weeks ago. He logs into S&C and, after filling out a questionnaire proposed by the company on the platform, he joins a video interview at the scheduled time directly through the platform.

2.1.13 Student and company finalize the internship

After a successful interview, FreshStart Inc. communicates that Mia was selected for the internship she applied for. Mia receives the official proposal through a notification on the S&C platform, where she can review the terms and confirm her acceptance. Upon acceptance, both Mia and FreshStart receive confirmation, and the internship status is updated to "Finalized" on S&C.

2.1.14 University monitors and handles a problem for its student

The university of Chicago has several students that are on an internship in various companies. One of its student, Alex, has recently reported that the ABC group, where he is currently working as an intern, has forced him to work until midnight for two weeks. He has asked to reduce his time shift, but the project leader is not willing to listen to Alex requests. At this point, Alex has thought that he can only ask support to his university filling up a complaint through the specific form on his "Ongoing Interniship" section. As soon as his internship referent receives the complaint by Alex, she uses the S&C tool to formally interrupt the experience.

2.1.15 After internship S&C asks for feedback

Frank is a student who recently completed his internship at DAINOI company. Shortly afterward, he received an email from the S&C platform prompting him

to log in and leave feedback about his experience. Similarly, DAINOI received a notification from S&C, requesting that they provide feedback on Frank's performance and overall experience. Both sets of feedback include multiple questions, designed to gather extensive data that can be used to enhance the platform's matching algorithm and improve future internship recommendations for both students and companies.

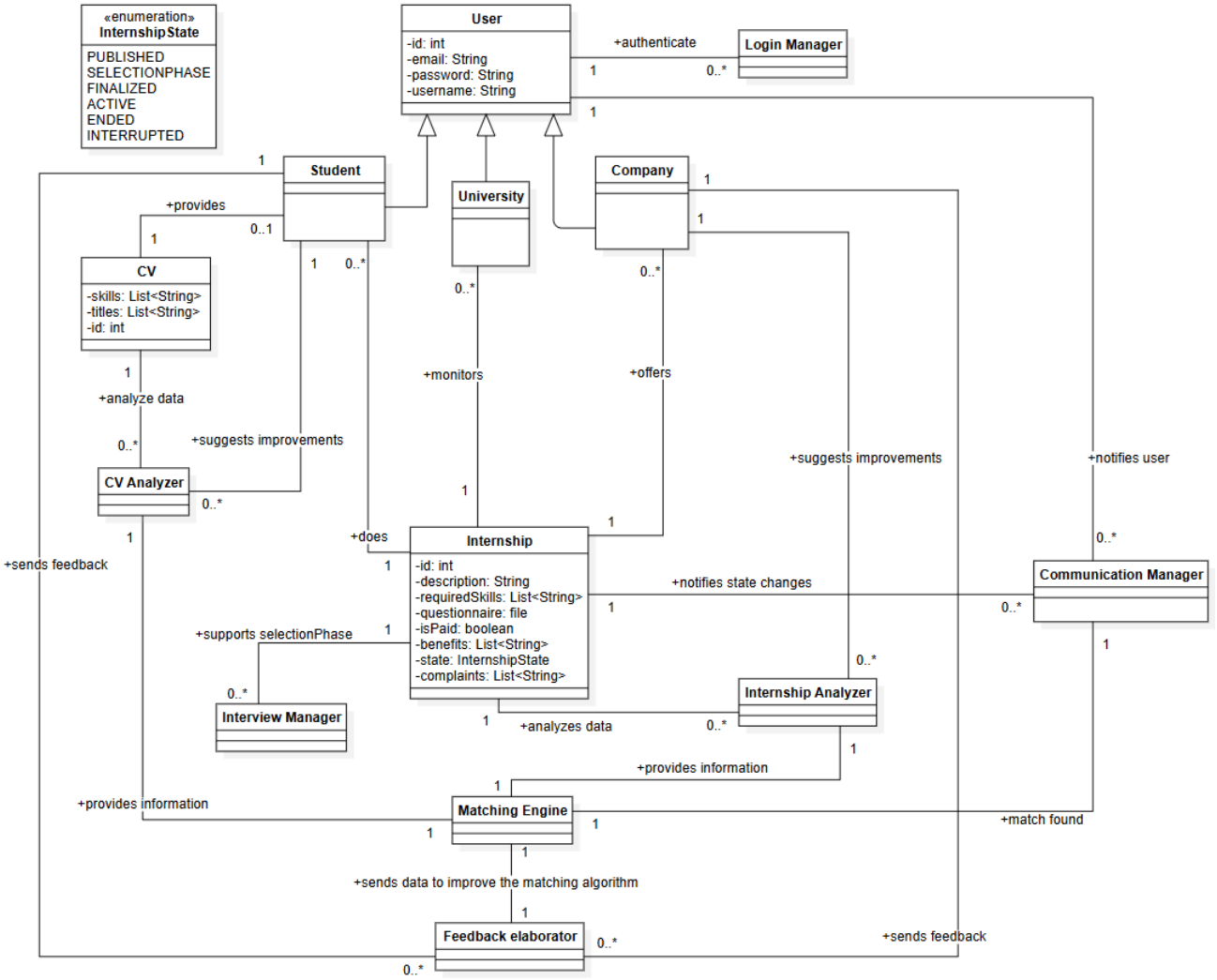
2.1.16 S&C suggest a Student how to make his CV more catchy

Luca, a Geology student, has been applying to different internships for weeks but was never selected by any company and he also received very little recommendations that he didn't like. The problem is how he composed his CV, with it appearing too long and missing his soft-skills' list. The system analyzed it when Luca uploaded the CV, after registering, and instantly suggested Luca on how he should have changed it but he ignored the advice.

2.1.17 The system updates recommendations

Based on feedback from students and companies, the system refines its recommendation algorithms to improve future matches. This ongoing optimization enhances the relevance and accuracy of future internship suggestions.

2.2 Domain Class diagram



In the figure above it is possible to observe the domain class diagram. Student, University and Company are represented as separate entities, though they are in hierarchy with the User entity because of the similarity of the selected attributes. Every **Student** can upload whenever he desires his CV (also updating it, substituting the previous instance) containing his skills and his study titles. When a CV is uploaded in the system, the CV Analyzer examines it in search of possible improvements (e.g. spelling and format check, missing fields...) and, if it finds any, the Student is notified in order for him to decide to correct these things.
Every **Company** can upload a new available Internship post whenever it wants, providing important information such as a description, the required skills and the list of benefits for the intern. A Company can have multiple published internships. As for the CV, also when uploading an Internship, the Internship Analyzer helps companies to produce the best possible post in order to match the best candidates, suggesting possible improvements. Both the CV and Internship Analyzers are related to the Matching Engine because they base their suggestions on how a match is found by the matching algorithm.

Every **University** can monitor its students' internships and manage complaints by students or companies, eventually formally interrupting the internship experience.
The main core of the domain model is the **Internship** entity, connecting all the main components of our domain. Each internship post is identified by an Id and has a "state" that changes:

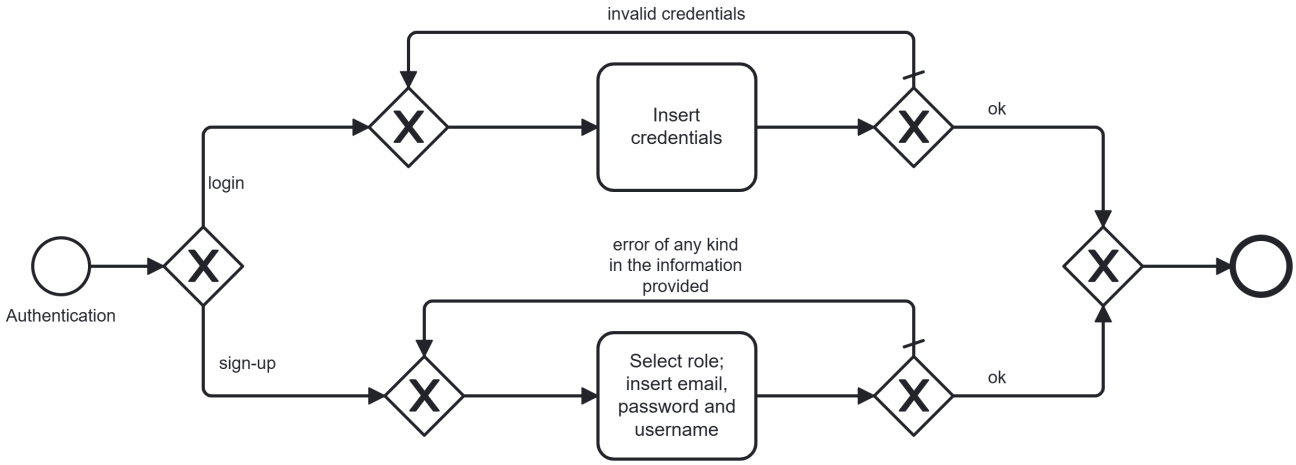
- **PUBLISHED**: means that the Internship is available on the platform and looking for students
- **SELECTIONPHASE**: indicating that students can no longer apply because the Company is conducting the interview process (helped by the Interview Manager)
- **FINALIZED**: means that a suitable candidate has been identified
- **ACTIVE**: the student is carrying on his internship
- **ENDED**: the student has gracefully terminated his experience
- **INTERRUPTED**: the internship, for any reason, has been interrupted when it was not actually finished yet

The **Matching Engine** compares internships' required skills and students' skills to find suitable match. When a match is identified, the **Communication Manager** processes the relevant information and sends notifications to the involved parties. This component also manages general communication, such as internship's state changes, ensuring that information is effectively distributed across the platform's components.
The **Feedback Elaborator** collect, refines and processes data to enhance its utility for the Matching Engine. It filters out inappropriate comments and irrelevant information, ensuring quality input for analysis.
The **Login Manager** is necessary to manage the sign-in and sign-up procedure, granting authentication only to verified users.

2.3 Product functions

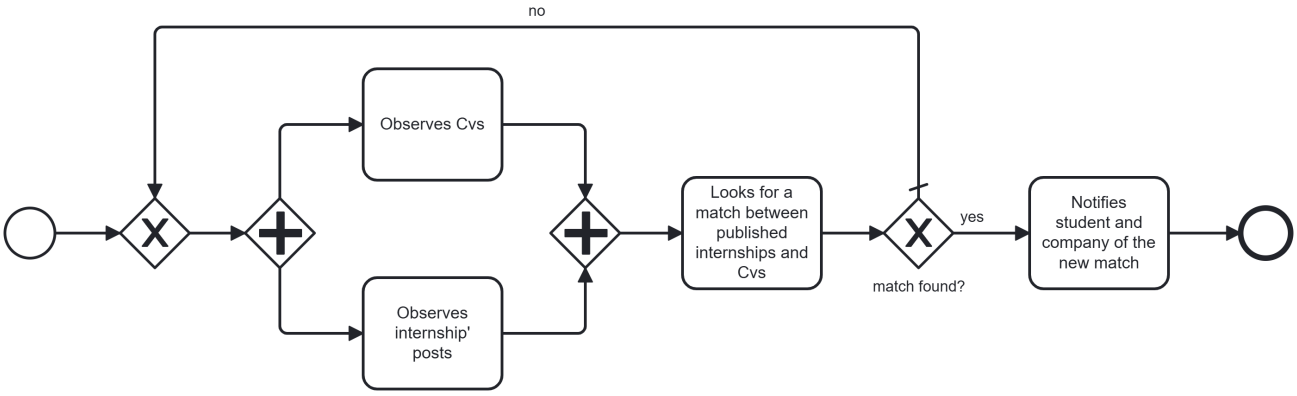
In this section, we show the most important product functions that characterize the S&C's platform (some less relevant interactions are ignored).

2.3.1 Sign in and sign up process



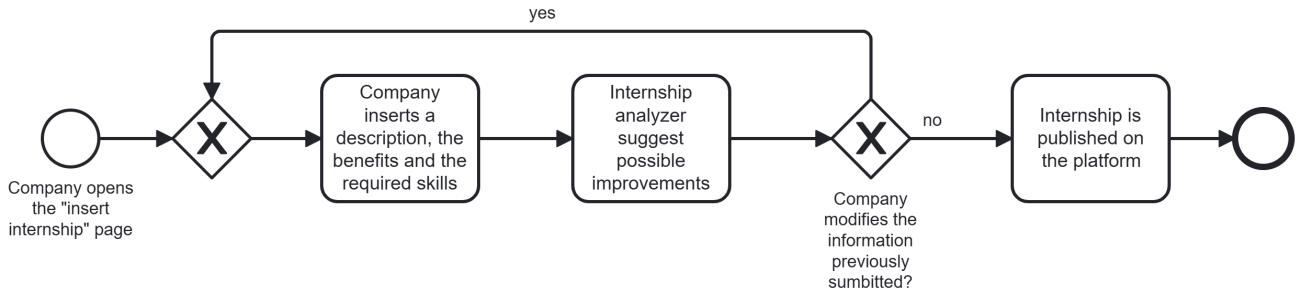
These functions allow users (both unregistered and registered) to access the S&C's platform. The access is based on the role selected while registering (Student, Company or University). Users must provide a valid email address, a password and a username (that will be unique) to authenticate.

