



**POLITECNICO**  
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

# Requirements Analysis and Specification Document

STUDENTS&COMPANIES

Author:

**Riccardo Bonfanti**  
**Jie Chen**

Student ID: 273115 276324

Advisor: Prof. Elisabetta Di Nitto

Academic Year: 2024–2025



# Deliverable Information

---

**Deliverable:** RASD  
**Title:** Requirement Analysis and Verification Document  
**Authors:** Riccardo Bonfanti, Jie Chen  
**Version:** 1.0  
**Date:** 13-December-2024  
**Download page:** <https://github.com/JieCver1/BonfantiChen>  
**Copyright:** Copyright © 2024, Riccardo Bonfanti, Jie Chen – All rights reserved

---

---

**Deliverable:** RASD  
**Title:** Requirement Analysis and Verification Document  
**Authors:** Riccardo Bonfanti, Jie Chen  
**Version:** 2.0  
**Date:** 22-December-2024  
**Download page:** <https://github.com/JieCver1/BonfantiChen>  
**Copyright:** Copyright © 2024, Riccardo Bonfanti, Jie Chen – All rights reserved

---

# Contents

<b>Deliverable Information</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
A Purposes and Goals . . . . .	1
A.1 Purpose . . . . .	1
A.2 Goals . . . . .	1
B Scope . . . . .	3
B.1 Scope . . . . .	3
B.2 Phenomena . . . . .	4
C Definitions, Acronyms, Abbreviations . . . . .	6
C.1 Definitions . . . . .	6
C.2 Acronyms . . . . .	7
C.3 Abbreviations . . . . .	7
D Revision History . . . . .	8
E Reference Documents . . . . .	8
F Document Structure . . . . .	8
<b>2 Overall Description</b>	<b>11</b>
A Product perspective . . . . .	11
A.1 Scenarios . . . . .	11
A.2 Domain-level diagram . . . . .	15
A.3 Statecharts . . . . .	16
B Product functions . . . . .	18
C User characteristics . . . . .	22
C.1 Students . . . . .	22
C.2 Companies . . . . .	22

C.3	Universities . . . . .	23
D	Assumptions,dependencies and constraints . . . . .	23
D.1	Domain Assumptions . . . . .	23
<b>3</b>	<b>Specific Requirements</b>	<b>25</b>
A	External Interface Requirements . . . . .	25
A.1	User Interfaces . . . . .	25
A.2	Hardware Interfaces . . . . .	28
A.3	Software Interfaces . . . . .	28
A.4	Communication Interfaces . . . . .	28
B	Functional Requirements . . . . .	29
B.1	Functional Requirements . . . . .	29
B.2	Requirement Mapping . . . . .	31
B.3	Use Case Diagram . . . . .	36
B.4	Traceability Matrix . . . . .	63
C	Performance Requirements . . . . .	65
D	Design Constraints . . . . .	66
D.1	Standards Compliance . . . . .	66
D.2	Hardware Limitations . . . . .	66
D.3	Any Other Constraint . . . . .	66
E	Software System Attributes . . . . .	66
E.1	Reliability . . . . .	67
E.2	Availability . . . . .	67
E.3	Security . . . . .	67
E.4	Scalability . . . . .	68
E.5	Maintainability . . . . .	68
E.6	Portability . . . . .	68
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>69</b>
A	Static part . . . . .	69
B	Dynamic part . . . . .	76
<b>5</b>	<b>Effort Spent</b>	<b>91</b>
<b>6</b>	<b>References</b>	<b>93</b>

# 1 | Introduction

## A. Purposes and Goals

### A.1. Purpose

Internships provide students with a valuable opportunity to apply their skills in real job environments while enabling companies to connect with fresh talent. However, the process of finding and securing internships can be challenging for both parties.

Students&Companies (S&C) is a platform designed to facilitate this connection throughout the internship process. It allows students to match their preferences with available opportunities, ensuring internships align with their experiences and skills. Companies can specify project requirements to attract suitable candidates.

The platform supports both students and companies in two phases: recommendation and selection. It utilizes keyword searches and statistical analyses to evaluate internship and student data, ensuring a match that meets the needs of both parties. Students can search for internships opportunities and the most appealing will be recommended to him, while companies can publish offers and be informed when student CVs match their criteria. Once mutual interest is established, the selection phase begins, where the platform assists with interviews and finalizes the process. It also monitors the internship journey, providing the possibility to exchange feedback and enabling direct communication to address any questions. Additionally, the universities of the students can oversee the process to ensure that everything is going smoothly.

### A.2. Goals

[G1] All unregistered Users must be able to create an account on the platform using their specific email address and log in to the platform.

[G2] Students must be able to upload their CVs on the platform.

[G3] Students must be able to search for available internship offers on the platform.

[G4] Companies must be able to publish internship offers on the platform.

[G5] Users must receive notifications about relevant events.

[G6] Students and Companies must receive recommendations based on statistical analyses.

[G7] Students must be able to proactively apply for internships, before the submission deadline.

[G8] Companies must be able to set up the interview process.

[G9] During the interview process, students must be able to respond to the company's questions through questionnaires.

[G10] Companies must be able to send the results of the interviews to the students.

[G11] Students must be able to accept or reject the internship offer after receiving the interview results.

[G12] Students and companies must be able to write feedback regarding their internship experiences.

[G13] Students and companies must be able to communicate with each other during internship process.

[G14] Universities must be able to monitor the internship process of their students.

## B. Scope

### B.1. Scope

Students&Companies (S&C) aims to provide the best matching service between students and companies for internships. The platform will be available to students, companies, and universities.

If it is their first time using the platform, Students looking for internships can create an account using their educational institution's email and select their university from a list of the ones that collaborate with the platform. After creating an account, they need to fill out their profiles to describe their skills, experiences, and preferences. To complete their profiles, they must also upload their CVs. Otherwise they can log in to the platform using their credentials.

The universities associated with the students through their educational emails will be notified about the students' registration on the platform. The system will then add the registered students to the university's list, allowing the universities to monitor their internship activities.

Companies wishing to announce internship opportunities can create an account on the platform, if they do not already have one. They need to provide the necessary information for the internship announcement such as the job description, requirements, and the number of interns needed etc. Companies can also specify the skills they are looking for in students.

The platform will extract key information from the students' profiles and the companies' internship announcements, along with historical feedback and information collected from previous internships, to recommend the best matches for both parties. Recommended internships will be displayed to the students, while students' profiles with CVs that meet the company's needs will be displayed to the company. Students will also receive notifications about recommended internships, and companies will be alerted when existing students with matching CVs meet their needs.

Once students apply for the offers they are interested in, the companies will receive the applications and can select the students they wish to interview. The platform will assist companies in setting up the interview process and will allow them to record and store students' responses using the platform. Students will be able to respond to companies' questions through the tools or channels provided.

