POLITECNICO
MILANO 1863

SOFTWARE ENGINEERING II

Students&Companies - Requirements Analysis and Specification Document

Authors
Priuli Elia
Viceconti Veronica
Zuccoli Marco

21 Dicember 2024
Version 2

Contents

1 Introduction

1.1 Purpose

When students reach the point where they can pursue internships, it can be a critical time due to numerous challenges such as finding opportunities that also match their interests. For example, a lot of time is wasted looking for these opportunities because there are no notifications or updates on available internships. On the other hand, companies, very often, struggle to find (or don't have access to) the proper information about potential student/candidates they are looking for the position, which once again leads to wasted of time. It's with these issues in mind that S&C steps in, allowing students to search for internships they are interested in while helping companies to create internship opportunities, specifying their requirements and terms.

1.1.1 Goals

1. Make it easier for students to find internships
2. Make it easier for companies to find students
3. Support interviews
4. Monitoring internships progress

1.2 Scope

1.2.1 World Phenomena

[WP1] Creation of projects for internships
[WP2] Students undertake internships in companies
[WP3] Companies supervise students during the internship
[WP4] Students form opinions about the company where they intern
[WP5] Companies form opinions about the students
[WP6] Companies contact potential candidates

1.2.2 Shared Phenomena

World controlled
[SWC1] Internship publication by companies
[SWC2] Student's application publication
[SWC3] Students select the received match notifications
[SWC4] Companies select the received match notifications
[SWC6] Creation of interview forms by companies
[SWC7] Universities review their students' feedback to manage complications
[SWC8] Students update their CVs and/or preferences
[SWC9] Companies update their projects and/or preferences

[SWC10] Students independently search for internships
[SWC11] Students submit internship complaints
[SWC12] Companies submit internship complaints

Machine controlled
[SMC1] Request final internship feedback from students
[SMC2] Request final internship feedback from companies
[SMC3] Notify students when new compatible companies are available
[SMC4] Notify universities upon the publication of a complaint
[SMC5] Notify companies when new compatible students are available
[SMC6] Provide suggestions to students on how to write their CV
[SMC7] Provide suggestions to companies on how to publish their projects

1.2.3 Machine Phenomena

[MP1] Mechanisms of various level of sophistication to match students with internships (from simple keyword searching, to statistical analyses based on final feedbacks made by students and companies)
[MP2] Analysis of new student publications for companies
[MP3] Analysis of new internship publications for students
[MP4] Analysis of interviews to assist companies

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Acronyms

- S&C: Student&Companies
- CV: Curriculum Vitae
- HTTPS: HyperText Transfer Protocol over Secure Socket Layer
- CSRF: Cross-site Request Forgery
- XSS: Cross-site scripting
- MTFB: Mean Time Between Failures
- GDPR: General Data Protection Regulation

1.4 Reference Documents

RASD document A.Y. 2024/2025 given by the professor in the slides.

1.5 Document Structure

The document is divided into six sections, each with its unique focus, as outlined below.

Introduction: In the first section, we outline the project's goals, aims, and provide a succinct analysis of universal and shared phenomena. This section also incorporates a glossary of abbreviations and definitions necessary for understanding the issue.

Overall Description: The second section offers a detailed summary of the problem. It explores additional aspects of the domain and various scenarios involved, as well as addressing product and User attributes, assumptions, dependencies, and limitations.

Specific Requirements: The third section focuses on a thorough examination of the specific requirements. It provides comprehensive details about external interface requirements, functional specifications, and performance criteria.

Formal Analysis Using Alloy: The fourth section utilizes Alloy for conducting a formal analysis. The main aim of this chapter is to verify the correctness of the model described in the earlier sections. It emphasizes presenting the outcomes of the conducted verifications and key assertions.

Effort Spent: Section five details the individual contributions of each team member in creating this document.

References: The final section acts as a bibliography, listing the references and supplementary materials utilized in drafting this document.

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

Scenario 1: Account creation
Description: A new user (student or company) that wants to find (or publish) an internship opportunity, access the platform on the specific page and inserts all the requested data. At this point the user account is created

Scenario 2: Managing internships
Description: A company accesses a section on the platform where it can: create/modify/remove an internship opportunity (possibly facilitated by our application) and its students' preferences. When requested the company has to insert the requested data.

Scenario 3: Managing CV and internship preferences
Description: The student access its personal page and insert/modify its CV (possibly facilitated by our application) and his Internships' preferences

Scenario 4: Student search an Internship
Description: The student accesses the platform, uses the search function to explore available internships based on their interests and skills, and submits an application for the internships they are interested in.

Scenario 5: Evaluation of Internship Opportunities for a Student
Description: The student receives a notification when a match is made or when the company they applied to responds to their application. If a match notification is received, the student has the option to accept or decline the internship.

Scenario 6: Evaluation of a new student available for the internship
Description: The company receives a notification when a student with a CV that matches the requirements is available, and can decide whether to accept or decline.

Scenario 7: Feedback during internship
Description: The company and the student can provide feedback on the current progress of the internship in a dedicated section on the platform.

Scenario 8: Manage selection process
Description: After receiving the applications, the company reviews the profiles of the interested students and selects some candidates for an interview. The company can use the platform to conduct the interviews.

Scenario 9: Feedback Post-Internship
Description: At the end of the internship, after receiving a feedback request from the platform, the student evaluates the training experience and the work environment, while the company assesses the student's performance.

Scenario 10: University monitor the student's internship
Description: Once the student starts the internship, the university must monitor the progress through the platform. If the student or the company reports any issues, the university receives a notification and can intervene to resolve the problem, managing communication between the student and the company or terminating the internship if necessary.

2.1.2 Domain class diagram

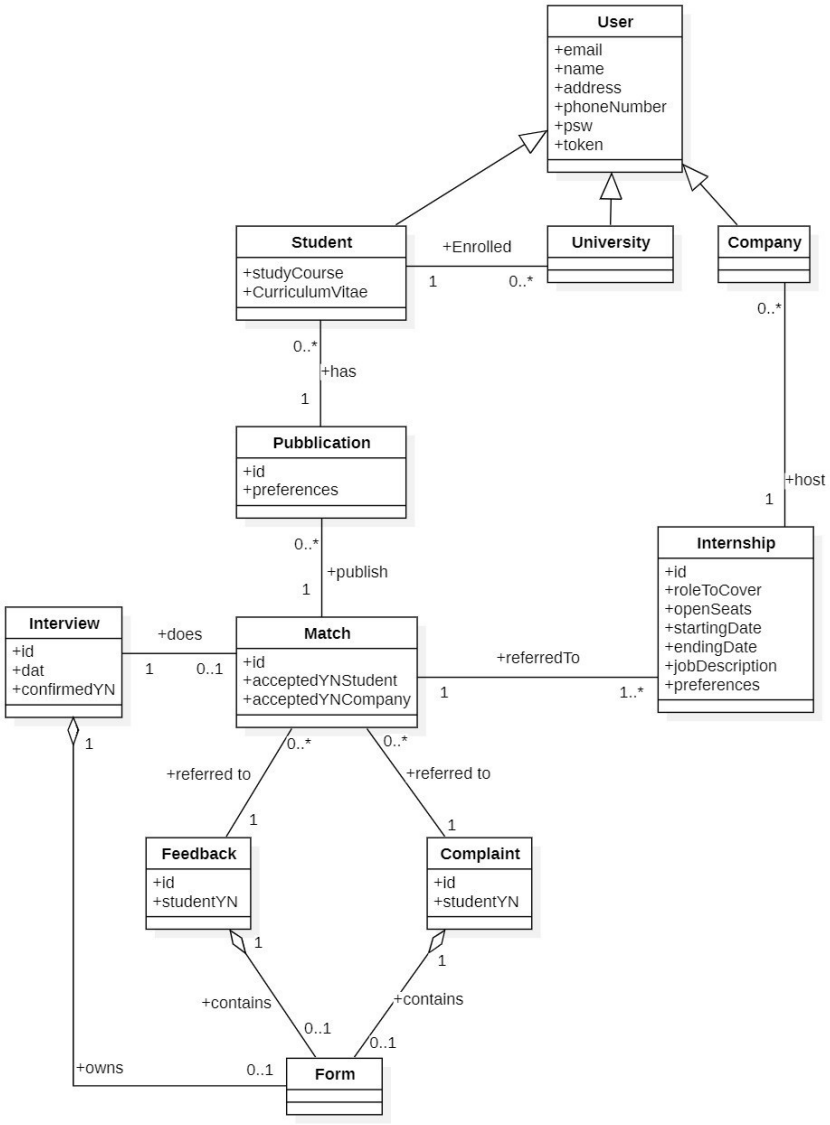

Figure 1: Class Diagram

1. User: the class that represents the general entity that interact with the system. The sub-classes Student, Company and University better represent the different types of clients in the platform. Each of them is able to visualize different part of the system and carry out different opera-

tions on it. Students upload their CV, working preferences and browse the platform scanning for an internship. Companies publish internship opportunities. Both can write feedbacks, eventual complaints and accept or decline matches found by the platform. Finally, universities address the complaints related to internships involving their students.

2. Publication: used to collect the student's information made available to compute eventual matches with internships. A student can have multiple publication because it will be possible to express different preferences. For example: Part time internship in software engineering and full time in computer security.

3. Internship : Represents an internship available at a company.

4. Match: Class useful to connect the matches that the platform identifies between students and companies' internships. It contains the information about the acceptance. In particular its attributes "AcceptedYNStudent" and "AcceptedYNCompany" are "Yes" when both student and company accept the match.

5. Interview: Represents the interaction between a company and a student during the interview process. It stores the questions asked during the interview and the corresponding answers provided by the student, which are manually entered into the system by the company representative. The system assists the company in designing and organizing the interview form, allowing all the necessary questions to be collected and structured efficiently. When the interview is done, the "ConfirmedYN" attribute is set and if it is set to "Yes", the Company has totally accepted the student and the internship can start. If it is set to "No", the Company doesn't accept the student and the Internship won't start.

6. Form: A standard structure used to represent a set of questions and the corresponding answers. It's used both to structure an interview but also to collect feedbacks and complaints (in these case there's only one question)

7. Complaint/Feedback : Used to collect feedbacks given both by companies and students about and ongoing internships. The first one is monitored by universities whereas the other ones are used to provide statistical data in order to improve the matching process.