2.3.2 Matching process



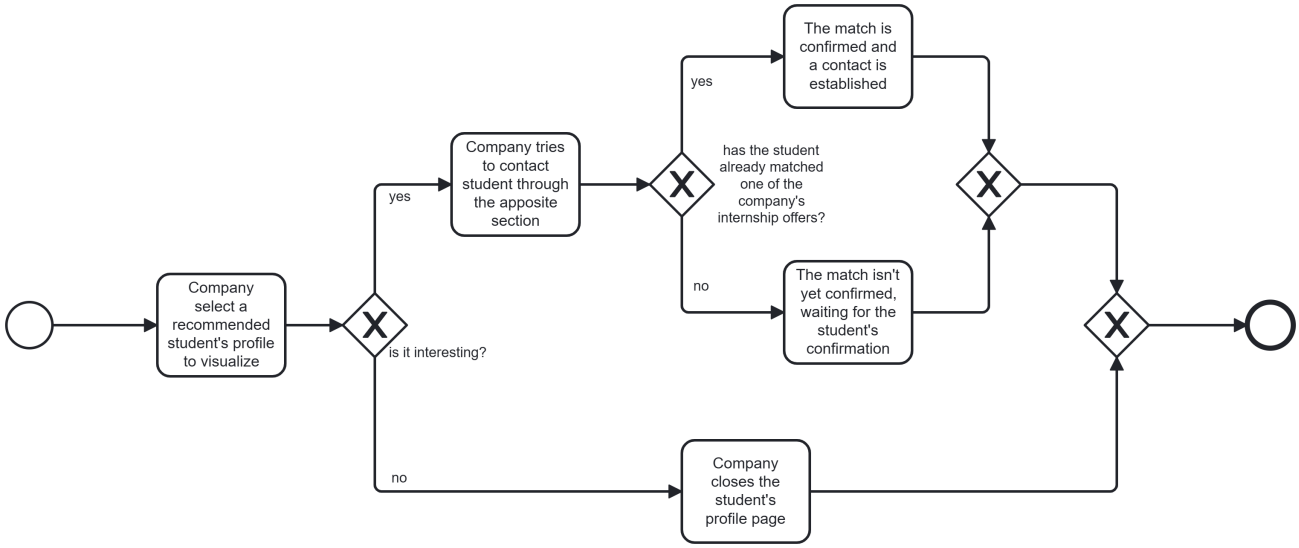
Here you can see the diagram describing the process that runs in the platform's "matching engine". Continually, it observes all the published (and not expired) internship' posts and the students' CVs, looking for a possible match. If found, the platform notifies the interested company and student. Otherwise it keeps searching for other matches.

2.3.3 Company creates an internship post



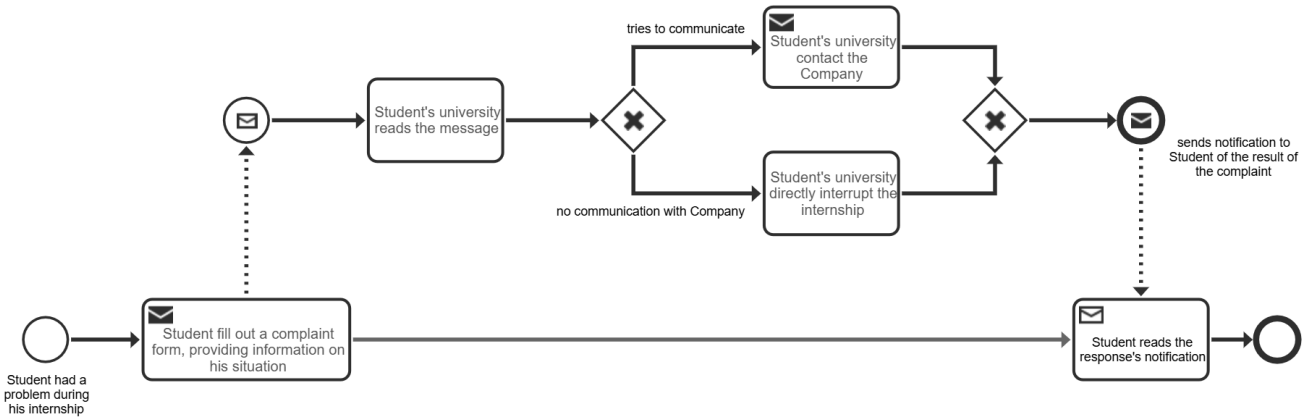
This function allows companies to insert in the platform new internship announcements. When creating a post, the company inserts the required data, such as: a description, the required skills, the intern's benefits and a questionnaire (in file format) for the applying students. After this phase, the "internship analyzer" checks for possible improvements in the insertion (e.g. spelling check, phrases construction...) and suggest them to the user that can decide to modify the post or proceed as it was. Then the announcement is published on the platform.

2.3.4 Company select a recommended student and send him a message



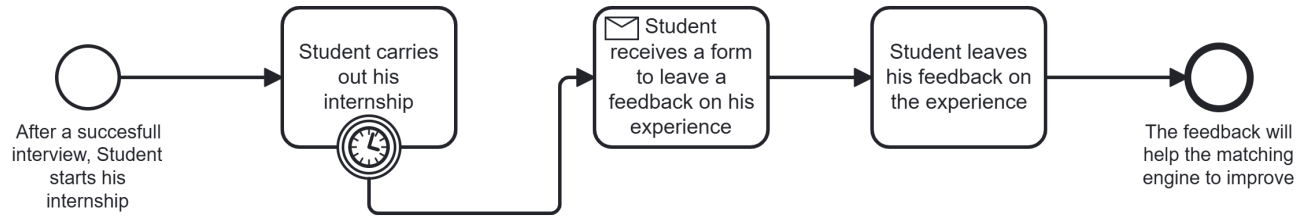
While scrolling through its dashboard, a company finds an interesting student profile that was recommended by S&C. The platform allows the company to open the student's profile page to better understand his skills and experiences. If the company finds the student interesting, tries to establish a contact: this is effectively established only if the student has already matched the company on one of its internships offers. Otherwise, the platform communicates the student the interest of the specific company, waiting him to match (if he wants to).

2.3.5 Student asks for his internship's interruption



If a student has any sort of problem during the duration of his internship he can contact, whenever he wants, his university through the platform filling out a complaint form. The university receiving the message can try to communicate with the company to solve the student's issue, letting him to continue his experience, or, if the problem signaled by the student is considered not fixable, it officially interrupts the student's internship. In the end of this process, the student receives a notification about his complaint's response.

2.3.6 Student leaves feedback



When an internship experience terminates, both the student and the interested company are asked to fill out a form in order to collect data to improve the matching algorithm. In this diagram it is possible to observe the student's process of leaving a feedback: he will point out which were the best things during his experience, if he had any sort of problem, if everything was as he expected to be.

2.4 User characteristics

2.4.1 Unregistered user

An unregistered user is only able to sign up to the platform either as Student, Company or University.

2.4.2 Student

After registering as Student, the user can mainly manage his profile by uploading his/her CV, whenever there is a change; look for available internships and apply to them; chat with matched companies and his/her university; provide feedback at the end of the internship; monitor the situation of the ongoing internship with the possibility to fill in a form with information or complaints.

2.4.3 Company

Once registered as a Company, the user (who is a HR representative) can create and post available internship positions; accept or reject applications from students; set up interviews; chat with students and universities; leave feedback when internships are concluded; monitor ongoing internships with the possibility to complain or provide any type of information through the platform.

2.4.4 University

After registering as University, the user (who is a university's delegate) is able to monitor its students' internships and chat with them and the company if needed to solve any sort of problems and, eventually, interrupt the specific student's intern experience.

2.5 Assumptions, dependencies and constraints

2.5.1 Domain assumptions

[D1] The users' and S&C E-Mail providers are assumed to work correctly.

[D2] It is expected that at least some students and companies fill out the feedback form to help improving the matching algorithm.

[D3] At least some students are interested in pursuing an internship.

[D4] At least some companies wants to find students for their internship.

[D5] Information in CVs is up to date and true.

[D6] Information in internship posts is true.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The system must provide intuitive and user-friendly graphical interfaces that ensure a smooth user experience across both desktop and mobile devices. The system will cater to its three main types of users: students, companies, and universities. Each of these users has their specific needs:

- **Students**: The interface will display a personalized dashboard where students can view internship recommendations, important notifications, and their application history. In the search section, students can filter available internships using criteria such as keywords, location, work mode, and salary.
- **Companies**: Companies will have access to tools for posting and managing internship opportunities, along with advanced filters to examine the profiles of potential candidates. A communication feature will also be available to interact directly with students and in particular selected candidates.
- **Universities**: Universities will be able to monitor the status of their students' internships, intervene in case of issues, and analyze aggregated reports regarding the students' experiences.

See the Design Document for the User Interfaces mock-ups.

3.1.2 Hardware Interfaces

The system requires the support of the following hardware components:

1. **End Users**:
   - Mobile devices and desktops with access to a stable internet connection.
   - Optional GPS modules for location-based features, such as showing internships near the user.
2. **Server Infrastructure**:
   - The server infrastructure must ensure high availability and scalability to manage simultaneous user access.
   - Automatic backups are required to ensure data recovery in case of failure.

3.1.3 Software Interfaces

To ensure the proper functioning of the system, the following software interfaces will be needed:

- **Relational Database**: The system will use a relational database, managed by a DataBase Management System, to store user data, CVs, internship proposals, and feedback. All communications with the database will be carried out through connections secured with TLS.
- **External APIs**: Integration with external APIs will allow the system to send email and push notifications (e.g., via SendGrid, Amazon SES) and to access mapping services for displaying internship locations (e.g., Google Maps).
- **Matching Algorithms**: The internal algorithms will analyze students' CVs and companies' internship announcements to suggest personalized opportunities.
- **University Integration**: The system will support synchronization with external university databases to verify students' identities and track the status of their internships.

3.1.4 Communication Interfaces

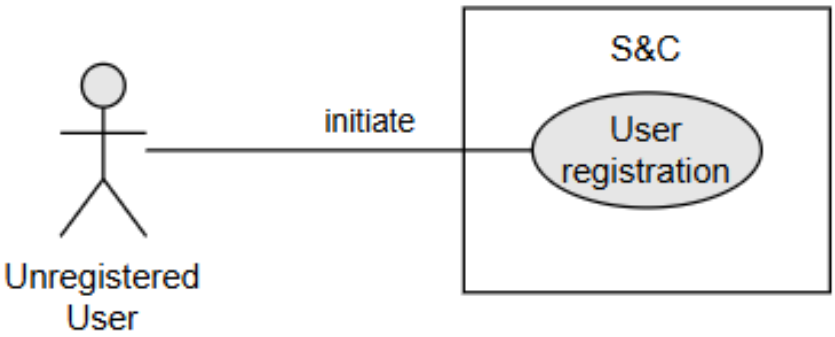
Communication between the system and its users will take place using secure protocols and built-in features:

- **HTTPS**: To ensure secure connections between clients and servers.
- **WebSocket**: For real-time notifications about new opportunities or updates on application statuses.
- **Integrated Chat and Video-Chat**: The platform will include a messaging feature for communication between students, companies, and universities, allowing the sharing of attachments such as CVs or supporting documents. Users will also have the possibility to participate to video-chat, using the built-in function, during the company interview's process.