At the end of the interviews, companies can send the results to the students. Upon

receiving the results, students can decide whether to accept or reject the offer, and the companies will be notified of their decisions. If a student accepts the offer and starts the internship, the platform will update the internship activities about that student at their universities. Students and companies can use the channels provided by the platform to communicate with each other during the internship, and also they can provide feedback and complains regarding their experiences in dedicated section which can be monitored by the students' universities.

## B.2. Phenomena

Referring to the Jackson-Zave distinction between the world and the machine in the context of the S&C platform, the following phenomena are identified, specifying which parts are controlled by the machine and which parts are controlled by the world, shown in table 1.1.

### Phenomena based on the Jackson-Zave model

Code	Phenomenon	Shared	Control
P1	User registration	Yes	World
P2	User login	Yes	World
P3	Check username and password	No	Machine
P4	Student creates CV using text editor	No	World
P5	Student uploads CV in profile	Yes	World
P6	Student updates profile information	Yes	World
P7	Student searches available internships	Yes	World
P8	Company publishes internship offers	Yes	World
P9	Platform notifies users that a deadline has expired	Yes	Machine
P10	Platform suggests recommendations	Yes	Machine
P11	Platform adds student to university's list	No	Machine
P12	Student submits an application for an internship	Yes	World
P13	Company selects candidates to interview	Yes	World
P14	Student participates to the interview	Yes	World
P15	Company sends interview results	Yes	World
P16	Student accepts or rejects the internship	Yes	World
P17	ST and CO communicate with each other	Yes	World
P18	ST and CO write feedback on the internship	Yes	World
P19	ST and CO view feedback on the internship	Yes	World
P20	User accesses and visualizes own or others' profiles details	Yes	World
P21	University tracks the internship processes of its students	Yes	World

Table 1.1: Phenomena in the S&C context

## C. Definitions, Acronyms, Abbreviations

### C.1. Definitions

- **User:** A generic term for students, companies, and universities who use the platform.
- **Student:** A person who is looking for internships.
- **Company:** An organization which wants to announce internship opportunities to students.
- **University:** An educational institution that is related to students and their internships.
- **Candidate:** A term for students whose applications are selected and that will take part in the interview process.
- **Internship:** A opportunity offered by companies to students to gain practical experience in a real job environment.
- **CV:** Curriculum Vitae, a document that contains all necessary information about students to be able to apply for internships.
- **Recommendation:** A suggestion made by the platform to students and companies based on statistical analyses and keyword searches.
- **Interview:** Start with a questionnaire form, that can be followed by an external meeting between students and companies, to evaluate the student preparation and to understand if the student is suitable for the internship.
- **Feedback:** Helpful information written by students and companies about their internship experiences to improve a performance of the two parties, including also negative aspects such as complains.
- **Comment:** The text that is written by students and companies to provide feedback or complaints about their internship experiences.
- **Complain:** A text that expresses dissatisfaction, issues, or annoyance about the internship experiences. It will be treated as a synonym of feedback in this document.
- **Notification:** A message sent by the platform to inform students and companies about important events, such as new internship offers, matching CVs, interview results etc.

- **Platform:** The Students&Companies (S&C) system that provides the services to students, companies, and universities about internships.
- **Keyword:** A significant word or tag used to describe content, such as the skills, experiences, and preferences of students and companies.

## C.2. Acronyms

- **S&C:** Students&Companies
- **CV:** Curriculum Vitae
- **UI:** User Interface
- **API:** Application Programming Interface
- **HTTPS:** Hypertext Transfer Protocol Secure
- **TLS:** Transport Layer Security
- **REST:** Representational State Transfer
- **JSON:** JavaScript Object Notation
- **SMTP:** Simple Mail Transfer Protocol
- **UTC:** Coordinated Universal Time
- **GDPR:** General Data Protection Regulation

## C.3. Abbreviations

- **[Gn]:** Used to number the goals, where Gn is the n-th goal.
- **[Pn]:** Used to number the phenomena, where Pn is the n-th phenomenon.
- **[Rn]:** Used to number the requirements, where Rn is the n-th requirement.
- **[An]:** Used to number the domain assumptions, where An is the n-th assumption.
- **CO:** Company
- **ST:** Student
- **UNI:** University

## D. Revision History

- **Version 1.0** – 13/12/2024
- **Version 2.0** – 22/12/2024
  - Changed phenomena [P20] description.
  - Added and modified some definitions and acronyms descriptions.
  - Made a small modification to the domain-level diagram and update its description accordingly.
  - Reformulated the requirements.
  - Fixed some inconsistencies in the descriptions of use case exceptions [U7] and [U14].
  - Provided a more specific and clear description of the Alloy models.

## E. Reference Documents

- “The World and the Machine: A model for the functional architecture” by Michael Jackson and Pamela Zave.
- Alloy Documentation - <https://alloy.readthedocs.io/en/latest/>
- Assignment RDD AY 2024–2025 specifications.

## F. Document Structure

This document is structured as follows:

- **Section 1: Introduction**

It contains the purpose, goals, scope, and phenomena identified in the context of the S&C project, including the specification of definitions, acronyms, and abbreviations of the terms used in this document. In addition, it notes the revision history for updates to the document and the reference documents that were used during the development of this document.

- **Section 2: Overall Description**

In this section, the general view of the S&C project is presented, including the product perspective, functions, characteristics, constraints, assumptions, and de-

pendencies. Through the use of UML diagrams, the statecharts, and the scenarios to explain the project's functionalities.

- **Section 3: Specific Requirements**

A detailed description of the requirements of the project will be provided in this section. Using the various point of views, the external interface requirements, functional requirements, performance requirements, design constraints, software system attributes, and other requirements. In particular to easier understanding, the use cases, sequence diagrams, mapping tables and simple drafts of the user interface will be presented.

- **Section 4: Formal Analysis**

Using the Alloy language to model the project and to verify the logical consistency of the model. It will also be used to check the correctness of the requirements and to present the main features guaranteed by the platform. Also, a part of the scenarios will be presented by predicates using show and run commands.

- **Section 5: Effort Spent**

The time spent by each group member on each task will be registered in this section. It will be used to present the effort dedicated by each member and to present the progress of the development of the project.

- **Section 6: References**

The other references that not include in the reference documents will be added in this section.



# 2 | Overall Description

## A. Product perspective

### A.1. Scenarios

#### **Scenario 1: Mr.Spongebob registers on the platform**

Mr.Spongebob, a final-year student at the University of Bikini Bottom, is looking for an internship to practice the knowledge he's gained. To do so, he asks advice from Professor Puff, who suggests using the Students&Companies platform to search for internship opportunities. Following her advice, Mr.Spongebob registers as a student on the platform, selecting the University of Bikini Bottom from the list of universities and verifying his student status with his educational email and password. Then, he fills in all the required personal information, including his name, date of birth, and other details. He also inserts keywords to describe his skills and the job fields he might be interested in. Finally, after uploading his CV, which contains all the necessary information and details, Mr.Spongebob completes his registration and can begin searching for internships. Meanwhile, Professor Puff, who oversees internship activities using the official account of the University of Bikini Bottom, is notified of Mr.Spongebob's registration.