2.1.3 State Diagram

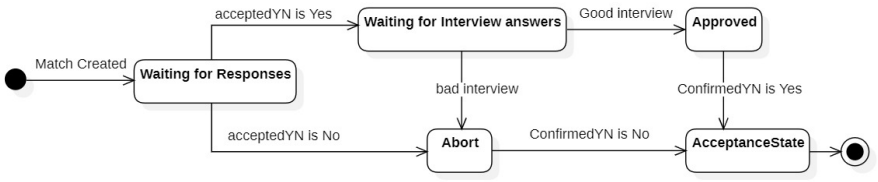

Figure 2: Match's state diagram

Figure 2 illustrates the state flow of the match entities. When a new match is identified between a student and an internship, a Match entity is created. Initially, it remains in an Idle state, awaiting decisions from both parties (Company and Student).

If both the Company and Student accept the match—indicating mutual interest—it transitions to the Waiting for Interview Answers state. In this stage, the company conducts an interview with the student.

Upon completing the interview process, the company determines whether to proceed with the internship. If the company decides to move forward, the match transitions to the Approved state, signifying a successful pairing. In this context the AcceptYNCompany and AcceptYNStudent in the match entity are set to YES and the internship starts. Conversely, if the company decides not to proceed, or if either party initially declines the match, the process transitions to an Abort state, terminating the match.

2.2 Product functions

Manage student information
It allows students, after creating their account, to create, update, and manage their profiles by entering details such as experiences, skills, abilities, and CV.

Manage company information
It allows companies, after creating their account, to create, update, and manage their profiles by entering details such as application domain, tasks to be performed, relevant adopted technologies and their terms, and the requirements they are looking for in students.

Analysis and matching generation
After the publication of profiles by students and companies, the system analyzes this data, enabling the creation of valid matches between the two.

Send notification to the users
After the creation of the profile, the system sends notifications to users to inform

them about the following areas: -New matches detected -Acceptance/rejection of internship -Student complaints to their university

Support Interview and Selection Process
After the matching, the system assists companies in the interview and selection process. In particular, companies can prepare structured questionnaires to assess the students' qualifications and suitability for the position. The system allows the interviewer to fill in the questionnaire with the student's answers during the interview.

Feedback from users during the internship
After accepting the internship, users (students and companies) can use their profile during the experience to provide ongoing feedback regarding the progress of the internship.

Management and monitoring of the internship by universities
Universities have the ability to view their students' feedback from various companies and can terminate the internship if they find it necessary following particularly negative feedback.

Feedback and Improvement Recommendations from Users
At the end of an internship, users (students and companies) can leave feedback on the completed internship. The platform uses this data to provide improvement suggestions for the corresponding profiles (for example, making internship descriptions more attractive or offering tips to improve the CV).

2.3 User Characteristics

There are mainly three types of user that interact with the platform: students, universities and companies.

2.3.1 Student

The students in this system are primarily university students who are looking for internships to gain work experience. They can be graduates or undergraduates, with varying levels of knowledge, available for part-time or full-time positions, and seeking both paid and unpaid internships.

2.3.2 Universities

Universities are the institutions where students attend their faculties, and their role, through the system, is to monitor the development of internships and ensure that no issues arise between the two parties.

2.3.3 Companies

Companies are those looking for students to include in their workforce for internships. They can be small, medium, or large in size, with or without benefits, and may require specific skills from the students.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain Assumptions

The following assumptions are made for the domain. They are properties or conditions that the system will take for granted.They must be checked to ensure a correct platform behaviour:
[D1] The students have to be at the end of their career
[D2] The CVs have to contain real data
[D3] The companies have to public only available internship
[D4] The student must have uploaded a CV in EU format to ensure the proper functioning of the application
[D5] Students and companies must have an email for communication
[D6] The student's email must be a university email

3 Specific Requirements

3.1 External Interface Requirements

The application will be developed through a WebApp.

3.1.1 User interfaces

In order to use all the functionalities provided by the platform, it must have the following interfaces:

1. Student :
    (a) An interface that allows them to view their profile with all their information
    (b) An interface that allows them to publish their preferences and CV
    (c) An interface that allows them to view companies and their available internships
    (d) An interface to manage matches (accept, reject, or view their status)
    (e) An interface to provide feedback and/or complaints
2. Company
    (a) An interface that allows them to view their profile with all their information

(b) An interface that allows them to publish their preferences and new internships

(c) An interface to manage matches (accept, reject, or view their status)

(d) An interface for managing interviews (creation, completion, acceptance, and rejection)

(e) An interface to provide feedback and/or complaints

3. University

(a) An interface to manage complaints and/or feedback from their students (including the option to prematurely terminate an internship)

Additionally, each user must have a login and sign-up page.

3.1.2 Hardware interface requirements

As for hardware, a common device is required that can connect to the internet and process a standard dynamic WebApp.

3.1.3 Software interface

A browser with sufficient capabilities to process JavaScript is required **in case some framework are used**.

3.1.4 Communication Interfaces

The communication interface required by the system is the HTTPS protocol because it ensures data encryption, and everything will be managed entirely within the Web App.

3.2 Functional requirements

3.2.1 Requirements

[R1] The system shall allow unregistered users to create an account on S&C only if their university already has an account on the platform.
[R2] The system shall allow registered users to log in to their accounts.
[R3] The system shall allow the universities to request an account on the platform.
[R4] The system shall allow the companies to publish their internships
[R5] The system shall allow the companies to modify their internships
[R6] The system shall allow the companies to delete their internships
[R7] The system shall allow the students to publish their CVs and their internships' preferences
[R8] The system shall allow the students to modify their CVs and their internships' preferences

[R9] The system shall allow the students to delete their CVs and their internships' preferences
[R10] The system shall allow the students to view all the matches found
[R11] The system shall allow the companies to view all the matches found
[R12] The system shall allow the companies and the students to accept or decline the proposed matches
[R13] The system shall allow the companies to create the interview form.
[R14] The system shall allow the companies to add responses of the students during the interview, to the form.
[R15] The system shall allow the universities to manage the complaints which involves their students
[R16] The system shall allow the universities to stop the on going internships involving their students
[R17] The system shall allow the students to browse through the available internships
[R18] The system shall allow the students to directly make an application to the available internships
[R19] The system shall allow the students to send complaints
[R20] The system shall allow the companies to send complaints
[R21] The system shall allow the students to give the final feedback
[R22] The system shall allow the companies to give the final feedback
[R23] The system shall notify the students whenever a new match is found
[R24] The system shall notify the university whenever a new complaint involving one of its students is submitted
[R25] The system shall notify the companies whenever a new match is found
[R26] The system shall suggests students on how to improve their CVs
[R27] The system shall suggest companies on how to improve their internships proposal
[R28] The system shall analyse the students pubblications in order to find a match with the right companies
[R29] The system shall analyse the companies pubblications in order to find a match with the right students
[R30] The system shall create matches using different level of sophistication, from simple keyword-search to statistical analyses based on final feedbacks made by students and companies
[R31] The system shall allow the universities to see their own students and their ongoing internships

3.2.2 Use Case Diagram

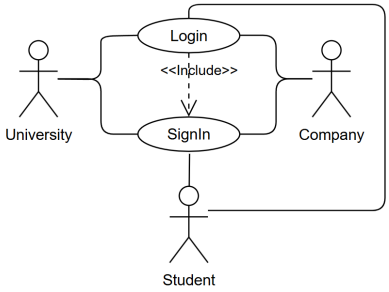

Figure 3: SignIn and Login UseCase Diagram

This diagram describe the use case that rapresent the creation profile and the login "al" profile, and this use case is mandatory for all the use cases that follows. In the following subsection there will be more detail on this.

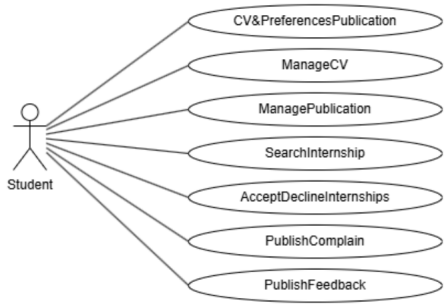

Figure 4: Student UseCase Diagram

This diagram describe the student use cases inside S&C. In the following subsection there will be more detail on this.

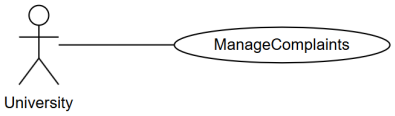

Figure 5: University UseCase Diagram

This diagram describe the university use cases inside S&C. In the following

subsection there will be more detail on this.

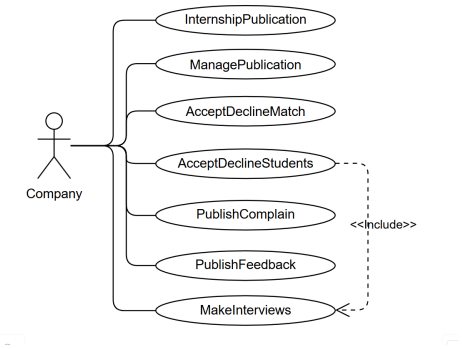

Figure 6: Company UseCase Diagram

This diagram describe the company use cases inside S&C. In the following subsection there will be more detail on this.

3.2.3 Use Cases Description

[UC1] SignIn

Actor(s)	User
Entry Condition	The user accesses the platform and click the signIn button
Event Flow	1. The user visualizes the registration form 2. The user fill-in the registration form (name,surname and CV if needed, phone-number, address,email,password) 3. The user click on the send button, sending the registration form
Exit Condition	The user is registered and can access the platform
Exceptions	If provided information is incomplete or invalid, the system prompts the user to correct errors with a human readable message.

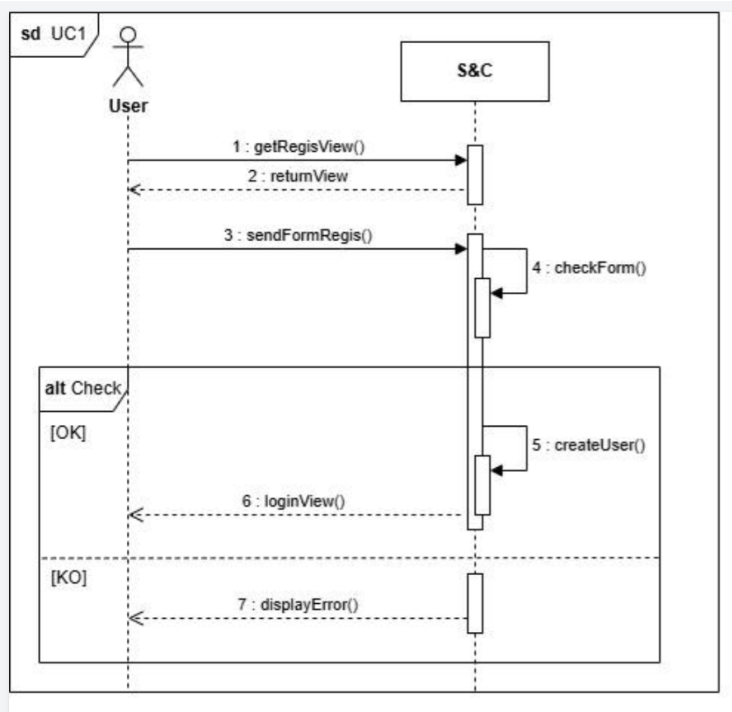

Figure 8: "Registration" Sequence Diagram

[UC2] Login

Actor(s)	User
Entry Condition	The user accesses the platform and has already created an account
Event Flow	1. The user visualizes the login form 2. The user fill-in the login form (email,password) 3. The user click on the enter button, entering in the platform
Exit Condition	The user has accessed to the platform and can, now, navigate inside the platform
Exceptions	If provided information is incomplete or invalid, the system prompts the user to correct errors with a human readable message.