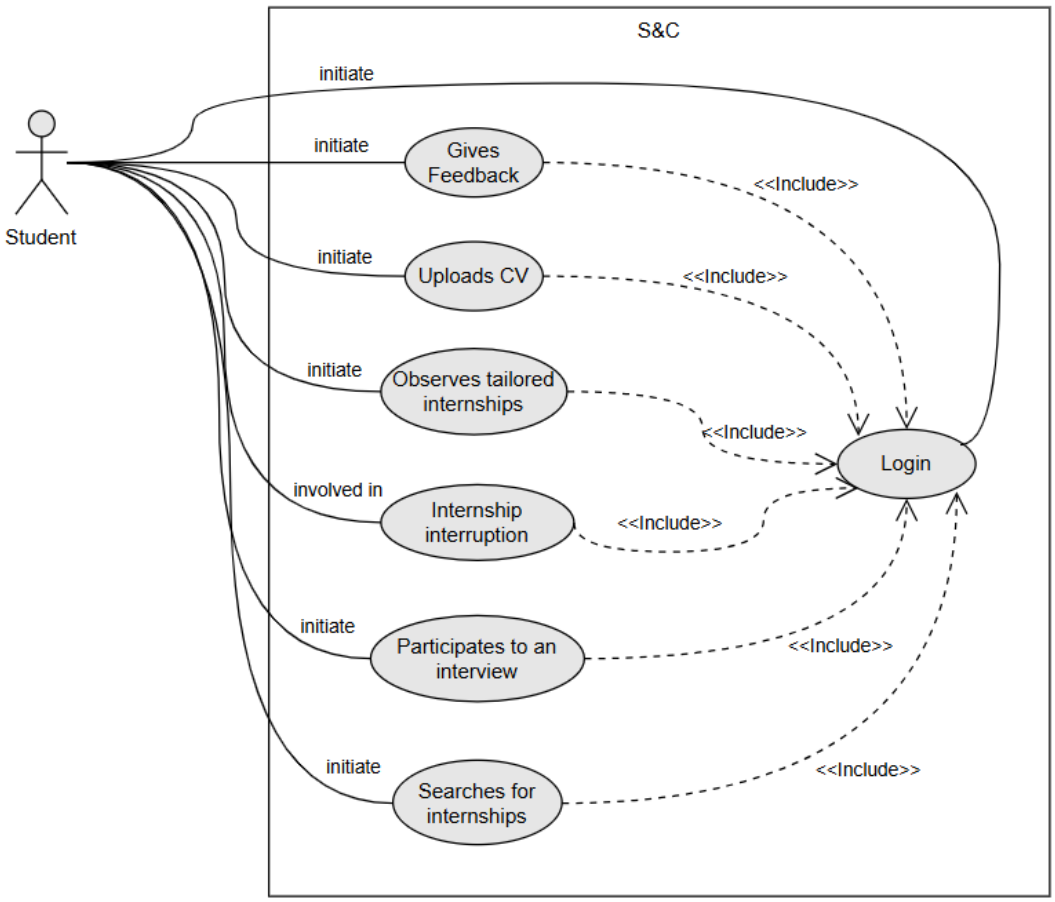
3.2 Functional Requirements

3.2.1 Use case diagrams

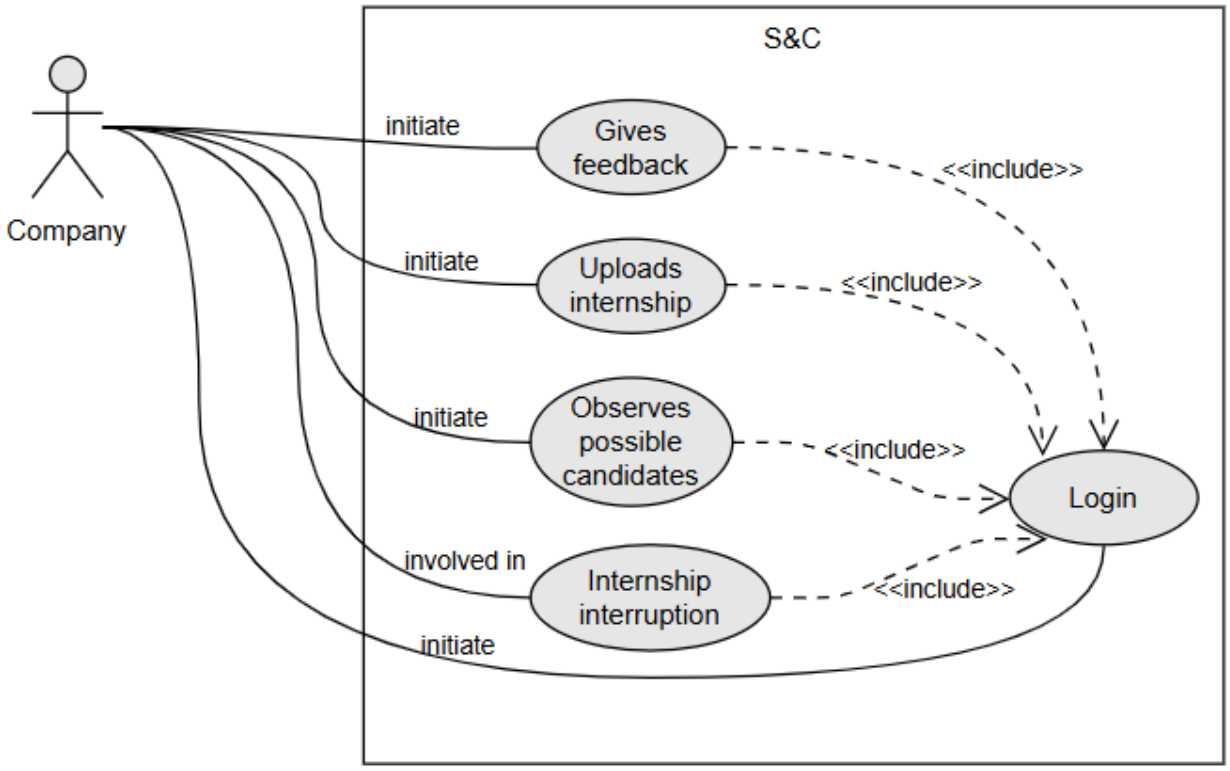
In this section it's possible to observe the use-case diagrams of S&C. They are divided into 4 different diagrams based on the actor: Unregistered User, Student, Company, University.



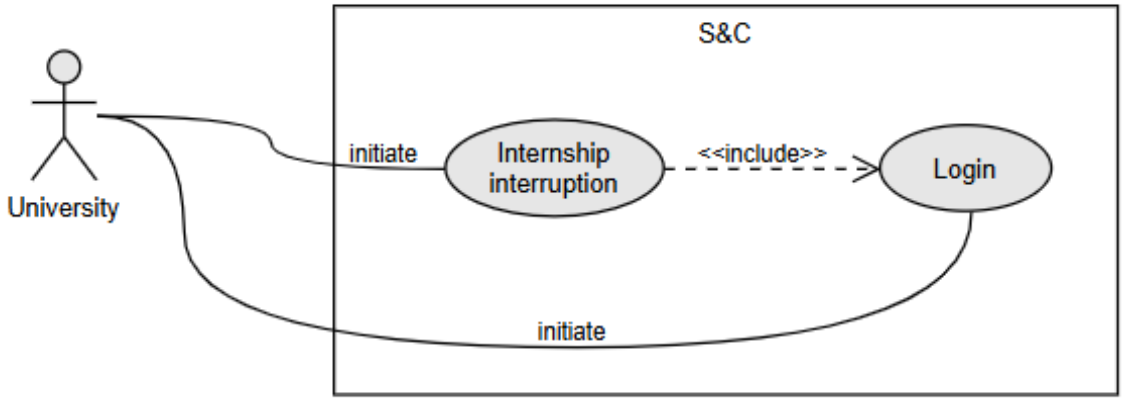
1_ This is the use-case diagram related to the unregistered users that can register to the S&C platform.



2_ The diagram presented up there is related to the Student actor: we modeled his main interaction with the platform as it will be possible to observe in the following sequence diagrams. Missing the UC on searching internships.



3_ This is the use-case diagram related to the Company actor: as for the Student it is presented an "Internship interruption" use case (that will also be in the University use-case diagram) that presents the situation in which a Company (or a Student) sends a complaint and the University has to manage it. As for the Student diagram there is the "Gives feedback" use-case that models the situation in which the users are required to provide information after the internship ends.



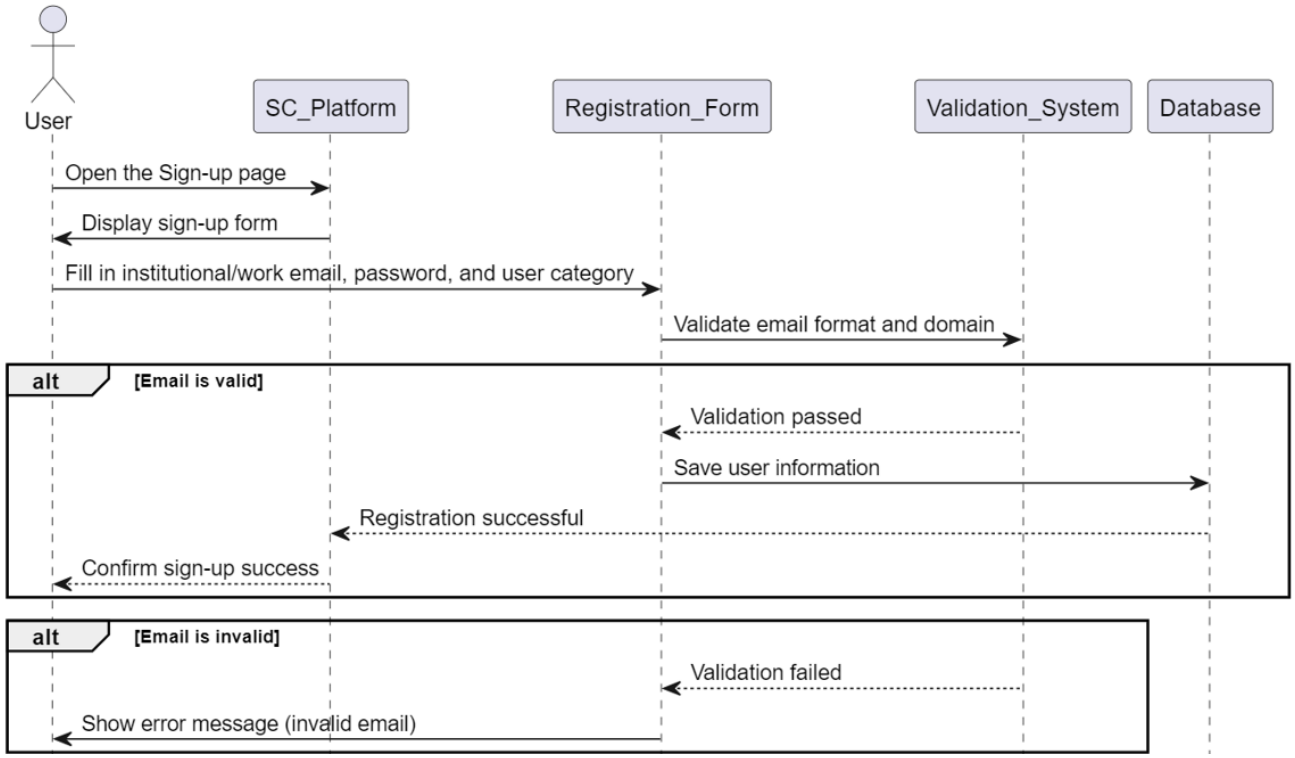
4_ The last one is a very simple diagram, also because the Universities has a reduced role on the platform. We modeled, as stated before, the "Internship interruption" use-case that is for us the most important one about this specific actor.

3.2.2 Use cases

In this section, the most critical use cases are presented. Each use case is detailed with an accompanying sequence diagram. A typical entry condition is that all actors must have a stable internet connection. Conversely, a common exception condition is the occurrence of an internet connectivity error. For simplicity and improved readability, certain exceptions have been omitted from the sequence diagrams. The conditions specified under "Entry Conditions" and "Exit Conditions" must be met to initiate or successfully complete the use case. Some blank spaces appear in the following pages due to the dimensions of some description or the relative sequence diagram.

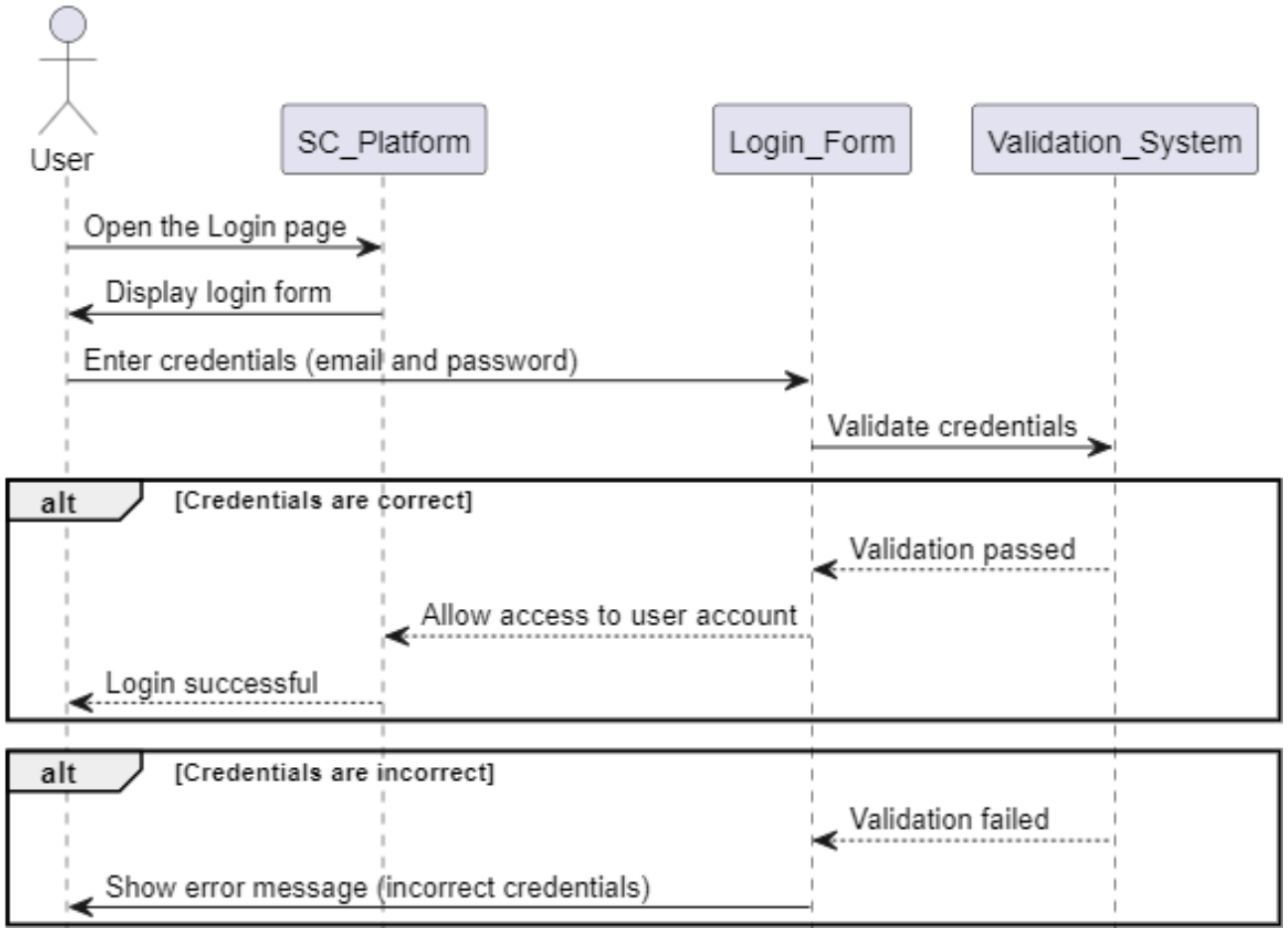
3.2.2.1 UC1: User registration

Actor	Unregistered user
Entry conditions	1. The user has a device with a browser and an internet connection.
Main Flow	1. The user browses on the sign-up page of S&C. 2. He/her have to: - Insert institutional/work email - Insert a Password - Insert username - Select the user category he/her want to sign up as. - Accept Terms of Service 3. The system accepts inserted data and allows the user to log in.
Exit conditions	1. The user is registered in the database.
Exceptions	1. The user uses a different email from the institutional/work one.