#### **Scenario 2: Mr.Krabs publishes an internship offer**

Due to the rapid evolution of Technology, Mr.Krabs decides to collaborate with his competitor Mr.Plankton, on a new project. They plan to develop a "Hamburger super secret formula detector" that can detect what the customer likes by looking at his facial expressions, in order to create the perfect personalized hamburger! To achieve this, they need to hire students specializing in Computer Vision, Deep Learning and AI to study the visual data. So Mr.Krabs logs into the Students&Companies platform using the official company account and navigates to the "Publish Internship" section to create a new announcement. He fills in all the required information: the title, location, role, application deadline, number of positions, duration, employment type, description, and required skills.

### "Computer Vision and AI Intern"

*Location:* Krusty Krab Technology Lab, Bikini Bottom

*Role:* Intern

*Application Deadline:* 25th December 2024

*Number of Positions:* 2

*Duration:* 6 months

*Employment Type:* Full-time

*Description:* We are looking for students who can contribute to the development of the Hamburger super secret formula detector by developing algorithms that collect and analyze visual data. A love for hamburgers is essential!

*Required Skills:* Computer Vision, AI, Python, Deep Learning

After that, Mr.Krabs publishes the internship offer, which can be seen in the "Available Internships" section and in the company's profile section. Immediately after the publication, he receives a notification of recommended student profiles that may match the offer requirements.

### Scenario 3: Mr.Patrick searches for internships

Mr.Patrick is very worried because he is in the last semester of his master's degree, so he has to find an internship or he won't be able to graduate in Food Engineering. He remembers he has an account on Students&Companies, which he used to find an internship at the beginning of his studies, so he decides to take another look at the platform. He logs in and updates his profile and CV, adding the new skills he has acquired during his master's degree. He navigates to the search bar and types "Hamburger" to look for internships related to hamburger studies. When he doesn't find anything that interests him, he scrolls through the recommended internships list. Finally, he finds Mr.Krabs' announcement and becomes very interested in the project, so he applies for the internship and waits for the company's response. After applying, he can check the status of his application in the "My Applications" section.

### Scenario 4: Krusty Krab Technology Lab Interview

After the application deadline, Mr.Krabs and Mr.Plankton review all the applications received for the intern position. They select the candidates by reviewing their profiles and CVs, and then send them an invitation for an interview. For the first round of interviews, Mr.Plankton decides to formulate a set of short questions to know more about the can-

dicates' skills and experiences. The candidate will be able to respond to these questions through a form inside the platform. In said form Mr.Plankton can also include the date, time and location for a live interview and the necessary information to better prepare the candidates. At the end of the interviews, Mr.Plankton evaluates the candidates and discusses with Mr.Krabs to decide who to hire. Finally, they send an offer to the best candidates and wait for their acceptance to start the internship.

### Scenario 5: Mr.Gary's application is rejected

Mr.Gary, an undergraduate student at the University of Bikini Bottom, applied for an internship at the "Krusty Krab Technology Lab" since he was very interested in the opportunity. After the application deadline, Mr.Krabs and Mr.Plankton review all the applications received for the intern position. Unfortunately, they do not think Mr.Gary has enough experience for the internship, so they reject his application. Mr.Gary receives a notification informing him that his application was not selected and explaining the reason for the decision. He will still be able to see his application in the "My Applications" section and the status will be marked as "rejected".

### Scenario 6: Mr.Squidward goes through the interview phase

Mr.Squidward, a student at the University of Bikini Bottom, was selected for an interview after he applied to the "Krusty Krab Technology Lab" as a data science intern. He goes to the "My Applications" section and fills the interview form with his answers. Then he checks the details for the live interview and prepares for it. After the interview, he waits for the company's final decision. Finally, if he is selected he will receive a notification with the offer from the company, he can still choose to turn down the offer if he changes his mind or finds a better opportunity, otherwise he can accept and start the internship.

### Scenario 7: Ms.Sandy starts her internship

Ms.Sandy is very passionate about space and is thrilled to have been selected for an internship at the Underwater Space Agency. She receives and accepts the offer from the company. At the same time, the Bikini Bottom University of Aerospace's professor Manta receives the notification about the new activity started by Sandy. However, Sandy is a bit anxious, so she logs into the platform and navigates to the "My Internships" section to review its details and opens the chat with the company's representative. She then asks some questions about how she should prepare in the coming days and for any advice that they think should be useful to know to have a good start in the internship.

## Scenario 8: Mr.Squidward leaves his feedback

At the end of his internship at the “Krusty Krab Technology Lab” as a data science intern, Mr.Squidward is surprised by the experience he has gained, even though he has several complaints about the heavy workload. He decides to leave a warm message for Mr.Krabs and Mr.Plankton to thank them for the opportunity. After that, he navigates to the “Internship” section to write feedback on his internship experience. It is also the section where he can check the suggestions left by his mentors regarding his internship work, which is only visible to the company and the university.

## Scenario 9: Track the internship activities

As the internship period progresses, Professor Puff is responsible for overseeing the students’ activities at the University of Bikini Bottom. She is curious to know how the students are doing in their internships, particularly Mr.SpongeBob, who is using the platform for the first time. She logs into the platform with the university credentials and looks at the list of students, searching for SpongeBob’s name. She finds, in his active internship section, that he has been selected for an internship at the “Krusty Krab Technology Lab”. By reading the comments left by the company throughout the different stages of the internship, she is very glad to see that SpongeBob is doing well in the field he is passionate about!