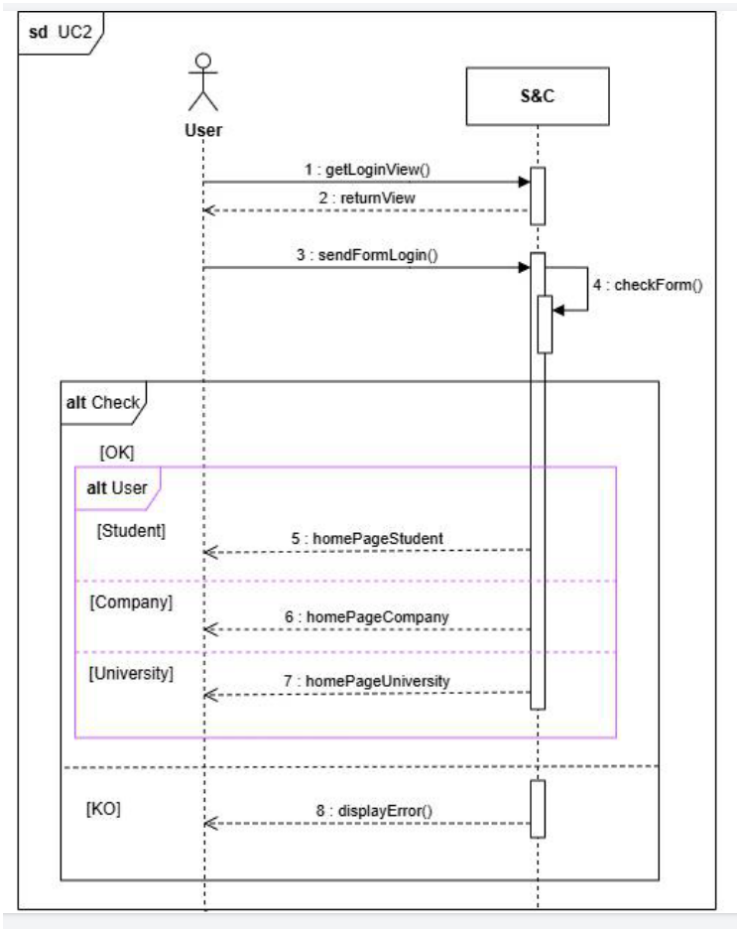

Figure 9: "Login" Sequence Diagram

STUDENT

[UC3] CV&PreferencesPublication

Actor(s)	Student
Entry Condition	The student has logged in the platform
Event Flow	1. The user clicks the button or the tab that allows him to publish his internship search announcement 2. The user insert his CV and send it to the server, which elaborates and sends back the suggestions. Then the student can modify and confirm the updated cv or the previous one. Later, the student enters their internship preferences description. 3. The user clicks the button to publish the announcement. 4. The system publishes the announcement and redirects the user to the homepage, then it starts the matchmaking.
Exit Condition	The user has published the announce and has returned to the homepage
Exceptions	If provided information is incomplete, the system prompts the user to add the missing informations with a human readable message.

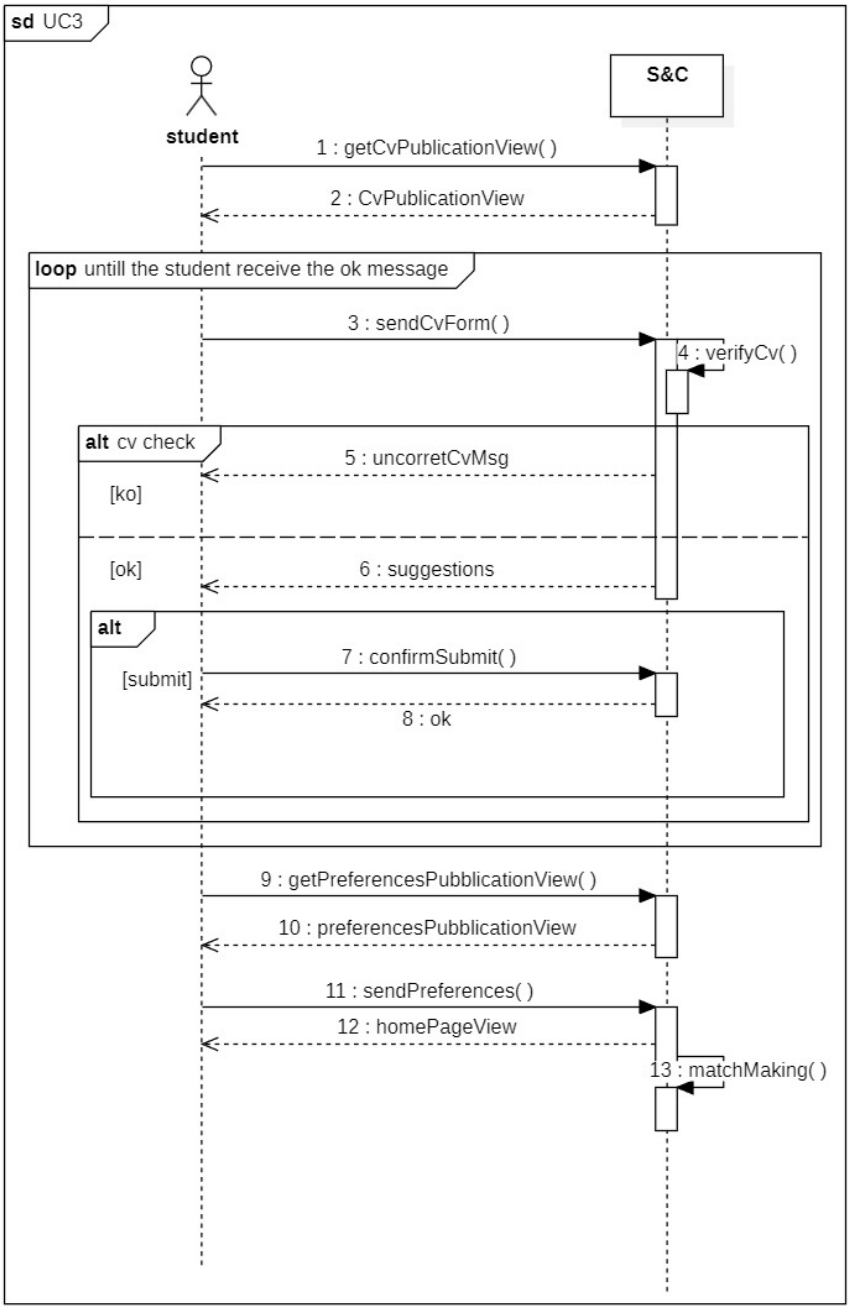

Figure 10: "CV&PublicationPrefernces" Sequence Diagram

[UC4] ManageCV

Actor(s)	Student
Entry Condition	The student has logged in the platform and has added an announce
Event Flow	1. The user clicks the button that allows them to modify their CV 2. The user modifies, updates (with system's suggestions) or deletes their CV 3. The user clicks the button to confirm or cancel the changes just made 4. The system updates their data in order to have right information for the matching 5. The system redirects the Student to the homepage
Exit Condition	The user has published the modified CV and has returned to the homepage
Exceptions	If provided information is incomplete, the system prompts the user to add the missing informations with a human readable message.

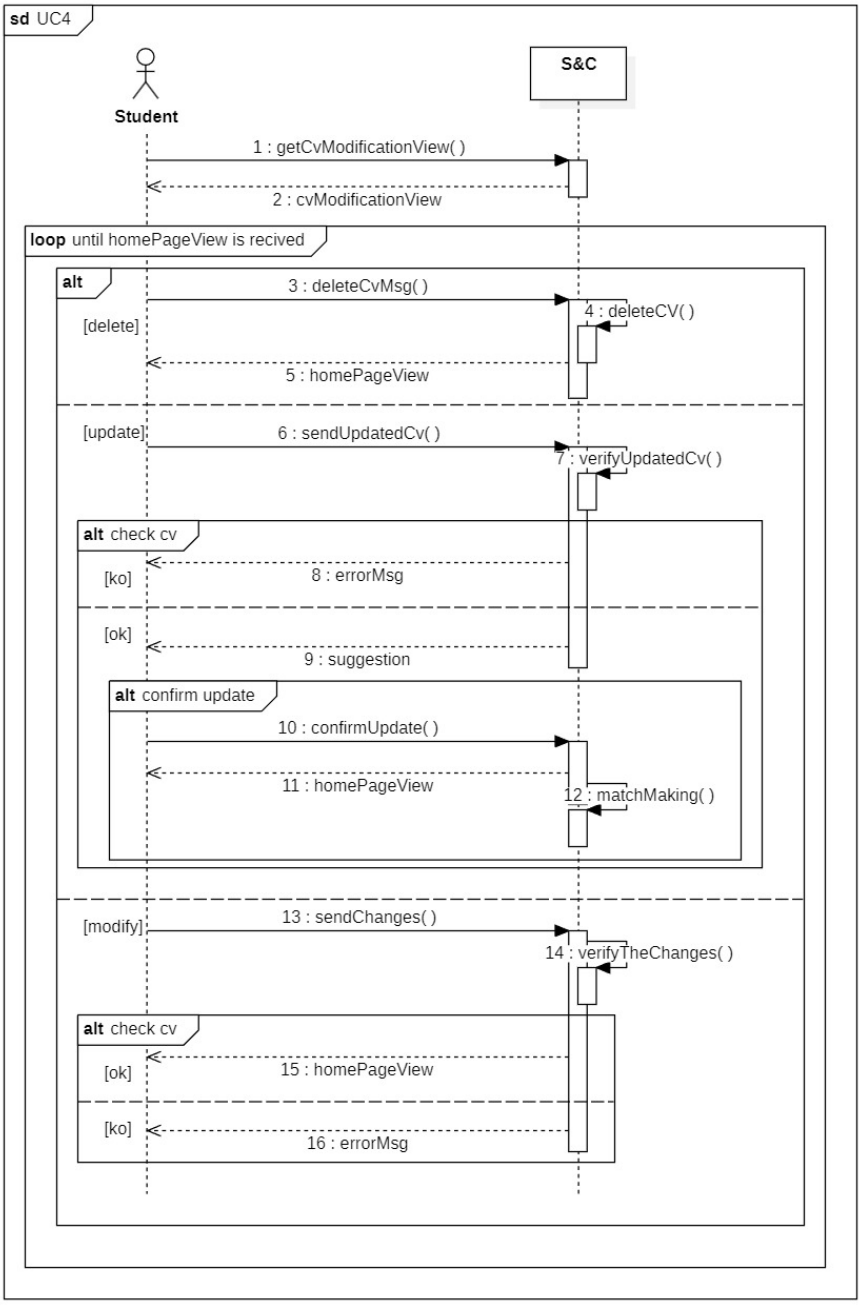

Figure 11: "ManageCV" Sequence Diagram

[UC5] ManagePublication

Actor(s)	Student
Entry Condition	The student has logged in the platform and has added an announce
Event Flow	1. The user click the button that allow him to modify a particular publication 2. The user modify or delete that publication 3. The user click the button to confirm or cancel the changes just made 4. The system update his data in order to have right information for the matching 5. The system redirect the Student to the homepage and then start the match making.
Exit Condition	The user has modified the publication and return to the homepage
Exceptions	None