User
SC_Platform
Registration_Form
Validation_System
Database
Open the Sign-up page
Display sign-up form
Fill in institutional/work email, password, and user category
Validate email format and domain
alt
[Email is valid]
Validation passed
Save user information
Registration successful
Confirm sign-up success
alt
[Email is invalid]
Validation failed
Show error message (invalid email)

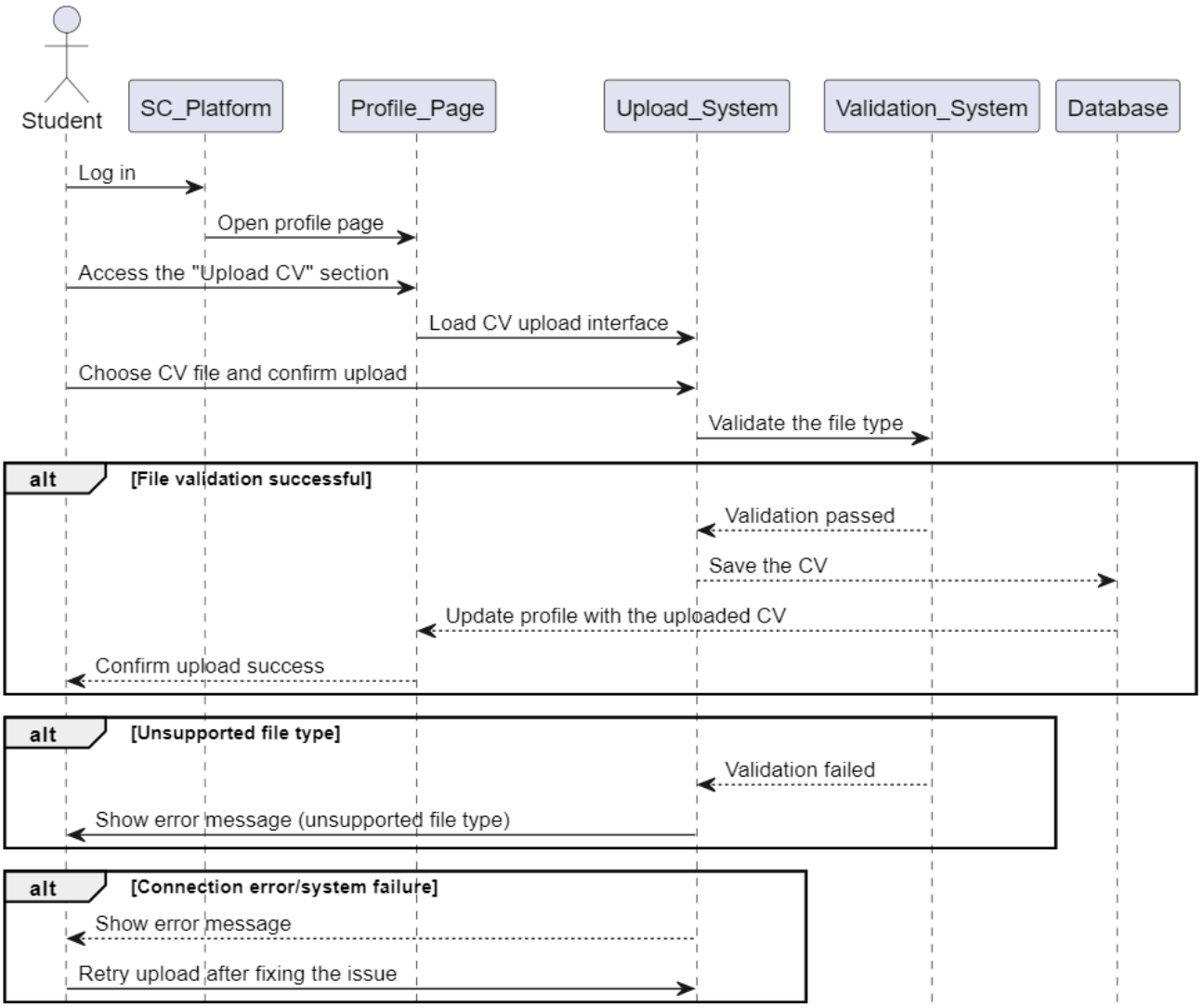
3.2.2.2 UC2: User login

Actor	User
Entry condition	1. The user has already an account on S&C platform.
Main Flow	1. The user browses to the login page. 2. The user inserts his/her credentials (email and password).
Exit Conditions	1. The system allows the user to log-in if the combination of credentials is correct.
Exceptions	1. The user inserts wrong credentials.



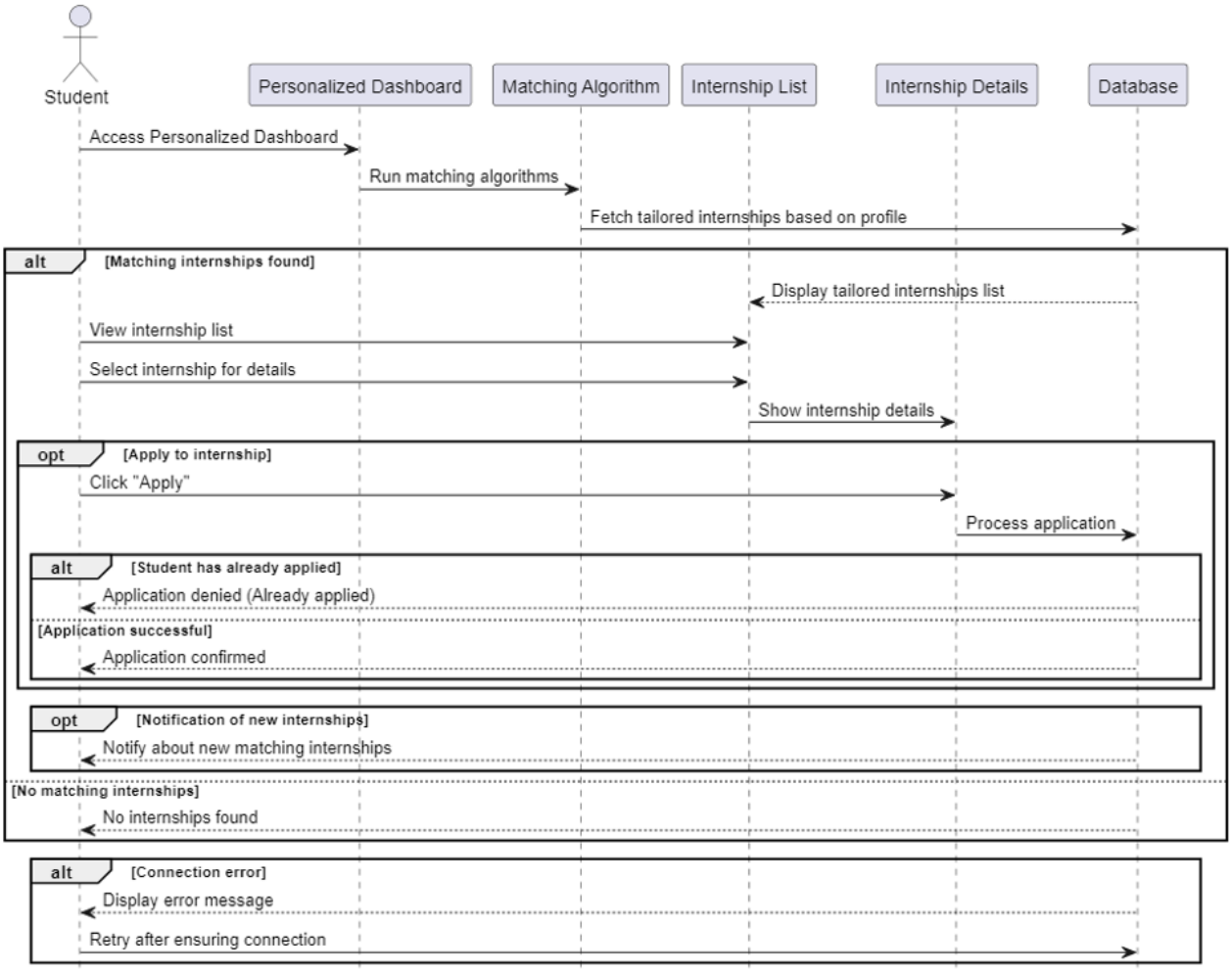
3.2.2.3 UC3: Student uploads his/her CV on platform

Actor	Student
Entry condition	1. The Student is logged-in S&C.
Main Flow	1. The Student accesses the "upload CV" section through his/her profile page. 2. The Student chooses the CV file to upload on his/her profile. 3. The Student confirms the upload.
Exit Conditions	1. The system confirms the correct upload.
Exceptions	1. The selected file isn't supported by the platform.



3.2.2.4 UC4: Student observes tailored internships

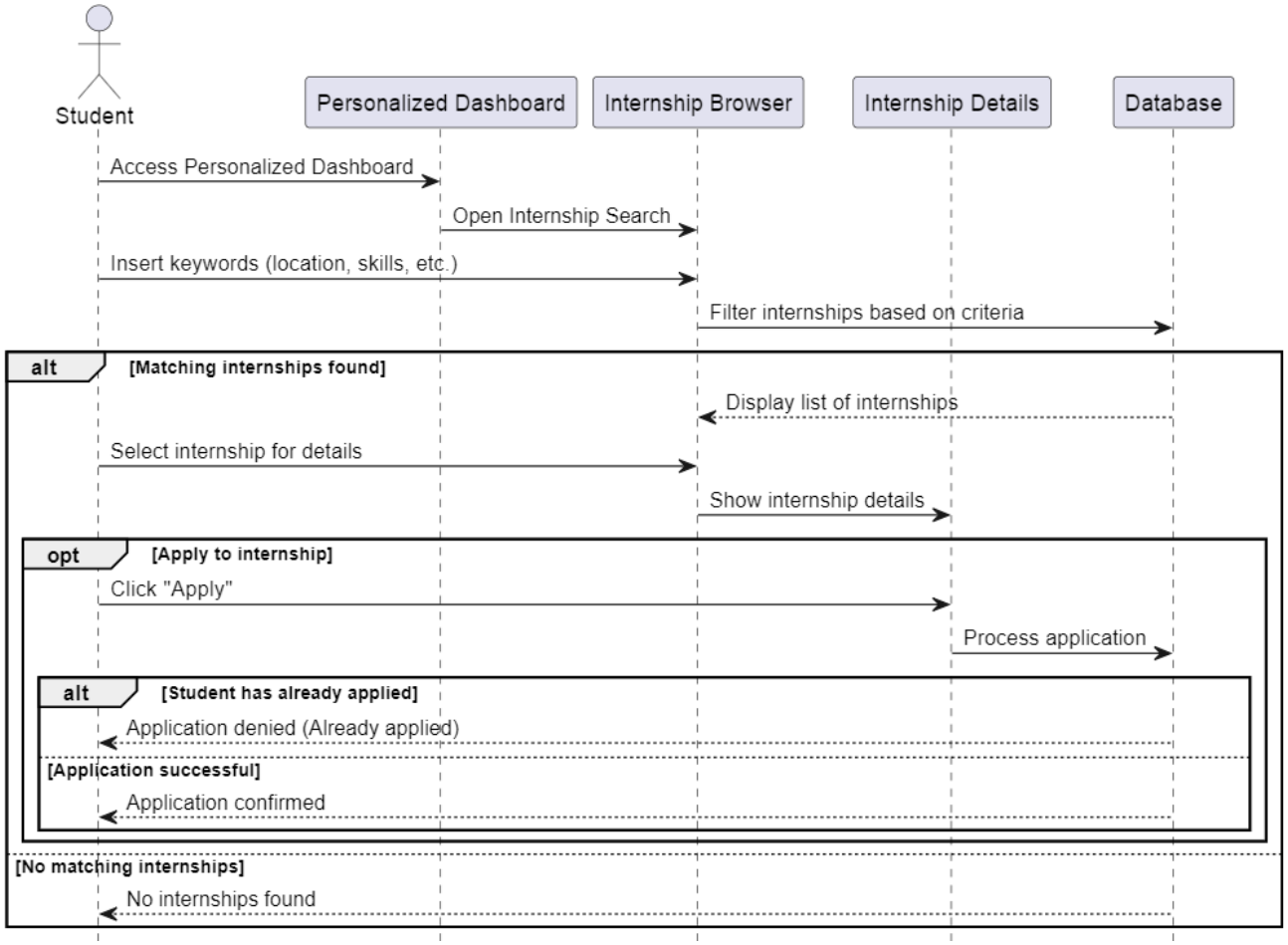
Actor	Student
Entry condition	1. The Student is logged into the platform. 2. The Student has a complete profile with relevant information listed in his/her CV (e.g., skills, qualifications, preferences). 3. The system has access to a stable internet connection.
Main Flow	1. The Student accesses the Personalized Dashboard via the user interface. 2. The system uses the matching algorithms to suggest relevant internship opportunities based on the Student's profile. 3. The Student views a list of recommended internship recommendations, including details such as keywords, location, salary, and work mode. 4. The Student selects an internship for more details. 5. The Student can apply to the internship, by clicking the button "Apply". 6. The system notifies the Student about new internships that match their profile.
Exit Conditions	1. The Student has visualized the tailored internship opportunities. 2. The student has applied to an internship and received confirmation. 3. The system has notified the student about new matching internships.
Exceptions	1. If there are no matching opportunities available, the system informs the Student that no matches were found. The Student can improve his/her profile to receive other recommendations. 2. The system shows an error message if there's an internet connection issue or another failure. The student is advised to try again after ensuring a stable connection.