## A.2. Domain-level diagram

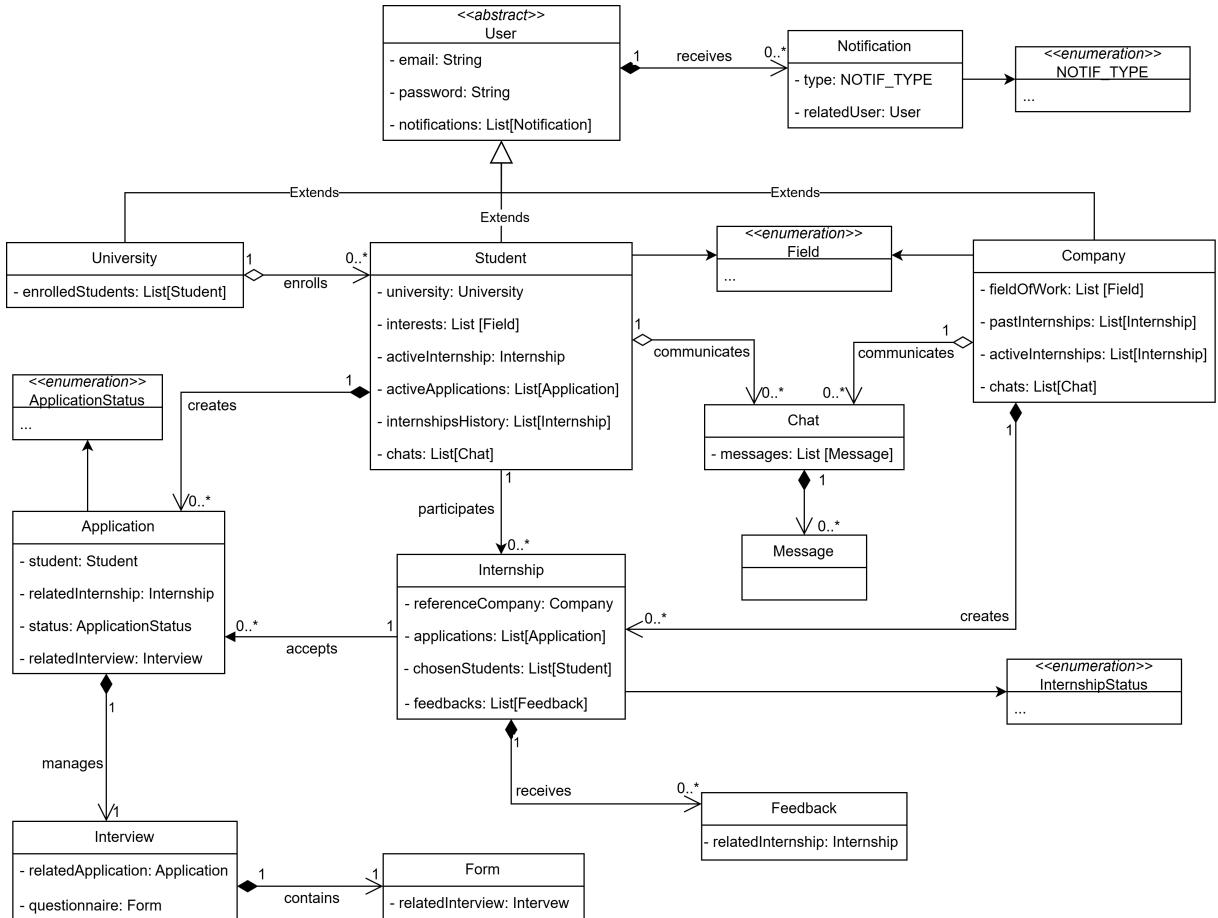


Figure 2.1: Domain-level diagram

Students, Companies and Universities are implemented by extending the abstract class User. The User class contains the common attributes: email, password and the list of notifications. Notifications have a type, description and date and are related to a specific User. Universities have a name and contain the list of all their enrolled students. The Student class contains specific attributes to save their profile information (like name, interests, university, CV,...) and can create applications for internships that are stored in a list. They also have references to ongoing chats and ongoing and past internships. Companies contain specific attributes to save their profile information (like legalName, EIN, field of work,...) and can create internships that are stored in a list. They also have references to ongoing and past internships and current chats.

The Chat is a collection of messages between a Student and a Company.

The Application class contains the information about the student that applied for an

internship and also records the status of the application. In the Application we also have a reference to an Interview object.

The Interview class contains the information about the interview session: the date and link to the live session, and the Form, an object that contains both the questions and the student's answers.

The Internship class contains all the information about it: the title, location, description, deadline. It also has references to the Company who made the proposal, and maintains a list of all the submitted applications, a list of the students that were chosen for the internship, and a list of the feedbacks left both by the students and the company at the end of it.

The Feedback class contains a reference to the related Internship and a description.

The ApplicationStatus enumeration is used to keep track of the status of the application: Waiting, Selected, AcceptedToInternship, AcceptedOffer, RejectedOffer, Rejected. The InternshipStatus enumeration is used to keep track of the status of the internship: InPublishing, InSelection, InProgress, Completed.

### A.3. Statecharts

The following sections present statechart diagrams for the platform's main processes, including internship recommendations, the internship selection process, and internship status tracking.

#### 1. Internship recommendation

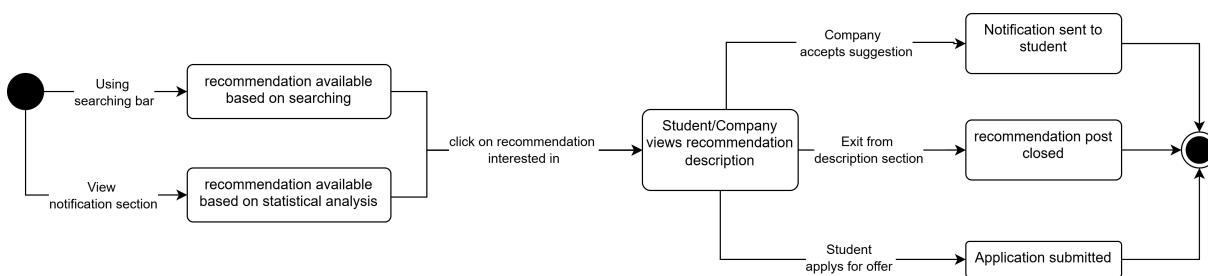


Figure 2.2: Statechart diagram for internship recommendation

The recommendation process is a key functionality of the platform. After logging in, both students and companies can use the search system to view recommendations that match their preferences. Alternatively, they can access suggestions in the notification section,

which are generated through a statistical analysis of data from profiles and internship details. As illustrated in the statechart diagram, when users find an interesting recommendation, they can click on it to view more details. At this stage, students can apply for the internship, while companies can decide whether to accept the student profile. If accepted, the student will be notified that a company is interested in their profile for a specific internship. Otherwise the student or the company can close the recommendation after reviewing it.