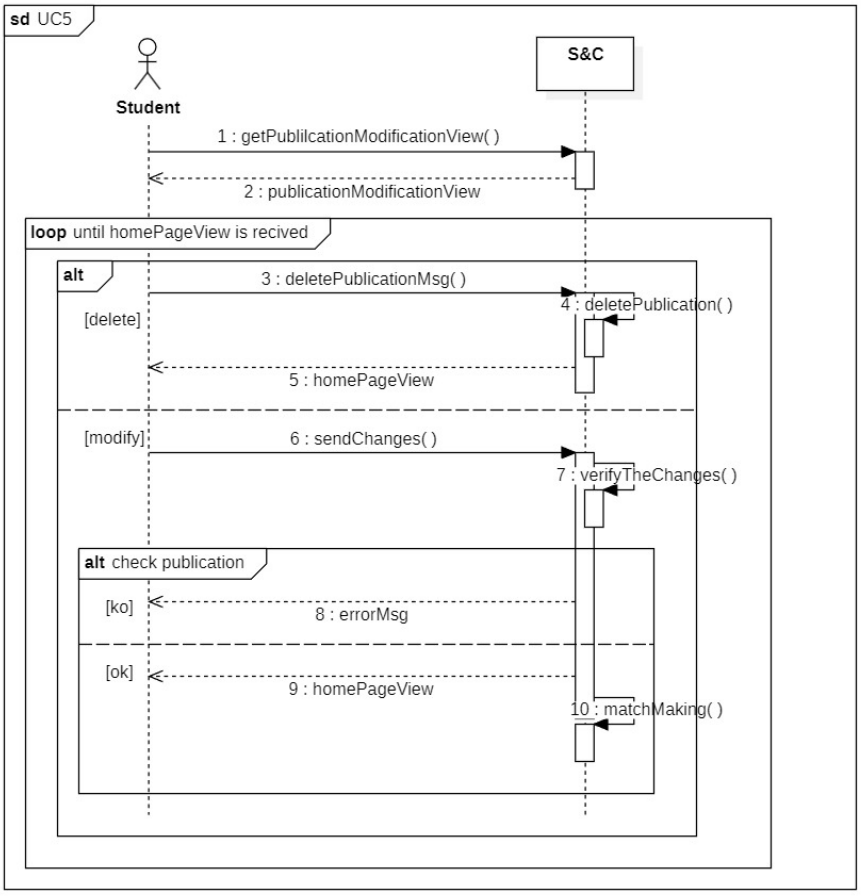

Figure 12: "ManagePubblications" Sequence Diagram

[UC6] Search Internship

Actor(s)	Student, Company
Entry Condition	The student has logged in the platform and has added an announce
Event Flow	1. The user scrolls the homepage and find an interesting internship 2. The user clicks the button to apply for that internship 3. The system sends a notification to the corresponding Company to let it know about the new match found 4. The system redirects the Student to the homepage
Exit Condition	The user has made an application and is now in the homepage
Exceptions	None

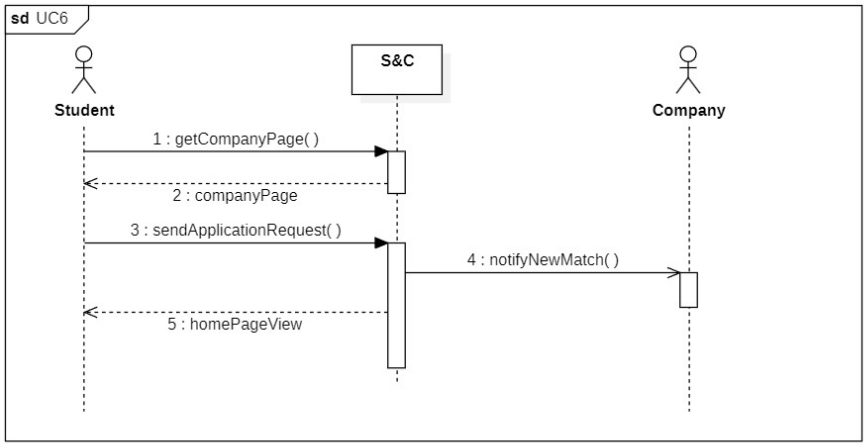

Figure 13: "SearchInternships" Sequence Diagram

[UC7] AcceptDeclineInternships

Actor(s)	Student, Company
Entry Condition	The student has logged in the platform and has added an announce
Event Flow	1. The student receives a notification about a new match found (optional) 2. The student goes to the section where he can view all the matches found (optional) 3. The student clicks on a match found 4. the student selects a match 5. The student clicks the button to accept or decline a particular match 6. The system sends a notification to the corresponding Company to let it know about the Student's decision
Exit Condition	The user has accepted or declined a match
Exceptions	None

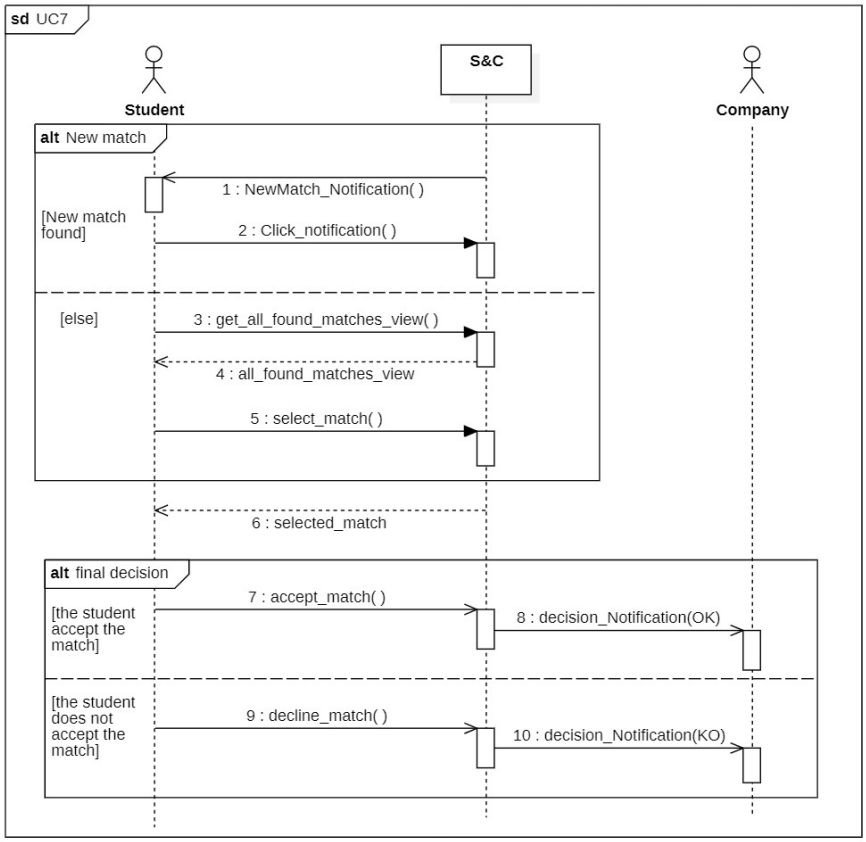

Figure 14: "AcceptDeclineInternships" Sequence Diagram

[UC8] PublishComplain

Actor(s)	Student, University
Entry Condition	The student has logged in the platform and has an ongoing internship
Event Flow	1. The user goes to the section where he can view his internship 2. The user click the internship he want to create a complain about 3. The user write the complain 4. The user click the confirm or cancel the complain 5. The system send a notification to his University to let it know about the Student's complain 6. The system redirect the Student to the homepage
Exit Condition	The user has uploaded the complain
Exceptions	If the text written is empty, the system prompts the Student to add a text with a human readable message.

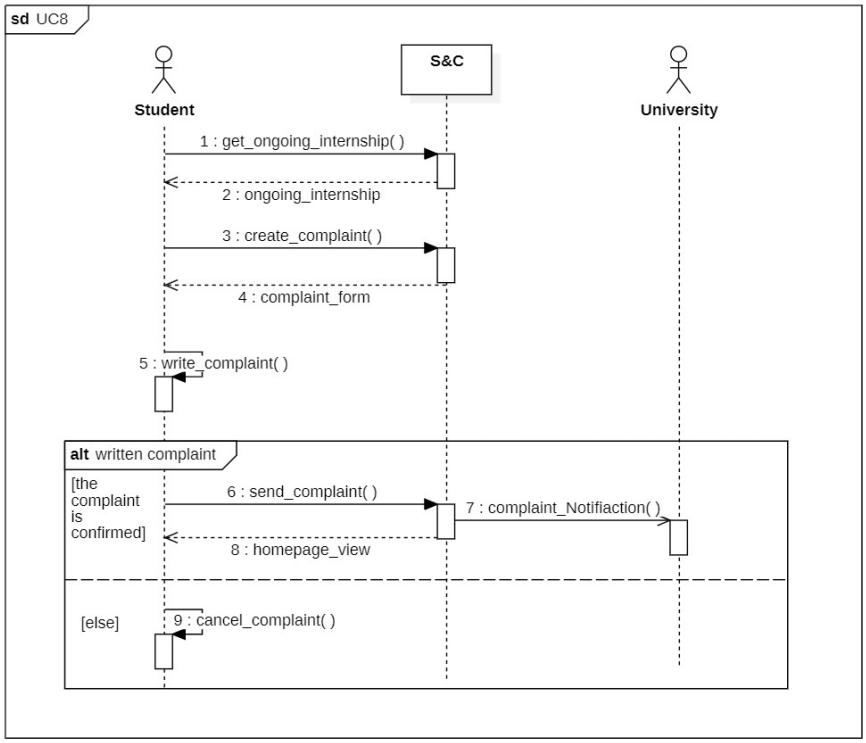

Figure 15: "PublishComplain" Sequence Diagram

[UC9] PublishFeedback

Actor(s)	Student, Company
Entry Condition	The student has logged in the platform and has finished an internship
Event Flow	1. The user receive a notification from the system to write a feedback about the internship just finished (optional) 2. The user click the button to write the feedback 3. The user write the feedback 4. The user click the button to confirm the feedback 5. The system collects the feedback 6. The system redirect the Student to the homepage
Exit Condition	The user has uploaded the feedback
Exceptions	If the text written is empty, the system prompts the Student to add a text with a human readable message.