3.2.2.5 UC5: Student search for an internship through the platform

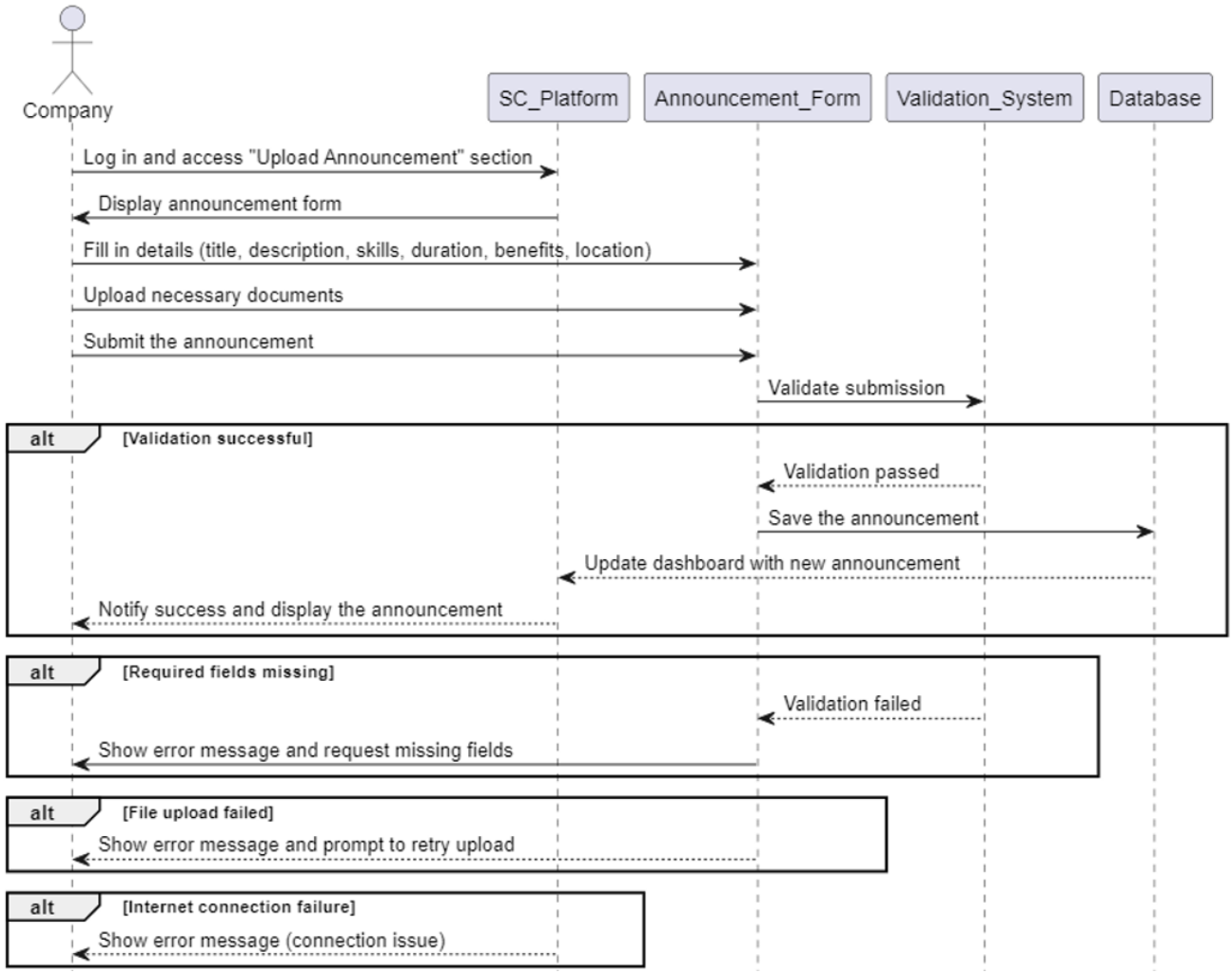
Actor	Student
Entry condition	1. The Student is logged-in S&C. 2. The Student has a complete profile with relevant information listed in his/her CV (e.g., skills, qualifications, preferences). 3. The system has access to a stable internet connection.

Main Flow	1. The Student accesses the Personalized Dashboard via the user interface. 2. He/She searches for a specific type of internship inserting in the platform's browser keywords (such as location, skills...). 3. The Student observes all the internships related to what he inserted. 4. The Student selects an internship for more details. 5. The Student can apply to the internship, by clicking the button "Apply".
Exit Conditions	1. The Student has visualized the filtered internship opportunities. 2. The student has applied to an internship and received confirmation.
Exceptions	1. If there are no matching opportunities available, the system informs the Student that no matches were found. 2. The student has already applied to this internship.



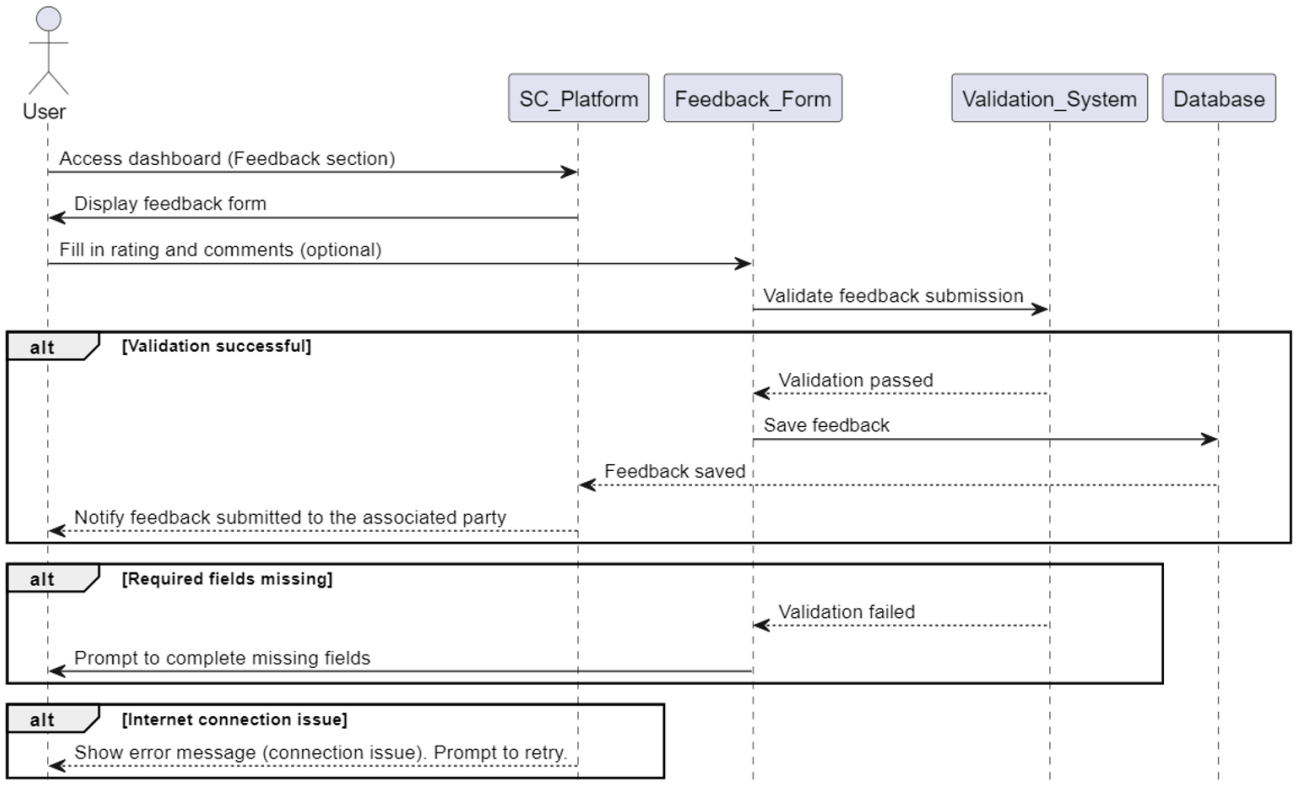
3.2.2.6 UC6: Company uploads an internship announcement

Actor	Company
Entry condition	1. The company is logged into the platform. 2. The company has a verified account with necessary permissions. 3. The system has access to a stable internet connection.
Main Flow	1. The company accesses the "Upload Internship Post" section. 2. The company fills in the announcement form with relevant details: - Internship title; - Description; - Required skills; - Duration, benefits (if paid, salary), and location; 3. The company uploads any necessary documents (such as questionnaires for the applicants). 4. The company submits the announcement. 5. The system validates the submission. 6. The system saves the announcement and notifies the company of success. 7. The system displays the new internship announcement in the company profile's dashboard.
Exit Conditions	1. The company has successfully uploaded the internship announcement. 2. The internship announcement is now visible to students in the system.
Exceptions	1. If required fields are missing, the system shows an error message. 2. If there is an internet connection failure, the system shows an error. 3. If the file upload fails, the system prompts the company to retry.


Company
SC_Platform
Announcement_Form
Validation_System
Database
Log in and access "Upload Announcement" section
Display announcement form
Fill in details (title, description, skills, duration, benefits, location)
Upload necessary documents
Submit the announcement
Validate submission
alt
[Validation successful]
Validation passed
Save the announcement
Update dashboard with new announcement
Notify success and display the announcement
alt
[Required fields missing]
Validation failed
Show error message and request missing fields
alt
[File upload failed]
Show error message and prompt to retry upload
alt
[Internet connection failure]
Show error message (connection issue)

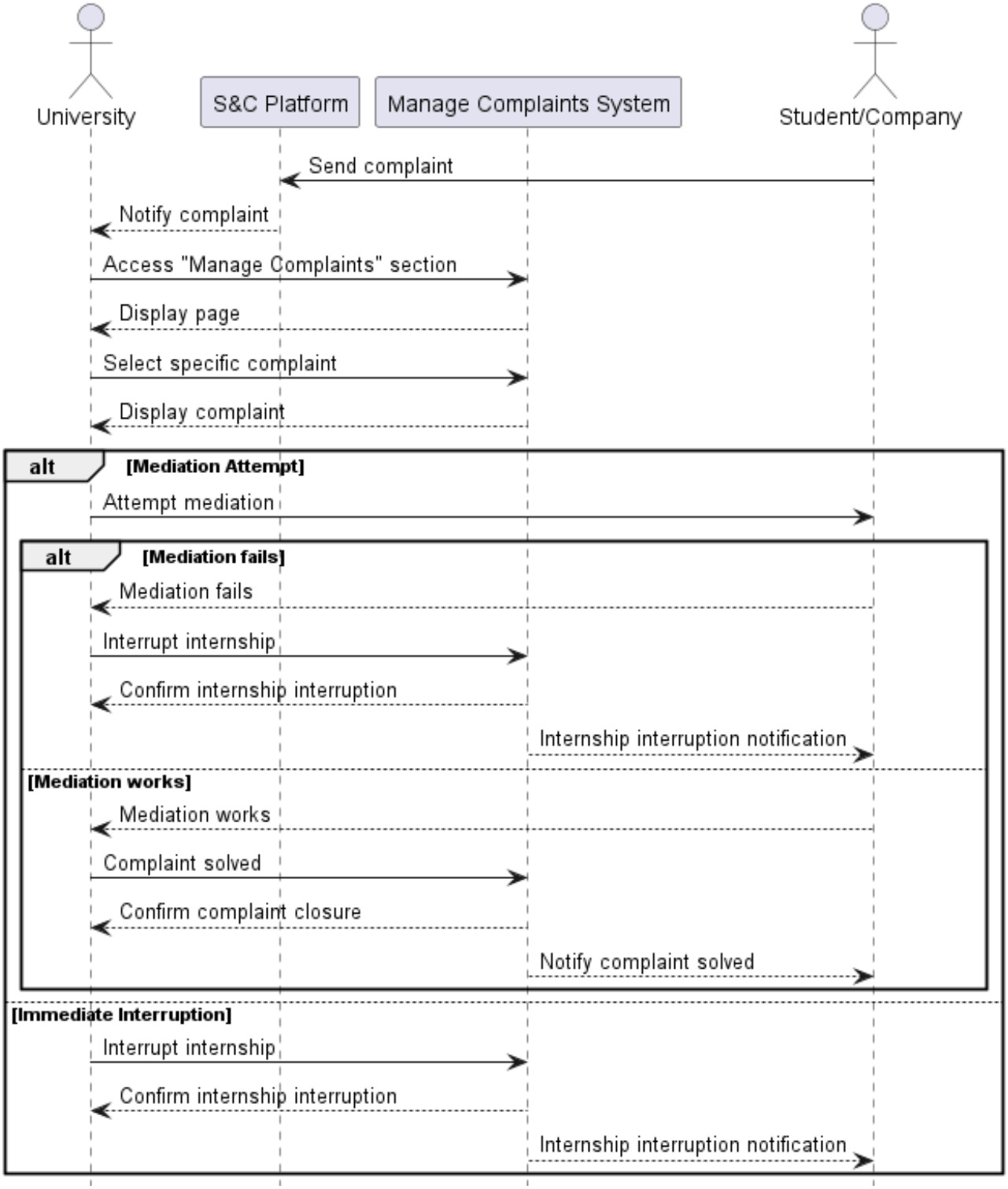
3.2.2.7 UC7: Student or Company gives feedback

Actor	Student and Company
Entry condition	1. The user (Student or Company) is logged into the platform. 2. The internship associated with the feedback has been completed/interrupted, or a specific interaction/event has occurred. 3. The system has access to a stable internet connection.
Main Flow	1. The user accesses the "Feedback" section via their dashboard. 2. The system displays a feedback form with the following fields: - Different ratings (e.g., 1 to 5 stars) to evaluate if the user expectation where satisfied, if it was a good experience and so on. - Comments or suggestions (optional). 3. The user fills out the feedback form and submits it. 4. The system validates the submission to ensure the required fields are completed (e.g., a rating is provided). 5. The system saves the feedback in the database. 6. The system notifies the other party (student or company) that feedback has been provided.
Exit Conditions	1. The feedback is successfully saved in the system. 2. The feedback is visible to the associated parties (student, company, and optionally university administrators).
Exceptions	1. If required fields are missing, the system prompts the user to complete them. 2. If there is an internet connection failure, the system shows an error and prompts the user to retry.