## 2. Selection process for internship

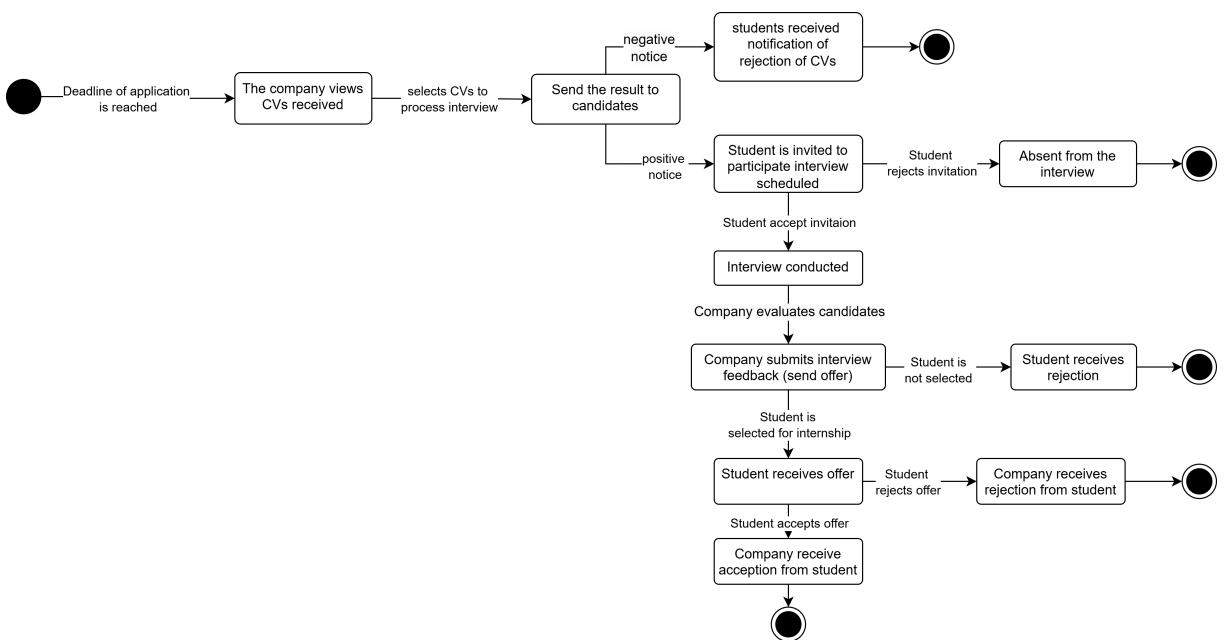


Figure 2.3: Statechart diagram for selection process for internship

Through the selection process, that will begin after the application deadline, companies will review all applications received for the intern position and select candidates for the interview stage. Once a decision is made, the company will send an interview invitation to the selected candidates through the platform, while those not selected will receive a notification explaining that, unfortunately, they were not chosen to proceed further. If a candidate does not attend the interview, the company will consider this as a rejection of the invitation. After all interviews are completed, the company will evaluate the candidates and decide whom to hire. The selected candidates will receive an offer, and the company will await their acceptance to confirm the internship role. At the same time, candidates not selected at this stage will receive a rejection notification. Once candidates respond to the offer, the company will be notified of their decision.

### 3. Internship status

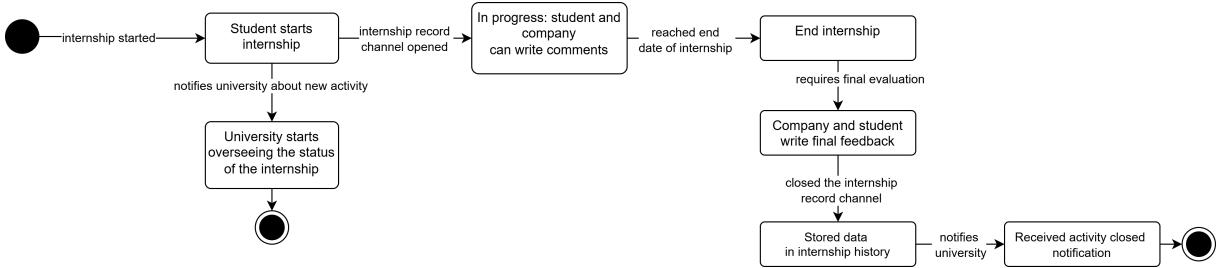


Figure 2.4: Statechart diagram for internship status

This statechart diagram illustrates the internship status tracking process. Once the internship begins, the student's university will be notified, allowing the university to monitor the student's activities and ensure the internship is progressing as expected. During the internship, the student can use a chat feature to communicate with the company representative and can provide feedback on their internship experience in the specific internship section. Similarly, the company can use the chat to communicate and leave feedback on the student's performance, offering suggestions for improvement. At the end of the internship, both the student and the company must complete the final evaluation and feedback process, marking the end of the internship phase. The internship will then be officially closed, and all related information will be archived in the database. At this stage, the university will be notified of the internship's completion. The student will find this internship listed as a past activity in their internship history, and the company will have access to this record in their records for future reference.

## B. Product functions

### Sign up and login

There are three types of users on the platform: students, companies, and universities. The platform allows users to create an account if they do not already have one. In the first step, each user will be prompted to specify whether they are a student, company, or university. All users will need to set up login credentials with an email address and password. For students, the system will ask them to select their university from a list. If their university is not listed, they will be unable to continue with registration. Students must also verify their status by using their educational email. For companies, multiple accounts can be created under the same organization, with each linked to a different department's unique email address. The company's identity shall be also verified using its unique EIN number and other re-

quired details. Once verified, users can access the platform with their email and password. On first login, the system will require users to complete their profile with essential information to finalize registration. Students, in particular, must upload a CV before accessing other features of the platform. After registration, users can log in simply by selecting their user type and entering their credentials.

### **Data management system**

The platform securely stores all user data in a protected database, including personal information, CVs, chat history, and other relevant data. Specifically, the system retains detailed internship information, such as announcements, applications, interview records, internship status, and feedback, which are used to analyze and optimize the matching algorithm. Additionally, other user data is recorded to provide better personalized recommendations and notifications. The system ensures that all data is encrypted and protected from unauthorized access. Furthermore, users have the option to modify personal information,in particular for students update their CVs as needed.

### **Search system**

This function allows students to search for internships using keywords such as job title, company name, location, and more. Similarly, companies can search for student profiles using criteria like name, university, skills, and other relevant details. Universities, on the other hand, can search a specific student within their student list by name to monitor activities and access the student's profile. The search system will display a list of recommendations based on the user's preferences and search history. Additionally, users can filter the search results by selecting specific categories to refine their search.

### **Recommendation system**

This feature is one of the platform's main functionalities, offering personalized recommendations to students and companies based on their search preferences and the statistical analysis of data from profiles and internship details. Recommendations can be viewed in the search results, in dashboard page or in the notification section. The system is designed to continuously improve the accuracy of these recommendations by analyzing user behavior and feedback in real-time, optimizing the quality of the results. Users can click on a recommendation to view more details about the internship posting or student profile.

### **Internship application**

Through this function, students can apply for internships that interest them after

viewing the details of the posting. Once the application deadline has passed, the company will review all received applications and select candidates for the interview stage. The platform will notify students of the results at this phase. If a student's application is accepted, they will be invited for an interview. Otherwise, the student will be notified that their application was rejected. However, the student can still apply for other internships, but cannot reapply for the same internship once rejected.

### **Internship Posting**

Companies can publish internship announcements on the platform using this function. In the "Publish Internship" section, companies must provide all required details, including the title, location, role, application deadline, number of positions, duration, employment type, description, and required skills. Once the internship offer is published, the company will receive recommendations for student profiles that match the it's criteria. Additionally, the company can access the "Available Internships" section and select a specific posting to view the list of applications received and select candidates for the interview stage.