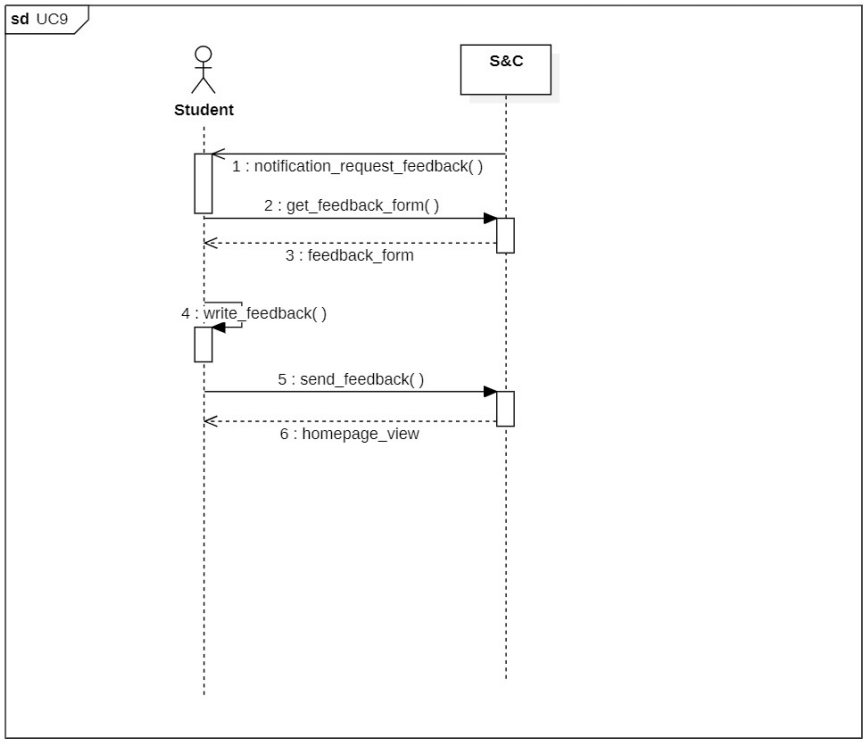

Figure 16: "PublishFeedback" Sequence Diagram

UNIVERSITY

[UC10] ManageComplains

Actor(s)	University
Entry Condition	The university has logged in the platform and has at least one student subscribed too
Event Flow	1. The university receive a notification from the system about a new complain wrote by one of its students 2. The university click on the terminate or cancel button deciding if it's better to conclude or not the internship 3. The system update the correspond data 4. The system redirect the university to the homepage
Exit Condition	The university has chosen what to do
Exceptions	None

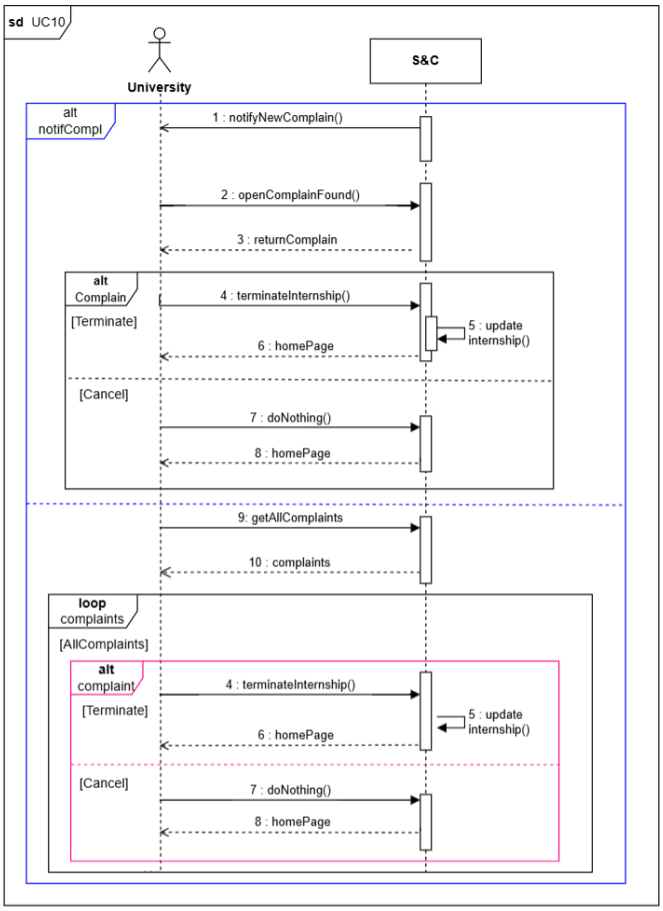

Figure 17: "ManageComplains" Sequence Diagram

COMPANY

[UC11] InternshipPublication

Actor(s)	Company
Entry Condition	The company has logged in the platform
Event Flow	1. The user clicks the tab that allows him to publish his announcement regarding the new internship available 2. The user inserts the project and submits it to the system, which computes the suggestions and send them back. Then the user can modify the project or confirm the previous version. Finally the the user writes a description about the student's requirements 3. The user clicks on the button to publish the announcement 4. The system publishes the announcement and redirects the user to the homepage
Exit Condition	The user has published the announce and has returned to the homepage
Exceptions	If provided information is incomplete, the system prompts the user to add the missing informations with a human readable message.

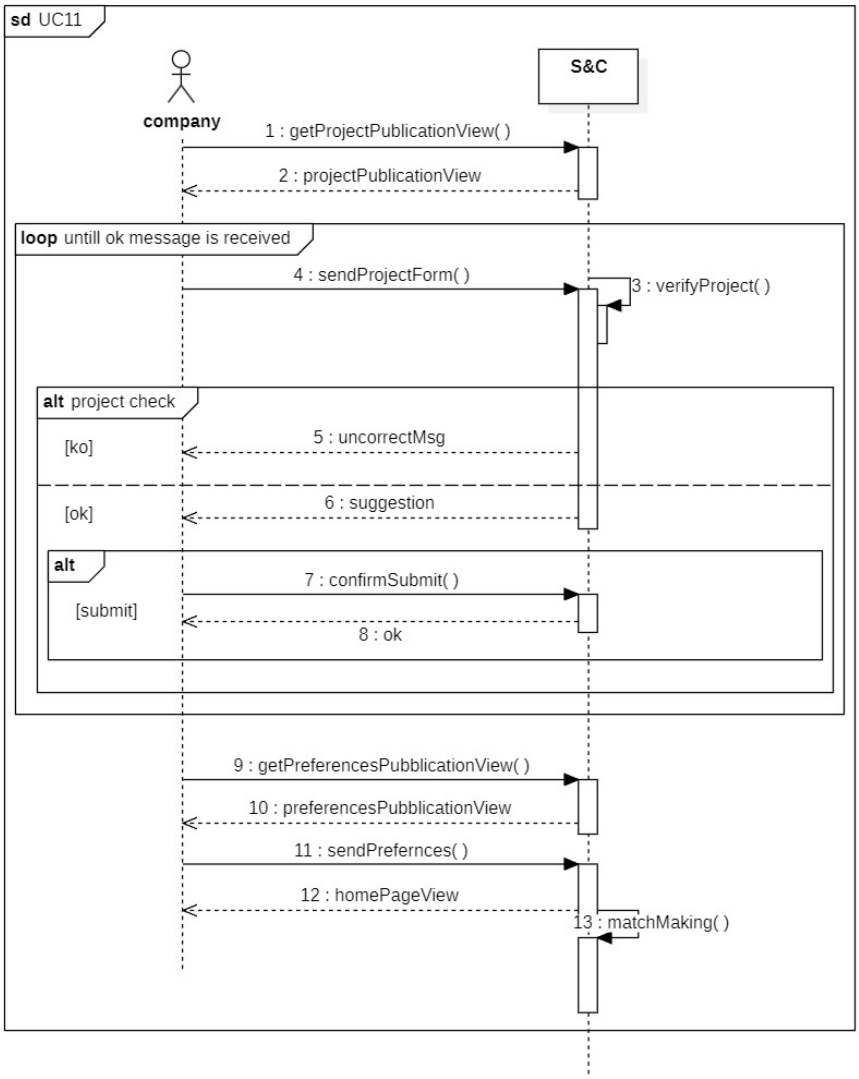

Figure 18: "InternshipPublication" Sequence Diagram

[UC12] ManagePublication

Actor(s)	Company
Entry Condition	The company has logged in the platform and has added an announce
Event Flow	1. The user clicks the button that allows him to modify a particular publication 2. The user modifies, update (with system's suggestions) it or deletes that publication, also with the help the system 3. The user clicks the button to confirm or cancel the changes just made 4. The system updates his data in order to have the right information for the matching 5. The system redirect the user to the homepage
Exit Condition	The user has modified the publication and return to the homepage
Exceptions	None