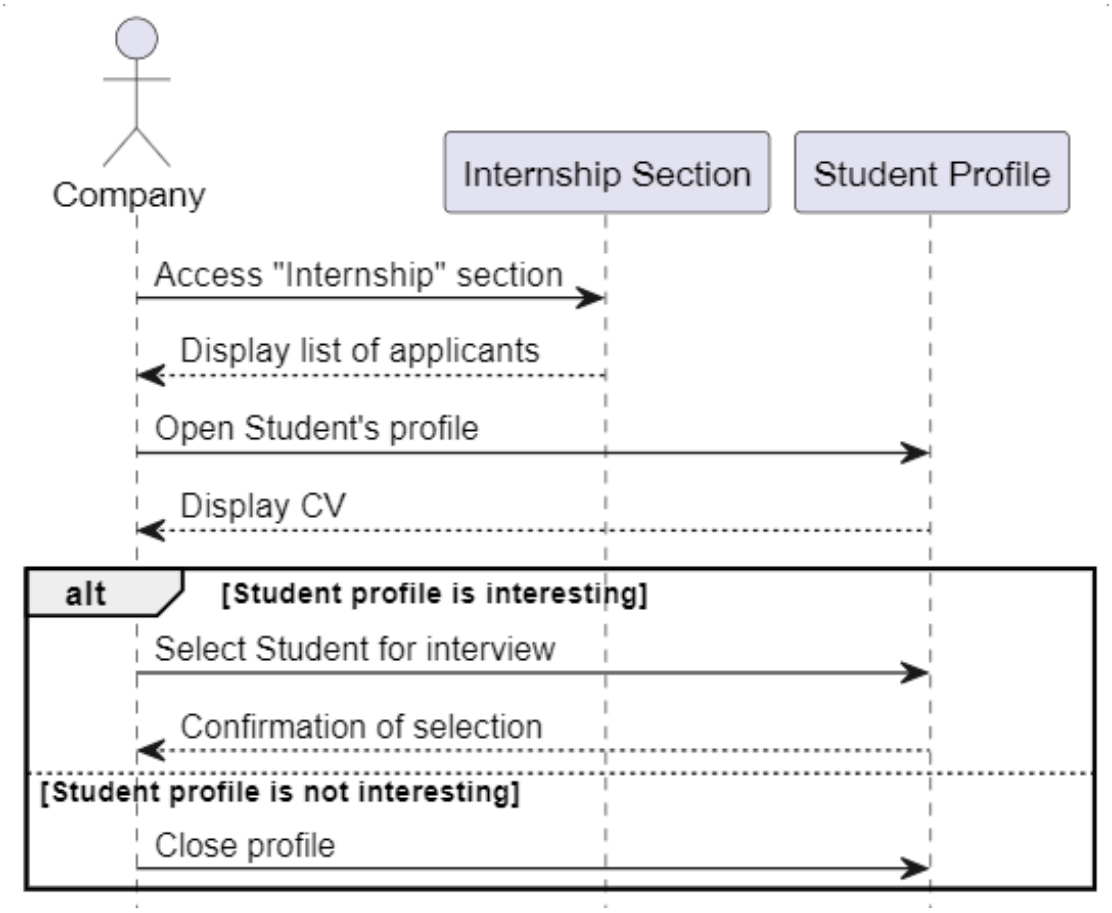
3.2.2.8 UC8: University manages one of its student internship

Actor	University, Student, Company
Entry condition	1. The University representative is logged-in S&C. 2. A university's student or the company he is working for as an intern sent a complaint to the university.
Main Flow	1. The University representative accesses the "manage complaints" section through his/her profile page. 2. The University representative reads the complaint and decide if: - Try to have a chat with the interested parties to solve the problem - Immediately interrupt the student's internship clicking the button "Interrupt" 3. If the mediation tentative doesn't go well, the university interrupts the student's internship anyway.
Exit Conditions	1. The system confirms the correct internship interruption. 2. The complaint is solved, thanks to university's mediation.


University
S&C Platform
Manage Complaints System
Student/Company
Send complaint
Notify complaint
Access "Manage Complaints" section
Display page
Select specific complaint
Display complaint
alt
[Mediation Attempt]
Attempt mediation
alt
[Mediation fails]
Mediation fails
Interrupt internship
Confirm internship interruption
Internship interruption notification
[Mediation works]
Mediation works
Complaint solved
Confirm complaint closure
Notify complaint solved
[Immediate Interruption]
Interrupt internship
Confirm internship interruption
Internship interruption notification

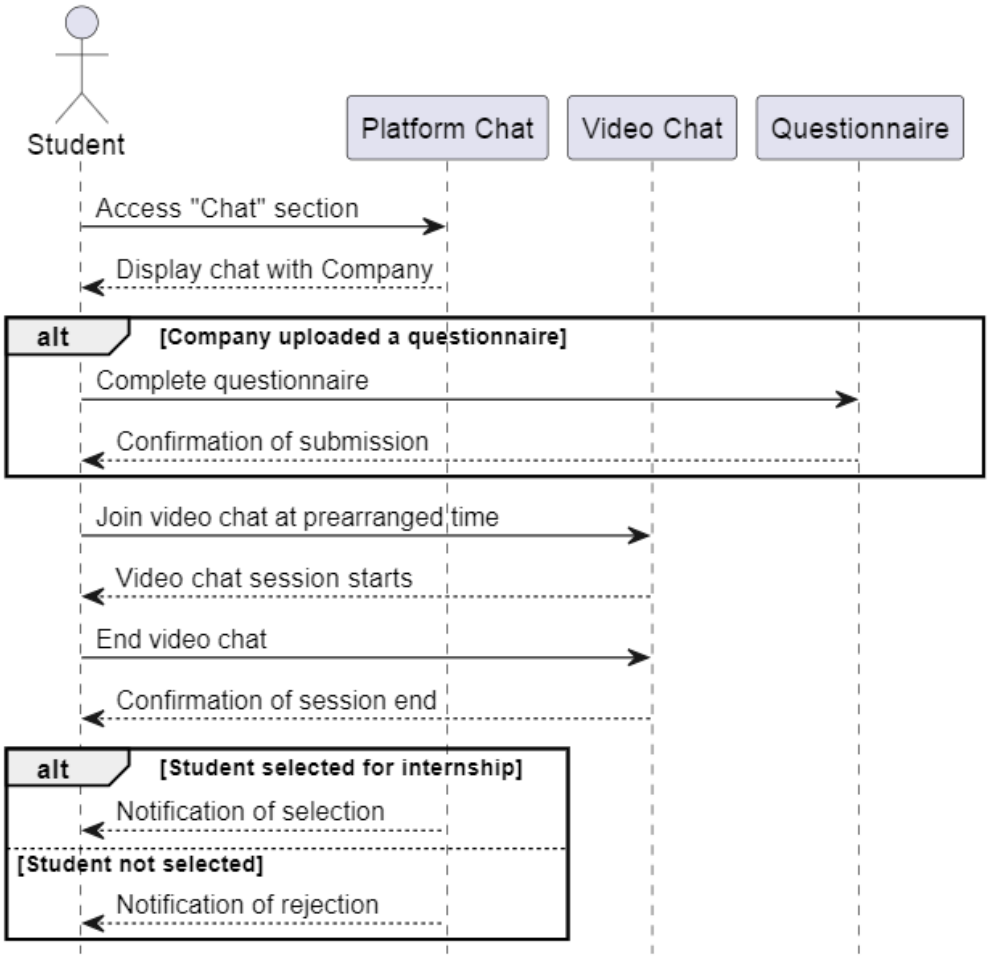
3.2.2.9 UC9: Company observes candidates for its internship

Actor	Company
Entry condition	1. The Company is logged-in S&C. 2. Company received some applications for its internship spot.
Main Flow	1. The Company accesses the specific "internship" section through his/her profile page. 2. Here the Company can visualize all the students who applied for the internship. 3. The Company opens a Student profile to inspect his/her CV. 4. If the Company finds the Student's profile interesting, they can select him/her to attend the interview process.
Exit Conditions	1. The Company closes the Student's profile page. 2. The Company invites the Student for an interview.



3.2.2.10 UC10: Student participates to an interview

Actor	Student
Entry condition	1. The Student is logged-in S&C. 2. Student was selected by a Company to attend an interview.
Main Flow	1. The Student accesses the platform's "chat" section through his/her profile page. 2. Here the Student finds the chat with the Company that selected him. 3. If uploaded by the Company, the Student is asked to compile a questionnaire that will be later discussed. 3. A video chat is activated at the prearranged time in order to carry out the interview.
Exit Conditions	1. The video chat ends. 2. The Student receives notification about being selected for the internship.



3.2.3 General Requirements (GR)

[GR1] Platform must **prevent** multiple accounts from using the same E-Mail address.

[GR2] Platform must **permit** unregistered users only to access the registering function.

[GR3] Platform must **allow** unregistered users to register as Student, Company or University.

[GR4] Platform must **ensure** to send all the necessary notifications to the correct users right after something relevant occurs.

[GR5] Platform must **find** optimal matching between Students and Companies, based on localization and a match between Company's internship requested skills and Student's abilities.

3.2.4 Student Platform Requirements (SR)

[SR1] Platform must **prevent** Students to access the platform as Company or University account.

[SR2] Platform must **allow** Students to pro-actively search for internships through the platform itself.

[SR3] Platform must **allow** Students to filter internships' announcements based on their specific likings.

[SR4] Platform must **allow** Students to apply for any internship they find interesting.

[SR5] Platform must **prevent** Students to apply more than once for the same internship.

[SR6] Platform must **allow** Students to complain or provide any type of information about the ongoing internship through the platform itself.

[SR7] Platform must **suggest** Students on how to get their CVs more appealing to Companies.

[SR8] Platform must **notify** Students about new suitable internships for them.

[SR9] Platform must **allow** Students to monitor their ongoing internship's situation.

[SR10] Platform must **allow** Students to participate to interviews if selected by any Company.

[SR11] Platform must **allow** Students to provide feedback after their internship experience to improve the platform's matching algorithm.

[SR12] Platform must **allow** Students to chat with their University's account and

with the Companies they are interested working in or with which they are already doing an internship with.

[SR13] Platform must **show** Students only available internships.

[SR14] Platform must **allow** Students to upload their CV and update it whenever they want to.

[SR15] Platform must **notify** Students when they are accepted for an internship experience by a Company.

[SR16] Platform must **deny** Students to upload a non European Format CV (see *Reference*)

[SR17] Platform must **notify** Students about the result of their complaint.

[SR18] Platform must **notify** Students when the Company of a recommended internship they tried to apply for, match with their profile.

3.2.5 Company Platform Requirements (CR)

[CR1] Platform must **prevent** Companies to access the platform as Student or University account.

[CR2] Platform must **allow** Companies to upload their internship announcements whenever they want to.

[CR3] Platform must **allow** Companies to contact recommended Students whose CVs appear interesting to them.

[CR4] Platform must **show** Companies only available Students, who are not already on an internship.

[CR5] Platform must **allow** Companies to chat with Students who are working with them as interns.

[CR6] Platform must **allow** Companies to chat with their specific interns' Universities in case of any problem.

[CR7] Platform must **allow** Companies to provide feedback after Students' internship experiences to improve the platform's matching algorithm.

[CR8] Platform must **allow** Companies to monitor their interns situation.

[CR9] Platform must **prevent** Companies to publish more than once the same internship's announcement.

[CR10] Platform must **allow** Companies to complain or provide any type of information about the ongoing internships of its interns through the platform itself.

[CR11] Platform must **notify** Companies about new suitable student candidates for their internships.

[CR12] Platform must **notify** Companies if an internship is interrupted by the student's university.

[CR13] Platform must **notify** Companies about the result of their complaint.

[CR14] Platform must **allow** Companies to plan and carry out interviews to select candidates for their internships

[CR15] Platform must **notify** Companies when a recommended Student they tried to contact, match with one of their internships

[CR16] Platform must **suggest** Companies on how to make their Internship post more appealing to Students.

3.2.6 University Platform Requirements (UR)

[UR1] Platform must **prevent** University to access the platform as Student or Company account.

[UR2] Platform must **allow** Universities' accounts to monitor and access the information on all the specific University's students and their ongoing internships.

[UR3] Platform must **allow** Universities' accounts to chat with the specific University's students and the companies its students are working as interns in.

[UR4] Platform must **prevent** Universities' accounts to monitor other Universities' students internships and situation.

[UR5] Platform must **allow** Universities' accounts to formally interrupt its students' ongoing internships if any kind of problem happens.