### **Selection phase Assistance**

Once the application deadline has passed, the company can use this function to select candidates and send interview notifications. The system will guide the company through the selection process by providing a list of received applications, allowing the company to review each candidate's profile and CVs. Additionally, the system will offer tools such as questionnaire forms to help the company know more about the candidates before the actual interview will take place. The system will also notify students of the results from the selection phase (application or interview result).

### **Internship Status Tracking**

This function allows users to monitor the progress of internships, including the past completed internships and the current active internships. Students can view internship status through their internship history, while companies can access it through their records of the history of their published internships. Universities can monitor student activities through the student list associated with them. The status tracking system is designed to record all important information about the internship and notify the relevant parties of any changes or updates.

### **Internship Feedback and complaints system**

This feature is activated when the internship begins. Each specific internship will have a dedicated area for both the student and the company to provide feedback and complains. It allows students and companies to share useful information, such

as suggestions, key issues, and ongoing feedback, separate from the chat system. This section is designed to record important highlights throughout the internship process. At the end of the internship, the system will prompt both the student and the company to submit their final evaluations or comments. Once the internship is concluded, this feature will be deactivated, and all related information will be archived in the database. The university can access this data in order to monitor and evaluate the student's performance.

### Chat system

The chat system is an important feature of the platform that allows users to communicate with each other. Unlike a typical chat system, this function is not always active. It is activated once a company selects a student to be part of one of their internships. It allows both parties to communicate during the internship period and maintain contact afterward for potential future collaboration, such as mentorship or networking. Through the chat system, they can exchange messages and files, while the chat history will be stored in the database for future reference. The activation and deactivation of the chat system ensures privacy and security, preventing unnecessary communication, such as spam or harassment, before real collaboration begins and filtering out unqualified messages. The key difference between the chat system and the feedback system is that the chat system is real-time and private, accessible only to the two parties involved. In contrast, the information collected through the feedback system is more formal and public.

### Notification system

This function creates a channel for the platform to send notifications to users. Notification types include recommendations, application results, interview invitations, interview results, internship status updates, and other important communications from the platform. For example, when a student starts an internship, the university will be notified through the notification system.

### Acceptance and Rejection

This function allows users to express their decisions: students can accept or reject internship offers, and companies can accept or reject student applications. Once a student accepts an offer, the company is notified, and the internship officially begins. If a student rejects an offer, the company is notified, and the reserved position is reopened for other candidates. Similarly, if a company rejects a student's application, the student is notified, and the application is marked as rejected. The acceptance and rejection system ensures that both parties are informed of the results and can proceed accordingly. Additionally, this system helps improve the platform's

matching algorithm by providing better recommendations in the future, based on the acceptance and rejection rates to better understand user preferences.

## C. User characteristics

This section will examine the needs of the platform's primary users: students, companies, and universities. Each user group has specific requirements, and the platform is designed to meet these needs with various features. Here's an overview of what each group expects from the platform:

### C.1. Students

Students are the primary users of the platform who are looking for suitable internship opportunities to gain practical experience. They need an intuitive and user-friendly interface that makes it easy to search for internships. Since many students are not familiar with the job market, they require a recommendation system that suggests internships match their skills, preferences, and career goals. They also expect through the platform they will be able to apply for internships that they are interested in and to update their profile and CV when they gain new skills or experiences. Furthermore, students want to communicate directly with companies once a collaboration is established. They would like to receive feedback from companies about their performance during the internship, as this helps them improve their skills and future careers. Additionally, they want to share their experiences and leave comments to help other students evaluate potential employers before applying. Finally, students need to be able to track the status of their internships and receive timely notifications, allowing them to take informed action.

### C.2. Companies

Companies are the users of the platform that provide internship opportunities to students. They require a platform that allows them to easily publish and manage internship postings, receive applications, and select candidates for interviews. Specifically, companies expect the platform to offer recommendations that match the requirements for the internships they've posted. This helps them identify the most suitable candidates and streamline the selection process. Companies also need an efficient communication system to interact with students for scheduling interviews and discussing internship details. They expect to receive feedback from students on how to improve the company's internship programs and provide evaluations on student performance. Additionally, companies

need access to student profiles and CVs to identify the best candidates for the internships. They require the ability to track the status of internships and receive notifications regarding students' decisions on whether to accept or decline an offer. The platform should allow companies to have multiple accounts, especially if they represent different departments, and manage them separately. Furthermore, companies should be able to select student profiles that interest them and contact students through the notification system to promote their internship announcements.

### C.3. Universities

As users who are closely related to the students, universities need to be able to monitor the activities of students who are registered as their students. They need to be able to track students' internship statuses and receive notifications about updates to students' activities. Additionally, they need to be able to view students' profiles and activity histories to evaluate their performance, using the search bar to find a specific student.

## D. Assumptions, dependencies and constraints

### D.1. Domain Assumptions

[A1] The university of the student who want to register on the platform must be listed in the platform's database. If the university is not listed, the student will not be able to complete the registration process.

[A2] All users who want to use the platform must have an email address and necessary information to verify their identity.

[A3] All students should have a CV that contains all the necessary information and details about their experience.

[A4] Every department of a company has a unique email address that identifies it.

[A5] The platform will be able to communicate correctly with email servers to verify users' identity.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[A8] The companies do not publish the same internship announcement more than once before the application deadline.

[A9] A student does not apply for the internship if she/he has already accepted an offer which starts in the same period.

[A10] Companies who have an account on the platform will eventually publish internships.

[A11] Students who have an account on the platform will eventually apply for internships.

# 3 | Specific Requirements

## A. External Interface Requirements

### A.1. User Interfaces

As the users of the platform include students, companies, and universities, the user interface is designed to facilitate interaction with the platform and meet their needs. For instance, many students may be unfamiliar with the internship search process and may lack experience in finding job opportunities, so the user interface should be simple and intuitive to use.

In the following section, we describe the drafts of the principal Graphical User Interfaces (GUIs). These will be further detailed using refined mockups in the Design Document.

- **Welcome page:** The welcome page is the first page users see when they open the platform. It contains the platform's logo and a brief description of its purpose. On this page, users can log in using their credentials or create an account if they are new to Students&Companies.
- **Registration Page:** In the registration page, users will first be asked to select their user type (student, company, or university). Based on the selected user type, the corresponding registration form will be displayed, and users will be prompted to provide the necessary information to create an account. In addition to common data such as name, email, and password, users will need to provide additional information based on their user type. For example, students will be asked to personalize their profiles by adding details such as their interests, skills, CV, university and a short description of themselves, while companies will be asked their EIN number, etc.
- **Notification Page:** It will display various types of notifications received from the platform. For example, students will see notifications about updates on their application status, companies will receive notifications about the student's decision, and universities will be notified about new student registrations on the platform.