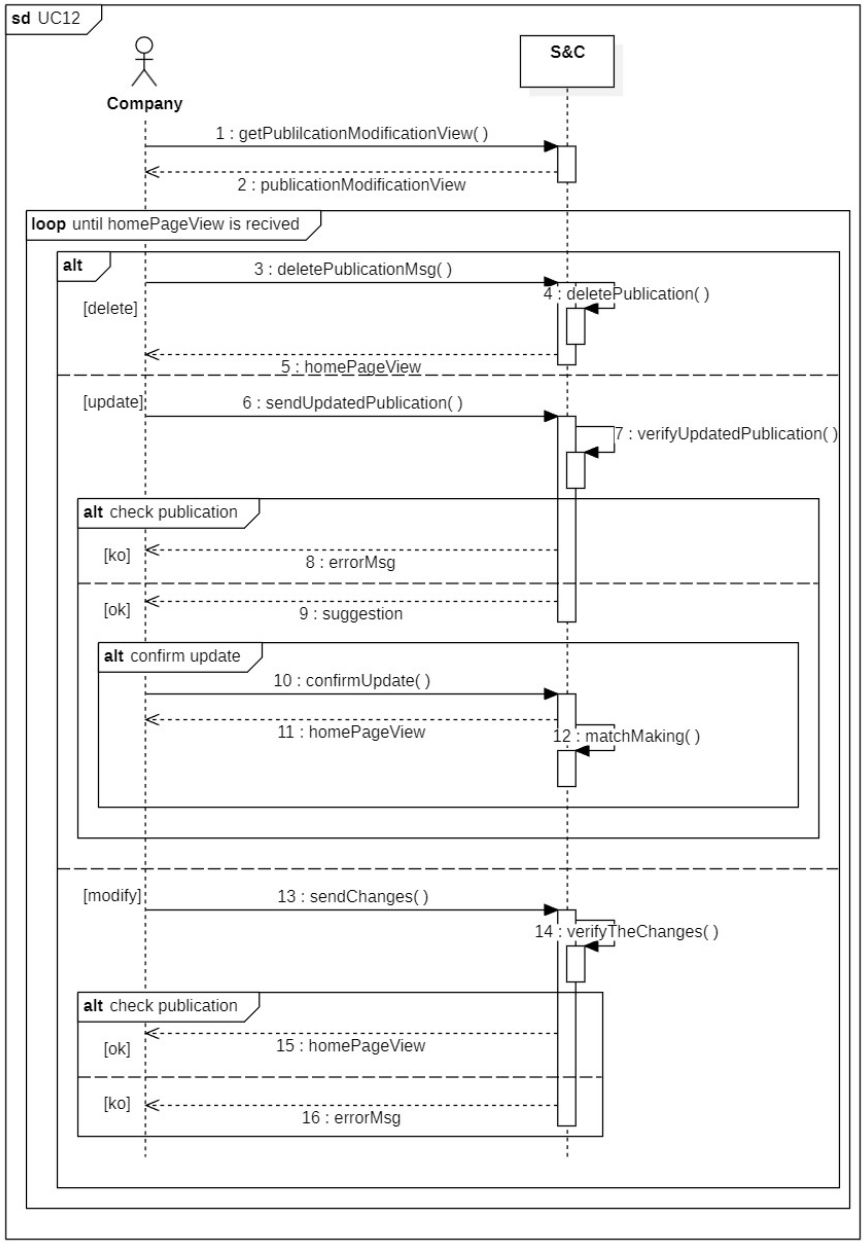

Figure 19: "ManagePublication" Sequence Diagram

[UC13] AcceptDeclineMatch

Actor(s)	Company,Student
Entry Condition	The company has logged in the platform and has added an announce
Event Flow	1. The company receive a notification about a new match found (optional) 2. The company goes to the section where it can view all the matches found (optional) 3. The company click on a match found 4. The company click the confirm or cancel button 5. The system send a notification to the corresponding Student to let it know about the Company's decision
Exit Condition	The company has accepted or declined the student after the match found
Exceptions	None

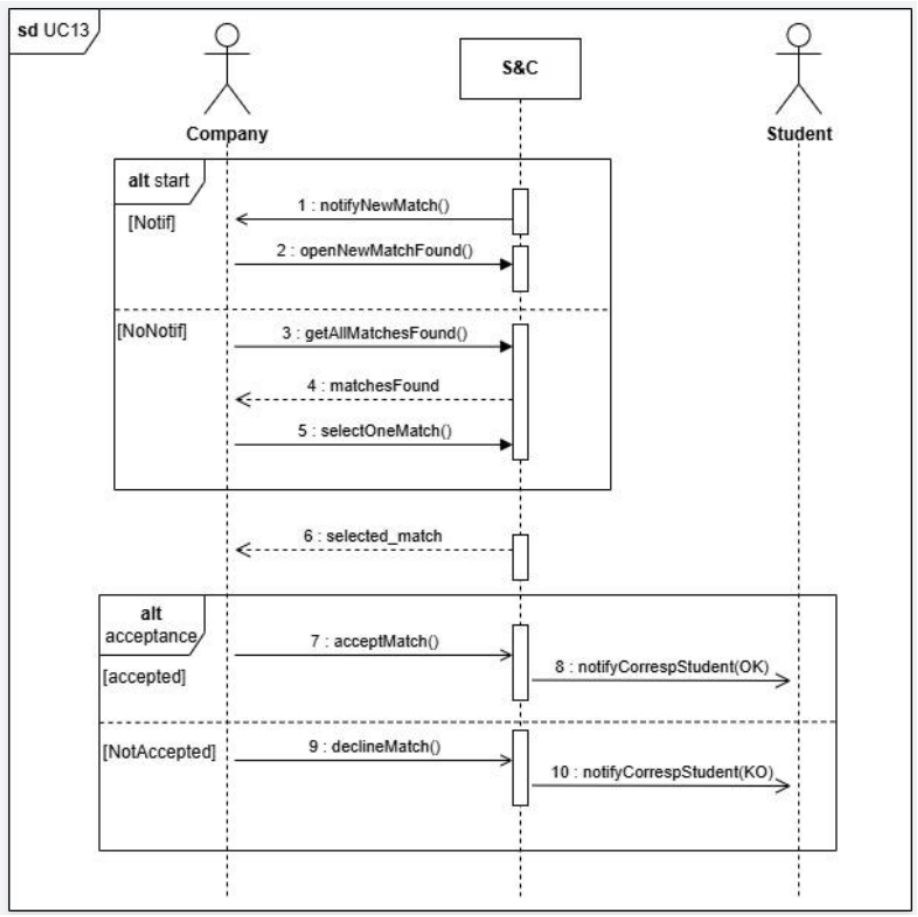

Figure 20: "AcceptDeclineMatch" Sequence Diagram

[UC14] AcceptDeclineStudent

Actor(s)	Company,Student
Entry Condition	The company has logged in the platform,has added an announce, has accepted a match and created an interview
Event Flow	1. The company goes to the platform's section where all the made interviews are visible 2. The company select an interview 3. The company click the accept/decline button 4. The system sends a notification to the corresponding Student to let it know about the Company's final decision
Exit Condition	The company has accepted or declined the student after the interview
Exceptions	None

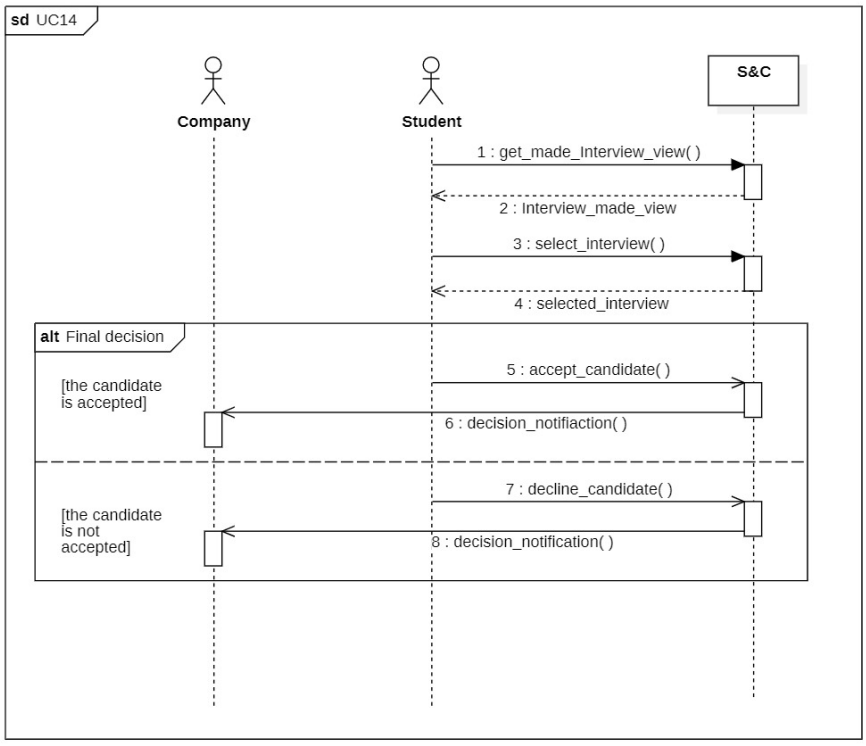

Figure 21: "AcceptDeclineStudent" Sequence Diagram

[UC15] PublishComplaint

Actor(s)	Company,University
Entry Condition	The company has logged in the platform and has an ongoing internship
Event Flow	1. The company goes to the section where it can view his ongoing internships 2. The company selects the wanted internship 3. The company clicks internship to create a complain about 4. The company writes the complain 5. The company confirm or cancel the complain 6. The system sends a notification to the student's university to inform them about the company's complaint regarding the internship. 7. The system redirect the Company to the homepage
Exit Condition	The user has uploaded the complain
Exceptions	If the text written is empty, the system prompts the Student to add a text with a human readable message.

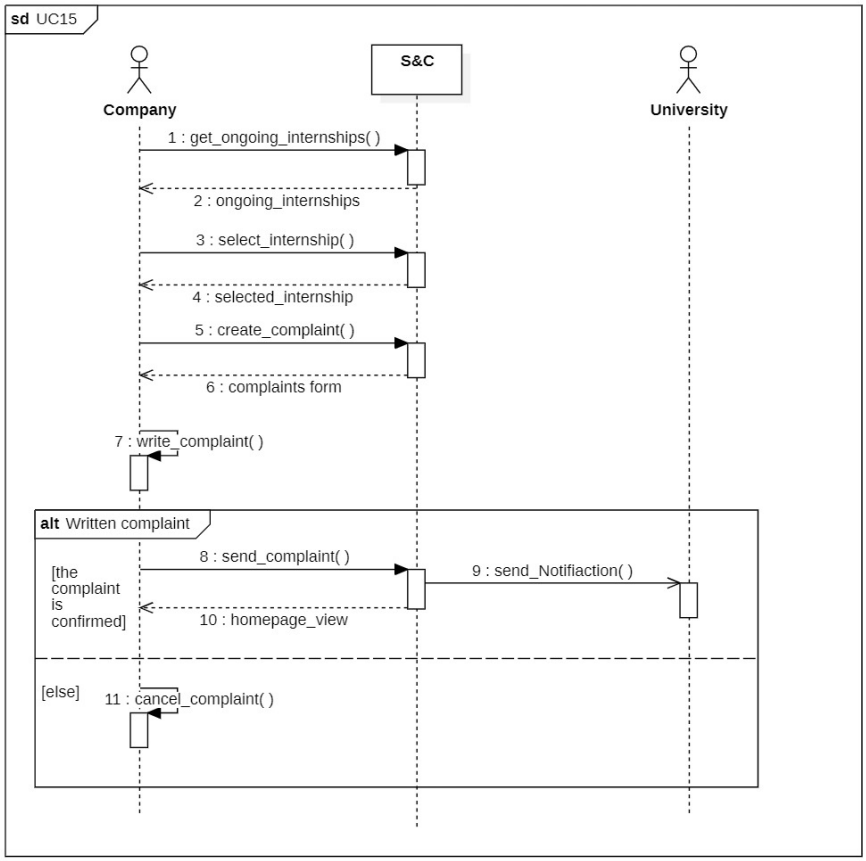

Figure 22: "PublishComplain" Sequence Diagram

[UC16] PublishFeedback

Actor(s)	Company
Entry Condition	The Company has logged in the platform and has ended an internship
Event Flow	1. The company receives a notification from the system to write a feedback about the internship just ended (optional) 2. The company clicks the button to write the feedback 3. The company writes the feedback 4. The company clicks the button to confirm the feedback 5. The system collects the feedback 6. The system redirects the company to the homepage
Exit Condition	The user has uploaded the feedback
Exceptions	If the text written is empty, the system prompts the Student to add a text with a human readable message.