[UR6] Platform must **notify** Universities when a complaint is filled out by one of its students or one of the companies its students are working in as interns.

[UR7] Platform must **allow** Universities' accounts to manage complaints coming from its students or the companies its students are working in as interns.

3.2.7 Mapping on Goals

G1: Students can find the most suitable internship for them.

- Consistency and prevention of improper usage: D 5; GR 1,2,3; SR 1,5,14,16
- Finding the best match: D 3,4,5,6; GR 5; SR 2,3,4,7,10,13
- Carrying out the internship: SR 6,9,12
- Improving the precision of the algorithm: D 2; SR 11
- Notifications: D 1; GR 4; SR 8,15,17,18

G2: Companies can promote their internships, posting them on the platform.

- Consistency and prevention of improper usage: D 6; GR 1,2,3; CR 1,9
- Promoting: D 4; GR 5; CR 2,16
- Notifications: D 1; GR 4

G3: Companies can select the most suitable candidates.

- Consistency and prevention of improper usage: GR 1,2,3; CR 1
- Finding the best match: D 3,4,5,6; GR 5; CR 3,4,14
- Monitoring the interns: CR 5,6,8,10
- Improving the precision of the algorithm: D 2; CR 7
- Notifications: D 1; GR 4; CR 11,12,13,15

G4: Universities can manage their students' internships.

- Consistency and prevention of improper usage: GR 1,2,3; UR 1,2,4
- Internships monitoring: D 3,4; UR 2,3,5
- Complaints management: UR 6,7
- Notifications: GR 4

3.2.8 Use-cases mapping on requirements

Use-case	Requirements
UC1	GR 1,2,3
UC2	SR 1; CR 1; UR 1
UC3	SR 7,14,16
UC4	GR 5; SR 4,5,8,13,18
UC5	SR 2,3,4,5,13
UC6	CR 2,9,16
UC7	GR 4; SR 11; CR 7
UC8	SR 6,9,12,17; CR 5,6,8,10,12,13; UR 2,3,4,5,6,7
UC9	CR 3,4,11,15
UC10	SR 10,15; CR 14

3.3 Performance Requirements

To design the hardware that will handle parallel connections we are considering that students in Italy are $\approx$ 2 millions. So, the system must ensure stability for, at least, 300 thousands connections. Download and upload minimum speeds aren't really a huge concern because the biggest files will be the students' CVs and, eventually, profile photos. Most common data will be textual descriptions. The server that will need more computational power is the one that hosts the matching engine. In order to be as more accurate as possible, it has to combine all the information regarding old internships and CVs and refresh it continuously with newer data. The service has to be available h24, 365 days a year with minimum downtime.

3.4 Design Constraints

3.4.1 Standard compliance

For the mere functioning, the system doesn't need any particular privilege. However it is recommended to allow browser notification in order to be notified whenever a message is received.

3.4.2 Hardware and Software constraints

The user who is accessing the server needs to have a web browser on his/her device and a reliable connection ($\geq$ 5 Mb/s). On the other hand, as previously described, the server has to be powerful enough to handle thousand of simultaneous connections and an application that can interact with a DBMS.

3.5 Software System Attributes

3.5.1 Security

The personal information of students, like the CVs, as well as sensitive data related to universities and companies, must be thoroughly protected to prevent unauthorized access and data breaches. To achieve this, encryption should be applied to all login credentials, ensuring that sensitive information is not exposed during authentication processes. Additionally, the database should be encrypted both at rest and in transit, employing industry-standard encryption techniques such as Advanced Encryption Standard (AES) for stored data and Transport Layer Security (TLS) protocols for data in transit. To protect against web-based threats such as Man-In-The-Middle (MITM) attacks, SQL injection, and cross-site scripting (XSS), the system should implement secure coding practices and perform regular security audits. Input validation and output encoding must be applied rigorously to prevent unauthorized data manipulation. Moreover, all

sensitive information transmitted over the network should be encrypted using TLS, and database connections must be established over secure channels to ensure that data cannot be intercepted or tampered with. Regular penetration testing and vulnerability assessments should also be conducted to identify and mitigate emerging threats in both the web application and database layers. By combining encryption, secure coding practices, and proactive security measures, the system can safeguard against both data and web-level security threats.

3.5.2 Reliability

The system must be able to tolerate possible failures that would increase downtime. To do so, a backup system should be created, that allows the replication of the data contained in the main server but also of the processes that provide the services of the system. If the system has a scheduled maintenance, the system must notify the users at least 2 days before. In any case, even if the system goes down unexpectedly, it must notify the users that it is back up running.

3.5.3 Availability

The system must ensure maximum availability, as users may need access to information about their internships or interns at any time during any working day of the year. We consider a 99.99% availability rate to be a reasonable target, which equates to approximately 52 minutes of downtime per year. To achieve this level of availability, the system should have a secondary server running in parallel with the main server, so that if the central server experiences a failure, the backup server can seamlessly take over.

3.5.4 Maintainability

The system should be implemented using design patterns that promote code re-usability and flexibility, making it easier to modify and extend in the future, as well as to address potential issues or malfunctions. It should be implemented using an object-oriented programming language, with comprehensive documentation for most classes and methods. This will ensure the code is easy to read, understand, and maintain, allowing other engineers to quickly grasp its structure and functionality when needed.

3.5.5 Portability

The system must be compatible with a variety of operating systems, including Android and iOS for mobile devices, as well as Windows, Linux, and macOS for desktops. This ensures that both students and company representatives can access the system and use email services seamlessly across different devices and platforms.

4 Formal Analysis using Alloy

The main goal of this formal model is to represent and manage the matching process between university students and internships offered by companies. The system supports recommendations, manages the status of internships, tracks participation, and ensures universities can monitor internships involving their students. The model consists of signatures representing users (students, companies, universities), internships, feedback, and skills. It ensures the consistency of internship data, verifies valid participation (one internship at a time per student), checks skill matching between students and internships, and manages feedback (both positive and negative). The model proves that internships are correctly monitored by universities, students are matched with relevant internships, and feedback is handled appropriately. Running the model show facts like uniqueness of user IDs, valid participation, and feedback consistency, confirming the reliability and correctness of the system. We will be using Alloy 6.1 version.

4.1 Alloy model

```
// Signature for Attributes
sig ID {}{one u:User| this = u.id}
sig Username {}{one u:User| this= u.username}
sig Password {}{one u:User|this= u.password}
sig Email {}{one u:User| this = u.email}
sig Interest {}{some c:CurriculumVitae| this in c.interests}
sig Attitude {}{some c:CurriculumVitae| this in c.attitudes}
sig Name {}{(one u:User| this =u.name) or (one s:Skill|this = s.
   name)}
sig Description {}{this in Experience.description+Internship.
   description+Feedback.text}
sig Address {}{this in University.headquarters + Company.
   headquarters}
sig Benefit{}{some i:Internship| this in i.benefits}
sig Title{}{one i:Internship| this =i.title}

// Abstract sig used for Bool
abstract sig Bool {}
one sig True,
        False extends Bool {}

// Valid Status for Internships
abstract sig InternshipStatus {}
one sig Published, SelectionPhase,
        Finalized, Active,
        Ended, Interrupted extends InternshipStatus {}

// Generic user
abstract sig User {
    id: one ID,
```

```
    name: one Name,
    username: one Username,
    password: one Password,
    email: one Email
}

// Student, extension of User
sig Student extends User {
    cv: one CurriculumVitae,
    participations: set InternshipParticipation,
    receivedFeedback: set Feedback,
    averageRating: one Int
} {one u:University| this in u.students
    averageRating>0
    #receivedFeedback > 0 implies
    (averageRating >= 1 and averageRating <= 10)
all f:receivedFeedback| f.directedTo=this
}

// Company, extension of User
sig Company extends User {
    headquarters: one Address,
    internshipsOffered: set Internship,
    receivedFeedback: set Feedback,
    averageRating: one Int
}{
averageRating>0
    #receivedFeedback > 0 implies
  (averageRating >= 1 and averageRating <= 10)
#receivedFeedback=1 implies averageRating =receivedFeedback.vote
all f:receivedFeedback| f.directedTo=this
}

// University, extension of User
sig University extends User {
    students: set Student,
    headquarters: one Address,
    monitoredInternships: set InternshipParticipation
} {
    #students > 1
}

// Curriculum Vitae
sig CurriculumVitae {
    skills: some Skill,
    experiences: set Experience,
    interests: some Interest,
    attitudes: some Attitude
}{one s:Student| this = s.cv}

// Skills
sig Skill {
```

```
    name: one Name,
    proficiency: one Int
} {
    proficiency >= 1 and proficiency <= 10
}

// Past and current working experience
sig Experience {
    description: one Description,
    company: one Company,
    startDate: one Date,
    endDate: one Date
} {
    startDate.year<endDate.year or
(startDate.year=endDate.year and startDate.month<endDate.month) or
(startDate.year=endDate.year and startDate.month=endDate.month and
    startDate.day<endDate.month)
}

// Internship
sig Internship {
    title: one Title,
    description: one Description,
    requiredSkills: set Skill,
    payment: one Bool,
    benefits: set Benefit
}{one c:Company| this in c.internshipsOffered}

// Date
sig Date{
day: one Int,
month: one Int,
year: one Int
}{some i:InternshipParticipation| this in i.startDate + i.endDate
some e:Experience| this in e.startDate+ e.endDate}

// Enrollment in an Internship
sig InternshipParticipation {
    student: one Student,
    internship: one Internship,
    company: one Company,
    startDate: one Date,
    endDate: one Date,
    status: one InternshipStatus
} {
one s:Student | this in s.participations
    startDate.year<endDate.year or
(startDate.year=endDate.year and startDate.month<endDate.month) or
(startDate.year=endDate.year and startDate.month=endDate.month and
    startDate.day<endDate.month)
}

```

```
// Feedback
sig Feedback {
    author: one User,
directedTo: one User,
    text: one Description,
    vote: Int
} {
author in Student and directedTo in Company or
author in Company and directedTo in Student
author!= directedTo
vote >= 1 and vote <= 10
}

//Uniqueness Facts

// Fact to ensure that each user has a unique ID
fact UniqueUserID {
     all disj u1, u2: User | u1.id != u2.id
}
// Fact to ensure that each user has a unique email address
fact UniqueUserEmail {
     all disj u1, u2: User | u1.email != u2.email
}
// Fact to ensure that each user has a unique username
fact UniqueUsername {
     all disj u1, u2: User | u1.username != u2.username
}
// Fact to ensure that no university has a company name and
   students doesn't have strange names
fact noStrangeNames{
all disj u1, u2: User| u1 in University implies u1.name!=u2.name
all disj u1,u2: User| u1 in Company implies u1.name!=u2.name
all s:Skill| all u:User| s.name!=u.name
}
// Fact to ensure that each university has a unique name
fact UniqueUniversityName {
     all disj u1, u2: University | u1.name != u2.name
}
//Fact to ensure that if the rating is>0,the # of received
   feedback is >0
fact ratingNeedsFeedback{
all c: Company | c.averageRating != none => #c.receivedFeedback >=
    1
}
// Fact to ensure that each university has a unique address,
   companies
//created by university could exist anyway
fact UniqueUniversityAddress {
     all disj u1, u2: University | u1.headquarters !=
     u2.headquarters
}
// Fact to ensure that each company has a unique address
```

```
fact UniqueCompanyAddress {
     all disj c1, c2: Company | c1.headquarters !=
     c2.headquarters
}
// Fact to ensure that each skill has a unique name
fact UniqueSkillName {
     all disj s1, s2: Skill | s1.name != s2.name
}
// Fact to ensure that each student's CV is unique
// (no two students have the same CV)
fact UniqueCurriculumVitae {
     all disj s1, s2: Student | s1.cv != s2.cv
}
// Fact to ensure that all skills in a student's CV are unique
fact UniqueCVSkills {
     all s: Student |
     no disj sk1, sk2: s.cv.skills | sk1.name = sk2.name
}
// Fact to ensure that all required skills for an internship
// are unique
fact UniqueRequiredSkills {
     all i: Internship |
     no disj sk1, sk2: i.requiredSkills | sk1.name = sk2.name
}
// Fact to ensure that each experience (role and company
// combination) is unique for every student
fact UniqueExperience {
     all disj e1, e2: Experience | e1.description !=
     e2.description or e1.company != e2.company
}
// Fact to ensure that each internship participation is unique for
// each student and internship combination
fact UniqueInternshipParticipation {
     all disj ip1, ip2: InternshipParticipation |
         ip1.student != ip2.student or
         ip1.internship != ip2.internship
}
// Fact to ensure that internship participations have unique start
// and end dates for each student and internship combination
fact UniqueInternshipDates {
     all disj ip1, ip2: InternshipParticipation | (ip1.student =
        ip2.student) implies
         ip1.startDate != ip2.startDate or
         ip1.endDate != ip2.endDate
}
//Fact to ensure that feedback is provided only when an
   internships is ended
fact FeedbackOnlyForEndedInternships{
 all f: Feedback | some ip: InternshipParticipation |
         f.author in ip.student or f.author in ip.company
          and ip.status in Ended + Interrupted
}
```

```
228 //Fact to ensure that university monitor only their students and
    if one of
229 //its student is on an internship, this internship is monitored by
    the university
230 fact MonitorStudents{
231 all u:University| all s: u.monitoredInternships.student | s in u.
    students
232 all ip:InternshipParticipation| one u:University| ip.student in u.
    students implies
233 ip in u.monitoredInternships}
234
235 // Fact to ensure that if a student participate in a internship,
    all the signature are correct
236 fact InternshipStudent{
237 all  ip:InternshipParticipation| all s:Student | s = ip.student
    iff ip in s.participations
238 }
```