- **Side Menu:** The side menu is a navigation menu that provides access to the main sections of the platform and depends on the user's type. It will be displayed on the left side of the screen, with different sections available based on the user type (further explanation below).
- **Header Bar:** The header bar is displayed at the top of the screen and includes the platform's logo, a Side Menu icon, a chat icon, a notification icon, and the user's profile icon. It remains visible on all pages once the user has logged in.
- **Chat Interface:** The chat interface is accessible from the header bar. It allows students and companies to communicate with each other in real-time. They can view a list of available chats, send and receive messages, and access their chat history.
- **Profile Page:** On this page, users can view other users' public information. If they own the profile, they can edit their data as needed (by filling the fields in the "Edit Profile" page). Specifically, students can update their CV by accessing this page.
- **Search page:** This page contains a search bar and displays the results based on the user's search. For example, a student can search for internships, and the platform will return a list of results based on the student's input.
- **Internship evaluation page:** This page is used by both students and companies to evaluate internship performance. Both parties can provide feedback or complaints during the internship. At the end, to complete the evaluation process, they must leave final comments and rate the internship.
- **Internship details page:** This page displays detailed information about a specific internship. For example, all users can view the internship's description, requirements, and application deadline.

Since the platform is designed for three types of users, the specialized interfaces for each of them are described in further detail below, considering their characteristics.

- **Student Interfaces:**
  - **Student Dashboard:** This is the home page that appears once the student logs in. On this page, the student gets an overview of all the main functionalities they can access. The dashboard consists of:
    1. Search Bar with Filter Command: Allows to perform keyword searching.
    2. Overview of Recommended Internships: Displays job search keywords and internships recommended by the platform's algorithm.

3. My Applications: Provides an overview of the student's applications' status.
  4. My Internships: Displays a record of the internships the student has applied for in the past, including the internship they are currently doing if exist.
- **Student Side Menu:** For students, the side menu will include sections such as Search Internships, Dashboard, My Applications and My Internships.
  - **My applications:** The student can track the status of their applications and view the details of the internships they have applied for. Through this interface, the student can also receive information related to the selection process, such as the interview date, the interview result, and the final decision of the company. Additionally, the student can filter their applications based on their status, such as “waiting”.
  - **My Internships:** The student can view the internships they have participated in so far, along with the evaluations related to their experiences. If the student is currently participating to an internship, it will be displayed at the top. The page also allows the student to access the Internship Evaluation page, where they can leave comments or notes about a specific internship.
- **Company Interfaces:**
    - **Company Dashboard:** It's the home page that appears once the company logs in. On this page, the company gets an overview of all the main functionalities they can access. The dashboard consists of:
      1. Search Bar with Filter Command: Allows to perform keyword searching.
      2. Overview of recommended Student Profiles: Displays student profiles recommended by the platform's matching algorithm for the internship announcements that have been published by the company.
      3. Internship Announcements: A list of internship announcements the company has published, along with the number of the applications received.
    - **Company Side Menu:** The side menu of companies includes sections such as Search Profile, Dashboard, Publish Internship, and Internships Management.
    - **Publish Internship Page:** This is the main functionality of the company interface. On this page, the company can create a new internship announcement and post it on the platform. The company is required to enter the necessary information, following the guidelines provided by the platform.

- **Internship management page:** This page is divided into several sections: one for internship announcements that have been published but are still in the publication phase, one for internships in the selection phase, one for internships currently in progress, and one for closed internships. From this interface, the company can manage internships of all phases.
- **University Interfaces:**
  - **University Dashboard:** The university dashboard displays a list of students and their associated activities. The university can view the list of students registered on the platform and select a student to see detailed information about their profile, activities, and internships they are currently doing or have completed in the past.
  - **University Side Menu:** For universities, the side menu will include sections such as Search Profile and Dashboard.

## A.2. Hardware Interfaces

Students&Companies is a web-based platform, so it can be accessed from any device with an internet connection. The platform is compatible with all modern web browsers. It is designed to be responsive and work on different screen sizes, including desktops, laptops, tablets, and smartphones.

The system will be hosted on multiple servers that meet the requirements for web hosting. They will be responsible for the platform's backend processes, such as the recommendation algorithm, the data collection and storing and the statistical analyses.

## A.3. Software Interfaces

S&C will interact with an emailing system to confirm the registration of new users. It will also integrate various APIs to provide additional functionalities, for instance an API for implementing statistical analyses based on collected data that will be used to improve the recommendation algorithm. Another example would be an API to interact with the database in order to store and retrieve information.

## A.4. Communication Interfaces

The system will use HTTPS to ensure secure communication between the client and the server. The platform will also use WebSockets to enable real-time communication between users, that is required by functionalities such as the chat. Since the platform is designed

to be a RESTful web application, it will use JSON as the data exchange format between the client and the server.

To interact with the mailing system, the platform will use SMTP protocol.

## B. Functional Requirements

The functional requirements that the platform must meet to achieve the user's needs are described below.

### B.1. Functional Requirements

- [R1] S&C allows unregistered Users to sign up
- [R2] S&C allows registered Users to login
- [R3] S&C allows STs to upload their CV in their profile section
- [R4] S&C allows STs to search for internships
- [R5] S&C allows COs to create internships by compiling all the information
- [R6] S&C allows COs to set a deadline for submitting the application to an internship
- [R7] S&C allows COs to publish internships
- [R8] S&C should notify STs when they are selected by the CO for the interview process
- [R9] S&C should notify STs when a CO is interested in their profile
- [R10] S&C should notify STs when he is successfully selected for a position
- [R11] S&C should notify STs when a CO rejects their application
- [R12] S&C should notify UNIs when their students start an internship
- [R13] S&C should notify STs when an internship available matches their interest
- [R14] S&C should notify COs when the deadline for a published internship has expired
- [R15] S&C should notify COs when the candidate accepts the position
- [R16] S&C should notify COs when the candidate refuses the position
- [R17] S&C should notify COs and STs when a new chat message is available
- [R18] S&C should notify COs when a ST with a CV that corresponds to their needs is available

[R19] S&C should be able to analyze the User's data to provide the recommendations to both STs and COs

[R20] S&C allows STs to submit their application for an internship

[R21] S&C allows STs to visualize information about a published internship

[R22] S&C allows COs to view the list of all applications that were submitted for a specific internship

[R23] S&C allows COs to record STs selection outcomes

[R24] S&C allows COs to create forms to submit to candidates for the interview process

[R25] S&C allows candidates to respond to the received forms

[R26] S&C allows COs to visualize the responses of the candidates who have replied to the forms

[R27] S&C allows COs to record the results of the interview

[R28] S&C allows STs to check the status of their applications

[R29] S&C allows STs to accept or reject the offer after receiving the interview results

[R30] S&C allows STs and COs to write feedback and complaints relating to the internship experience

[R31] S&C allows Users to view feedback and complaints relating to the internship experience

[R32] S&C allows STs and COs to exchange information using the chat, only if the ST is participating or has participated in an internship offered by the company

[R33] S&C allows UNIs to check the status of the internship records of their students

[R34] S&C allows UNIs to access the list of all the enrolled STs that are registered on the platform

[R35] S&C allows Users to visualize the own and other users' profiles

[R36] S&C allows Users to modify their own profile data

[R37] S&C should notify UNIs when their students register on the platform

## B.2. Requirement Mapping

In the following table, represent the mapping between the goals, requirements and domain assumptions. The requirements are mapped to the goals they contribute to, and the domain assumptions they depend on.