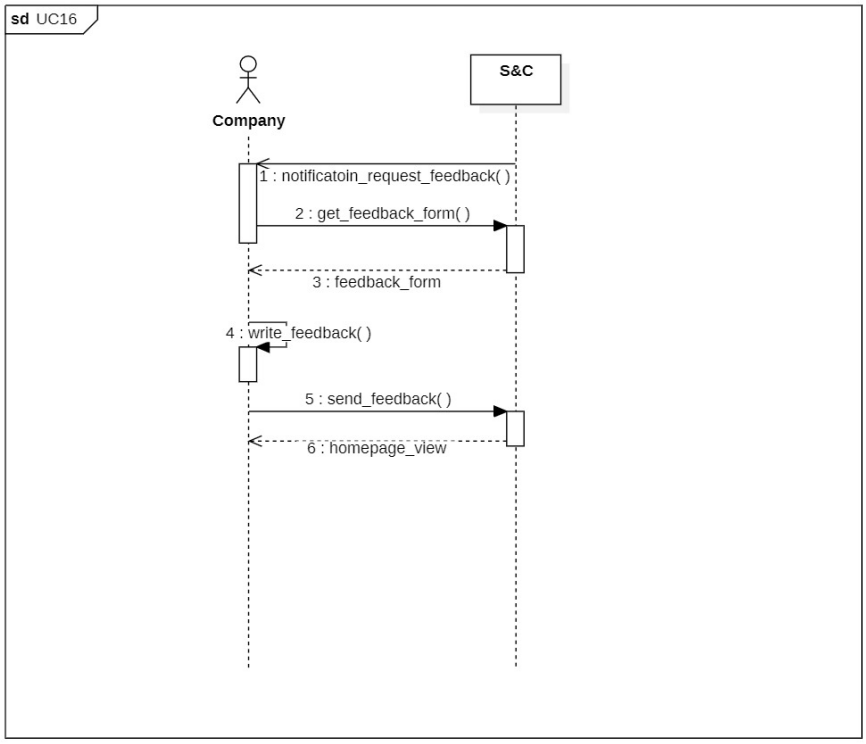

Figure 23: "PublishFeedback" Sequence Diagram

[UC17] MakeInterviews

Actor(s)	Company,Student
Entry Condition	The Company has logged in the platform and has accepted a match
Event Flow	1. The company creates the interview form 2. The company fill-in the form with the Student's responses 3. The company click the button to finish the interview 4. The system collects the interview 5. The system redirect the user to the homepage
Exit Condition	The user has uploaded the interview
Exceptions	If the text written is empty, the system prompts the Student to add a text with a human readable message.

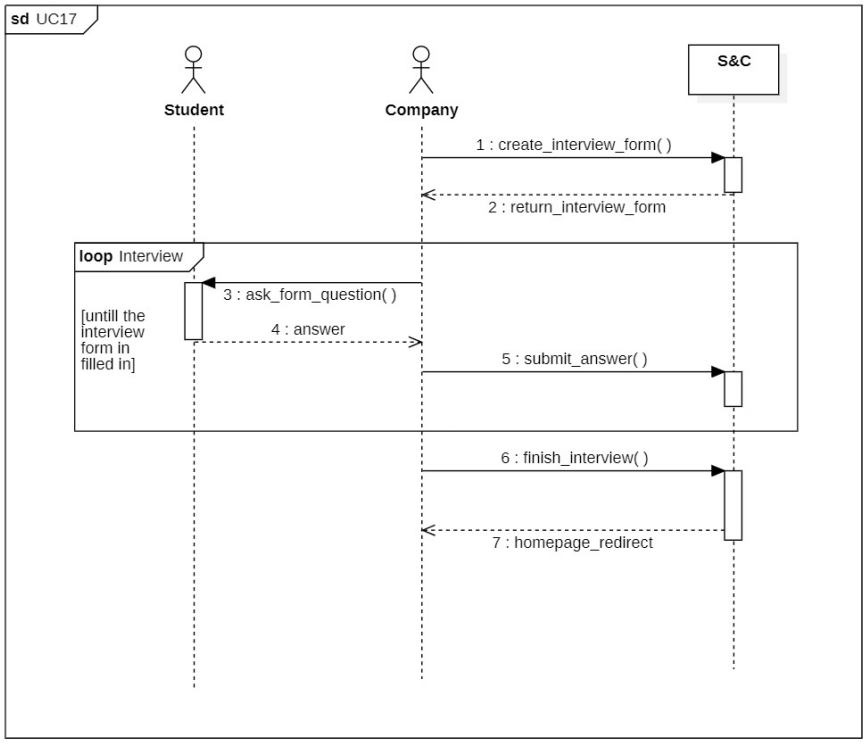

Figure 24: "MakeInterview" Sequence Diagram

3.2.4 Mapping on requirements

G1: Facilitate internship searches for students
Requirements mapping
[R1] The system shall allow unregistered users to create an account on S&C only if their university already has an account on the platform. [R2] The system shall allow registered users to log in to their accounts. [R7] The system shall allow the students to publish their CVs and their preferences delle internship [R8] The system shall allow the students to modify their CVs and their preferences delle internship [R9] The system shall allow the students to delete their CVs and their preferences delle internship [R10] The system shall allow the students to view all the matches found [R12] The system shall allow the companies and the students to accept or decline the proposed matches [R17] The system shall allow the students to update their CVs [R18] The system shall allow the students to browse through the available internships [R19] The system shall allow the students to directly make an application to the available internships [R20] The system shall allow the students to send complaints [R22] The system shall allow the students to give the final feedback [R24] The system shall notify the students whenever a new match is found [R27] The system shall suggests students on how to improve their CVs [R29] The system shall analyse the students pubblications in order to find a match with the right companies [R31] The system shall create matches using different level of sophistication, from simple keyword-search to statistical analyses based final feedbacks made by students and companies
Domain assumption mapping
[D5] Students and companies must have an email to communicate. [D1] The students have to be at the end of their career [D2] The CVs have to contain real data [D4] The student must have uploaded their CV in EU format to ensure the proper functioning of the application. [D6] The student's email must be a university email

Table 1: Mapping on first goal

G2: Facilitate student searches for companies
Requirements mapping
[R1] The system shall allow unregistered users to create an account on S&C only if their university already has an account on the platform. [R2] The system shall allow registered users to log in to their accounts. [R4] The system shall allow the companies to publish their internships [R5] The system shall allow the companies to modify their internships [R6] The system shall allow the companies to delete their internships [R11] The system shall allow the companies to view all the matches found [R12] The system shall allow the companies and the students to accept or decline the proposed matches [R21] The system shall allow the companies to send complaints [R23] The system shall allow the companies to give the final feedback [R26]The system shall notify the companies whenever a new match is found [R28] The system shall suggest companies on how to improve their internships proposal [R30] The system shall analyse the companies pubblications in order to find a match with the right students [R31] The system shall create matches using different level of sophistication, from simple keyword-search to statistical analyses based on final feedbacks made by students and companies
Domain assumption mapping
[D5] Students and companies must have an email to communicate. [D3] The companies have to public only available internship

Table 2: Mapping on second goal

G3: Interviews support
Requirements mapping
[R13] The system shall allow the companies to create the interview form. [R14] The system shall allow the companies to add responses of the students during the interview, to the form.
Domain assumption mapping

Table 3: Mapping on third goal

G4: Monitoring internship progress
Requirements mapping
[R1] The system shall allow unregistered users to create an account on S&C only if their university already has an account on the platform. [R2] The system shall allow registered users to log in to their accounts. [R3] The system shall allow the universities to request an account on the platform. [R15] The system shall allow the universities to manage the complaints which involves their students [R16] The system shall allow the universities to stop the on going internships involving their students [R25] The system shall notify the universities whenever a new complaint involving one of its students is submitted [R32] The system shall allow the universities to see their own students and their ongoing internships
Domain assumption mapping

Table 4: Mapping on fourth goal

3.3 Performance Requirements

Number of concurrent Users: WebApps with a similar goal of S&C, has million of Users. To guarantee this, means S&C should be able to handle up at least 10.000 Users simultaneously. This is important for making sure our WebApp works well for a good number of people, giving them a smooth and enjoyable experience.
Data storage: S&C needs to save and manage all the details about Students, Universities and Companies. Additionally, the database needs to process queries in an efficient way in order to allow user to view all the information they need in short time.
Time response: Every operation that is directly executed by S&C, i.e. register, login, create, modify and evaluate, should be in the domain of milliseconds.
Resources: S&C needs to scale its resources up or down based on demand, utilizing cloud based services and load balancing techniques.

3.4 Design Constraints

3.4.1 Standard compliance

The application should follow all the specific regulations in matter of digital services such as GDPR and the most recent European Digital Service Act. These regulations guarantee safety and transparency over how user data is handled, how content is managed and how the application works (for example explaining how matches are done)

3.4.2 Hardware limitations

Being a webApp, the application should be compatible with a large number of devices, from PCs to smartphones. In order to operate correctly a recent version of one of the most used browser (Chrome, Edge, Firefox, Safari, etc) is required. Server-side, the hardware must be chosen in order to effectively store all the current and historical information and to provide a response time ¡ 2 sec even under heavy load.

3.4.3 Any other constraints

The development of the application should consider that it will be used by a large number of people, thus it should be structured with usability in mind. For example it should be easy to use, accessible and should support multi-language interface. Moreover, the entire infrastructure should be designed to be scalable. Finally the matching system should be ethical. This means that the matching algorithm should be neutral and not only based on the students' expertises and internships characteristics but also on final feedbacks made by students and companies (used for statistical analyses matching).