4.2 World description and predicates

4.2.0.1 Assertions

Here we present some assertions we verified while working on our model.

```
1  assert FeedbackOnlyForEndedInternships {
2      all f: Feedback | some ip: InternshipParticipation |
3          f.author in ip.student or f.author in ip.company
4           and ip.status in Ended + Interrupted
5  }
6
7  check FeedbackOnlyForEndedInternships for 5 Int, exactly
8      4  User, 5 CurriculumVitae, 5 Skill, 5 Address, 5 ID,
9      5 Experience, 5 Internship, 6 InternshipParticipation,
10     5 Feedback, 5 Username, 5 Password, 5 Email, 5 Name,
11     5 Interest, 5 Attitude, 5 Description, 5 Benefit, 5 Title, 5
          Date
12
13 assert NoDuplicateInternships {
14     all s: Student | all disj ip1, ip2: s.participations |
15         ip1.internship != ip2.internship
16 }
17
18 check NoDuplicateInternships for  5 Int, exactly
19     4  User, 5 CurriculumVitae, 5 Skill, 5 Address, 5 ID,
20     5 Experience, 5 Internship, 6 InternshipParticipation,
21     5 Feedback, 5 Username, 5 Password, 5 Email, 5 Name,
22     5 Interest, 5 Attitude, 5 Description, 5 Benefit, 5 Title, 5
          Date
23
24 assert NoNegativeAverages{
25 all s:Student| s.averageRating>0
26 all c:Company| c.averageRating>0
```

```
}

check NoNegativeAverages for  5 Int, exactly
    4  User, 5 CurriculumVitae, 5 Skill, 5 Address, 5 ID,
    5 Experience, 5 Internship, 6 InternshipParticipation,
    5 Feedback, 5 Username, 5 Password, 5 Email, 5 Name,
    5 Interest, 5 Attitude, 5 Description, 5 Benefit, 5 Title, 5
       Date
```

4.2.1 eligibleStudentsForInternship predicate

```
// Predicate to mimic the matching process
pred eligibleStudentsForInternship[i: Internship] {
    some s: Student | (all sk: i.requiredSkills
                      | sk in s.cv.skills)
}

/*
run eligibleStudentsForInternship for  5 Int, exactly
    4  User, 5 CurriculumVitae, 5 Skill, 5 Address, 5 ID,
    5 Experience, 5 Internship, 6 InternshipParticipation,
    5 Feedback, 5 Username, 5 Password, 5 Email, 5 Name,
    5 Interest, 5 Attitude, 5 Description, 5 Benefit, 5 Title, 5
       Date
*/
// Variant with proficiency - an integer in the range of 1 to 10
// to determine whether the candidate ability is enough or not
pred eligibleStudentsForInternshipWithProficiency[i: Internship] {
     some s: Student | some sk:Skill| sk in i.requiredSkills and
        sk in
          s.cv.skills and sk.proficiency >= 7
}

run eligibleStudentsForInternshipWithProficiency for 5 Int,
   exactly
    4  User, 5 CurriculumVitae, 5 Skill, 5 Address, 5 ID,
    5 Experience, 5 Internship, 6 InternshipParticipation,
    5 Feedback, 5 Username, 5 Password, 5 Email, 5 Name,
    5 Interest, 5 Attitude, 5 Description, 5 Benefit, 5 Title, 5
       Date
```

In this predicate, we focus on the relationship between students and internships, specifically matching students with internships based on the required skills. By running this predicate, we can verify that the system correctly identifies students who are qualified based on their CV and the skills needed for a specific internship.

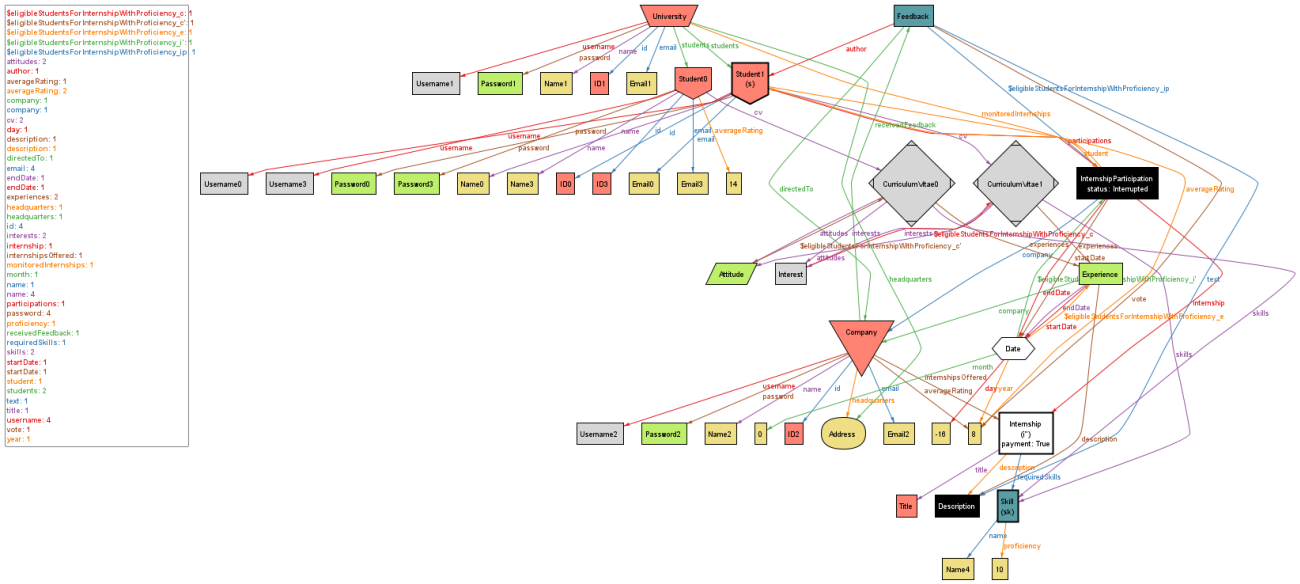


Figure1: Metamodel generated after running the predicate with proficiency

4.2.2 studentsWithNegativeFeedback predicate

```
pred studentsWithNegativeFeedback{
one u:University| some s: u.students |
                some f: Company.receivedFeedback |
                    f.author = s and f.vote < 5
    }

run studentsWithNegativeFeedback  for 5 Int, exactly
    4  User, 5 CurriculumVitae, 5 Skill, 5 Address, 5 ID,
    5 Experience, 5 Internship, 6 InternshipParticipation,
    5 Feedback, 5 Username, 5 Password, 5 Email, 5 Name,
    5 Interest, 5 Attitude, 5 Description, 5 Benefit, 5 Title, 5
       Date
```

By executing this predicate, we can identify students who are dissatisfied with their ongoing internships. This information can be valuable for university representatives to initiate communication with these students, understand their concerns, and address any issues before finalizing the internship.

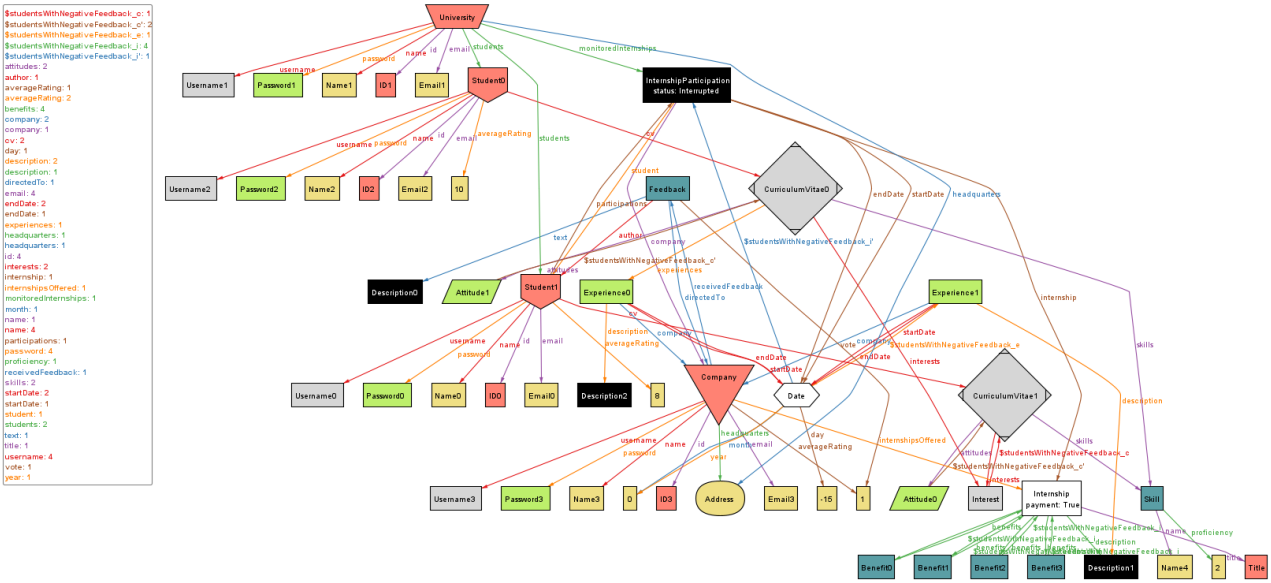


Figure2: Metamodel generated after running studentsWithNegativeFeedback

4.2.3 Metamodel

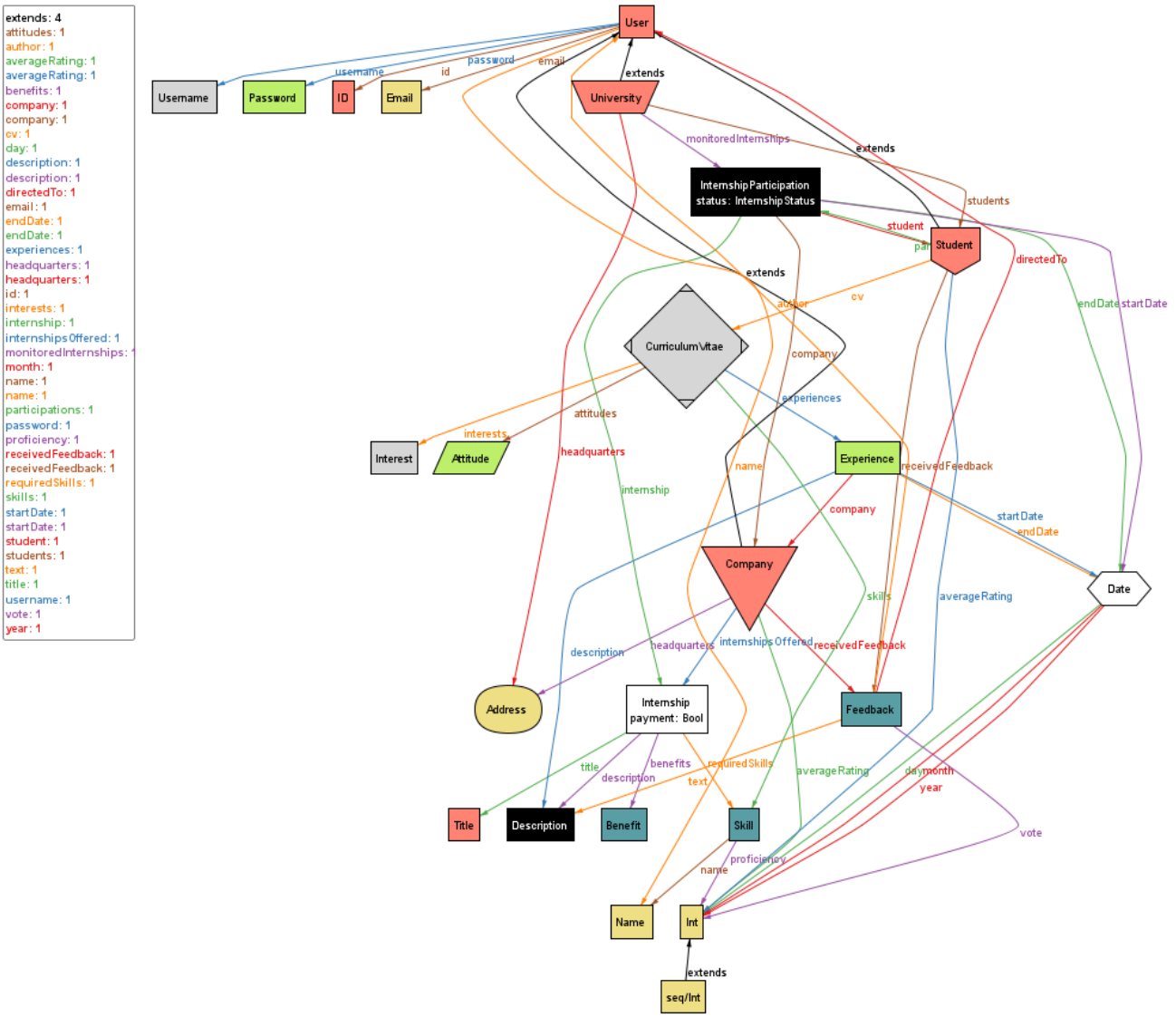


Figure3: Metamodel generated after running show metamodel

5 Effort Spent

Here we include a table indicating how many hours each component has worked in every section.

Student	Section 1	Section 2	Section 3	Section 4
Alessandro Vitobello	7h	11h	17h	6h
Jurij Diego Scandola	7h	10h	11h	23h
Francesco Raimondi	6h	12h	23h	10h

6 References

- Ministro per la Pubblica Amministrazione: European CV Format
- ISTAT: Number of University Students