3.5 Software System Attributes

3.5.1 Reliability

The MTFB of the S&C should be at least of 1 month and the MTTB should be at most one hour in order to be better than the expected below constraint. To achieve this reliability we need to implement the application on a set of replicated server, to be resilient in case of failure.

3.5.2 Availability

The needed availability should be at least the 99.7% to have one day down in a span of a year, if the above constraint in reliability will be reached the probable availability will be more than 99.8%

3.5.3 Security

The S&C system has to be implemented over HTTPS to communicate in a secure manner with the central server, with the machine in charge of the communication placed in a DMZ to protect the DB. The system must to be able to avoid the SQL INJECTION, the CSRF and the XSS attacks.

3.5.4 Maintainability

The S&C system should be designed in a modular concept in order to be maintainable and fixed with ease. In this context each part of the software must be available through APIs. It is also needed to keep the documentation up to date and run automated tests after each update or fix.

3.5.5 Portability

The system should be designed to work on all the common browser (like edge, Chrome, Safari and Firefox etc.) also on the mobile device (like smartphone and tablet).

4 Formal Analysis Using Alloy

```
enum Accepted {Yes, No}

abstract sig User{
}


sig Student extends User{
    uni: one University
}

sig University extends User{
}

sig Company extends User{

}

sig Publication{
    student : one Student
}

var sig Match {
    var acceptedYN: one Accepted,
    var pub : one Publication,
    var ref : one Internship,
    var feedbacks : set Feedback,
    var complaints : set Complaint
}

sig Internship {
    hosted : one Company,

}

var sig Interview {
    var match : one Match,
    var confirmedYN: one Accepted
```

```
}

var sig Complaint {
    var writer : one User
}

var sig Feedback {
    var writer : one User
}


//there is no feedback pointed from more Matches
fact {
    always (no m1, m2: Match | some f : Feedback|
        f in m1.feedbacks and f in m2.feedbacks and m1 != m2)
}

//there is no complaint pointed from more internships
fact {
    always (no m1, m2: Match | some c : Complaint|
        c in m1.complaints and c in m2.complaints and m1 != m2)
}

//all the feedback must have an internship
fact {
    always (all f : Feedback | some m : Match |
        f in m.feedbacks  )
}

//all the complaint must have an internship
fact {
    always (all c : Complaint | some m : Match |
        c in m.complaints)
}


//check the number of writer
fact{
    always (all m : Match |
        #m.feedbacks <= 2 and #(m.feedbacks.writer & Student) < 2 and
#(m.feedbacks.writer & Company) < 2 and
        #(m.feedbacks.writer & University) = 0)

}

//only student and company can write complaints
```

```
fact {
    always (all m: Match |
        #(m.complaints.writer & University) = 0)
}

//check that if the feedback is written by a company then the company
is the one int the match
fact {
    always (all m : Match | no f : Feedback|
        f in m.feedbacks and #(f.writer & Company) > 0 and f.writer
!= m.ref.hosted)
}

//check that if the complaint is written by a company then the company
is the one in the matche
fact {
    always (all m : Match | no c: Complaint|
        c in m.complaints and #(c.writer & Company) > 0 and c.writer
!= m.ref.hosted)
}

//check the students writes feedback only to his own matched matches
fact {
    always (all m : Match | no f : Feedback|
        f in m.feedbacks and #(f.writer & Student) > 0 and m.pub.student
!= f.writer)
}

//check the students writes complaints only to his own matched matches
fact {
    always (all m : Match | no c : Complaint|
        c in m.complaints and #(c.writer & Student) > 0 and m.pub.student
!= c.writer)
}

//no feedback of the same match can have the same writer

fact {
    always (all m : Match | no f1, f2 : Feedback|
        f1 in m.feedbacks and f2 in m.feedbacks and f1.writer = f2.writer
and f1 != f2)
}


//check that there is no match with the same publication and internship
fact {
```

```
    always (all m1 : Match | no m2 : Match |
       m1.ref = m2.ref and m1.pub = m2.pub and m1 != m2)
}

//only match with confirmed interview can have feedback

fact {
    always (all f: Feedback| some m: Match, i : Interview |
        f in m.feedbacks and i.match = m and i.confirmedYN = Yes )
}


//only match with confirmed interview can have complaints
fact {
    always (all c: Complaint| some m: Match, i : Interview |
        c in m.complaints and i.match = m and i.confirmedYN = Yes)
}


//every match can have at most one interview
fact {
    always (no i1, i2 : Interview|
        i1.match = i2.match and i1 != i2)
}

//there can be an interview only if the match is accepted

fact{
    always( no m: Match | some i : Interview |
        m.acceptedYN = No and i.match = m )
}



//the match once they are added they can't be deleted
fact {
    always (Match in Match')
}

//the match cant change his attributes
fact{
    always all m : Match|
       (m.acceptedYN = No and after(m.acceptedYN = No ))
      or
       (m.acceptedYN = Yes and after(m.acceptedYN = Yes ))
```

```
    always all m : Match| some p: Publication|
        m.pub = p and after(m.pub = p)

    always all m : Match| some i: Internship|
        m.ref = i and after(m.ref = i)
}


//the interview once they are added they can't be deleted
fact{
    always (Interview in Interview')
}

//the interview cant change his attributes
fact{
    always all i : Interview|
      always (i.confirmedYN = No implies historically(i.confirmedYN
= No ))
      or
      always (i.confirmedYN = Yes implies historically(i.confirmedYN
= Yes ))

    always all i : Interview  | some m: Match|
        i.match = m and after(i.match = m)

}

//the Feedback once they are added they can't be deleted
fact {
    always (Feedback in Feedback')
}

//the feedback can't change his attributes
fact{
    always all f : Feedback  | some u: User|
        f.writer = u and after(f.writer = u)
}

//the Complaint once they are added they can't be deleted
fact{
    always (Complaint in Complaint')
}

//the Complaint can't change his attributes
fact{
    always all c : Complaint  | some u: User|
```

```
        c.writer = u and after(c.writer = u)
}

pred show{
    no Match; #Match = 2
    no Interview; no Interview; some Interview
    no Complaint; no Complaint; no Complaint; some Complaint
    no Feedback; no Feedback; no Feedback; some Feedback
}


run show for 4
```

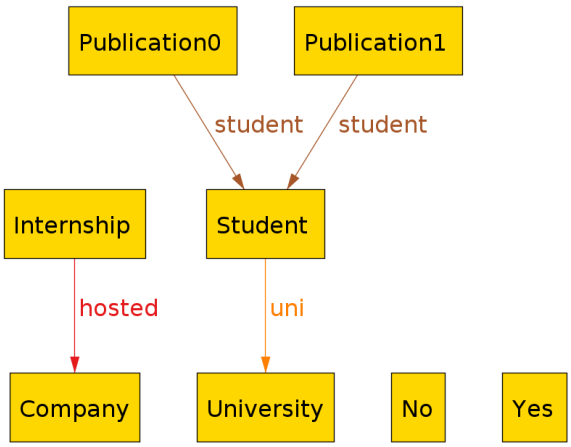

Figure 25: Alloy first state

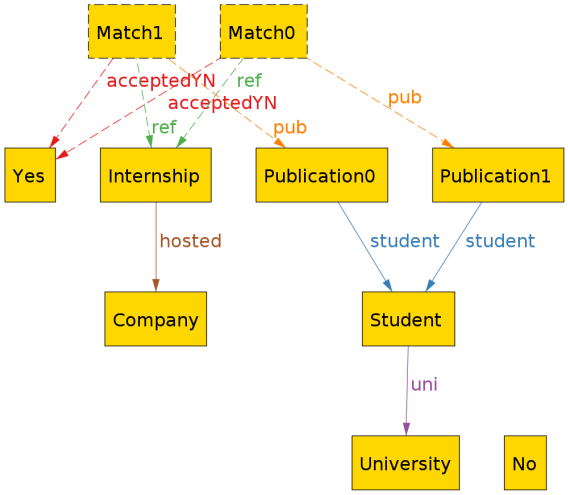

Figure 26: Alloy second state

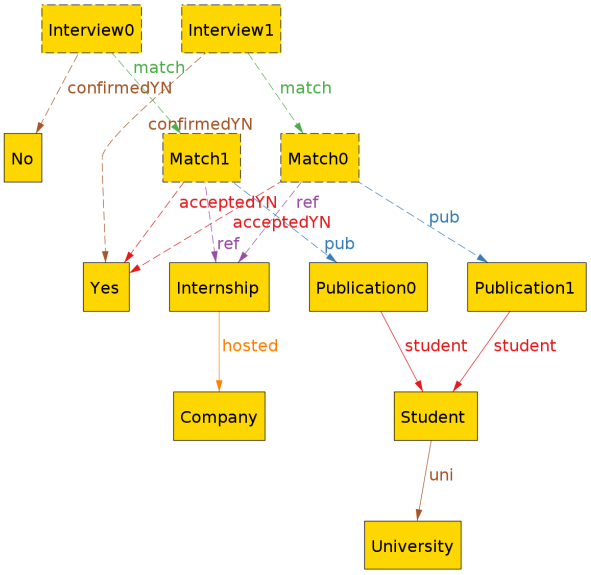

Figure 27: Alloy third state

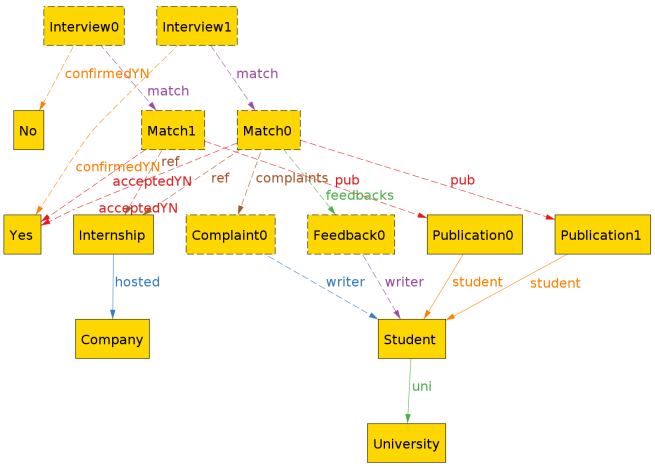

Figure 28: Alloy fourth state

The results of the model described with alloy are showed in the images 25, 26, 27 and 28; where it's possible to see the evolution of the model during the match between publications and internships, with the conducting of interviews and the creation of feedbacks and complaints.

5 Effort Spent

The table below show the number of hours that each member of the group worked for the RASD document.

Member	Hours
Elia Priuli	25h
Veronica Viceconti	26h
Marco Zuccoli	22h

6 References

1. Software Engineering II course slides
2. Draw.io
3. StarUML
4. AlloyAnalyzer