Goal	Requirement	Domain Assumption
G1	R1, R2	A1, A2, A5, A6, A7
G2	R3, R36	A3, A5, A6, A7
G3	R4, R21	A6, A7, A10
G4	R5, R6, R7	A4, A6, A7, A8, A10
G5	R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R37	A6, A7
G6	R13, R18, R19	A3, A6, A7
G7	R20, R21	A3, A6, A7, A9, A10, A11
G8	R22, R23, R24	A4, A6, A7, A11
G9	R25	A6, A7
G10	R26, R27	A4, A6, A7
G11	R28, R29	A6, A7
G12	R30, R31, R37	A6, A7, A10, A11
G13	R32	A6, A7, A10, A11
G14	R33, R34, R35	A6, A7

[G1] All unregistered users must be able to create an account on the platform using their specific email address and log in to the platform.

[R1] S&C allows unregistered user to sign up

[R2] S&C allows registered Users to login

[A1] The university of the student who want to register on the platform must be listed in the platform's database. If the university is not listed, the student will not be able to complete the registration process.

[A2] All users who want to use the platform must have an email address and necessary information to verify their identity.

[A5] The platform will be able to communicate correctly with email servers to verify users' identity.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G2] Students must be able to upload their CVs on the platform.

[R3] S&C allows STs to upload their CV in their profile section

[R36] S&C allows Users to modify their own profile data

[A3] All students should have a CV that contains all the necessary information and details about their experience.

[A5] The platform will be able to communicate correctly with email servers to verify users' identity.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G3] Students must be able to search for available internship offers on the platform.

[R4] S&C allows STs to search for internships

[R21] S&C allows STs to visualize information about a published internship.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[A10] Companies who have an account on the platform will eventually publish internships.

[G4] Companies must be able to publish internship offers on the platform.

[R5] S&C allows COs to create internships by compiling all the information.

[R6] S&C allows COs to set a deadline for submitting the application to an internship.

[R7] S&C allows COs to publish internships.

[A4] Every department of a company has a unique email address that identify it.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[A8] The companies do not publish the same internship announcement more than once before the application deadline.

[A10] Companies who have an account on the platform will eventually publish internships.

[G5] Users must receive notifications about relevant events.

[R8] S&C should notify STs when they are selected by the CO for the interview process.

[R9] S&C should notify STs when a CO is interested in their profile.

[R10] S&C should notify STs when he is successfully selected for a position.

[R11] S&C should notify STs when a CO rejects their application.

[R12] S&C should notify UNIs when their students start an internship.

[R13] S&C should notify STs when an internship who matches their interest is available.

[R14] S&C should notify COs when the deadline for a published internship has expired.

[R15] S&C should notify COs when the candidate accepts the position.

[R16] S&C should notify COs when the candidate refuses the position.

[R17] S&C should notify COs when a new chat message is available.

[R18] S&C should notify COs when a ST with a CV that corresponds to their needs is available.

[R37] S&C should notify UNIs when their students register on the platform.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G6] Students and Companies must receive recommendations based on statistical analyses.

[R13] S&C should notify STs when an internship who matches their interest is available.

[R18] S&C should notify COs when a ST with a CV that corresponds to their needs is available.

[R19] S&C should be able to analyze the User's data to provide the recommendations to both STs and COs.

[A3] All students should have a CV that contains all the necessary information and details about their experience.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G7] Students must be able to proactively apply for internships, before the submission deadline.

[R20] S&C allows STs to submit their application for an internship.

[R21] S&C allows STs to visualize information about a published internship.

[A3] All students should have a CV that contains all the necessary information and details about their experience.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[A9] The student do not apply for the internship if she/he has already accepted offer which starts in the same period.

[A10] Companies who have an account on the platform will eventually publish internships.

[A11] Students who have an account on the platform will eventually apply for internships.

[G8] Companies must be able to set up the interview process.

[R22] S&C allows COs to view the list of all applications that were submitted for a specific internship.

[R23] S&C allows COs to record STs selection outcomes.

[R24] S&C allows COs to create forms to submit to candidates for the interview process.

[A4] Every department of a company has a unique email address that identify it.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[A11] Students who have an account on the platform will eventually apply for internships.

[G9] During the interview process, students must be able to respond to the company's questions through questionnaires.

[R25] S&C allows candidates to respond to the received forms.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G10] Companies must be able to send the results of the interviews to the students.

[R26] S&C allows COs to visualize the responses of the candidates who have replied to the forms.

[R27] S&C allows COs to record the results of the interview.

[A4] Every department of a company has a unique email address that identify it.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G11] Students must be able to accept or reject the offer after receiving the interview results.

[R28] S&C allows STs to check the status of their applications.

[R29] S&C allows STs to accept or reject the offer after receiving the interview results.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[G12] Students and companies must be able to write feedback regarding their internship experiences.

[R30] S&C allows STs and COs to write feedback and complaints relating to the internship experience.

[R31] S&C allows Users to view feedback and complaints relating to the internship experience.

[R37] S&C should notify UNIs when their students register on the platform.

[A6] Users have a stable internet connection to access the platform.

[A7] The users of the platform must have a device that supports the platform's features.

[A10] Companies who have an account on the platform will eventually publish internships.

[A11] Students who have an account on the platform will eventually apply for internships.

**[G13]** Students and companies must be able to communicate with each other during internship process.

**[R32]** S&C allows STs and COs to exchange information using the chat, only if the ST is participating or has participated in an internship offered by the company.

**[A6]** Users have a stable internet connection to access the platform.

**[A7]** The users of the platform must have a device that supports the platform's features.

**[A10]** Companies who have an account on the platform will eventually publish internships.

**[A11]** Students who have an account on the platform will eventually apply for internships.

**[G14]** Universities must be able to monitor the internship process of their students.

**[R33]** S&C allows UNIs to check the status of the internship records of their students.

**[R34]** S&C allows UNIs to access the list of all the enrolled STs that are registered on the platform.

**[R35]** S&C allows Users to visualize the own and other users' profiles.

**[A6]** Users have a stable internet connection to access the platform.

**[A7]** The users of the platform must have a device that supports the platform's features.

### B.3. Use Case Diagram

In following section, the use case diagrams, the use cases description and the sequence diagrams are presented to describe the interaction between the users and the platform in a more detailed way.

#### Use Case Diagrams

The use case is divided into three main actors: Students, Companies and Universities. Each actor has a set of use cases that describe the functionalities he can access on S&C.

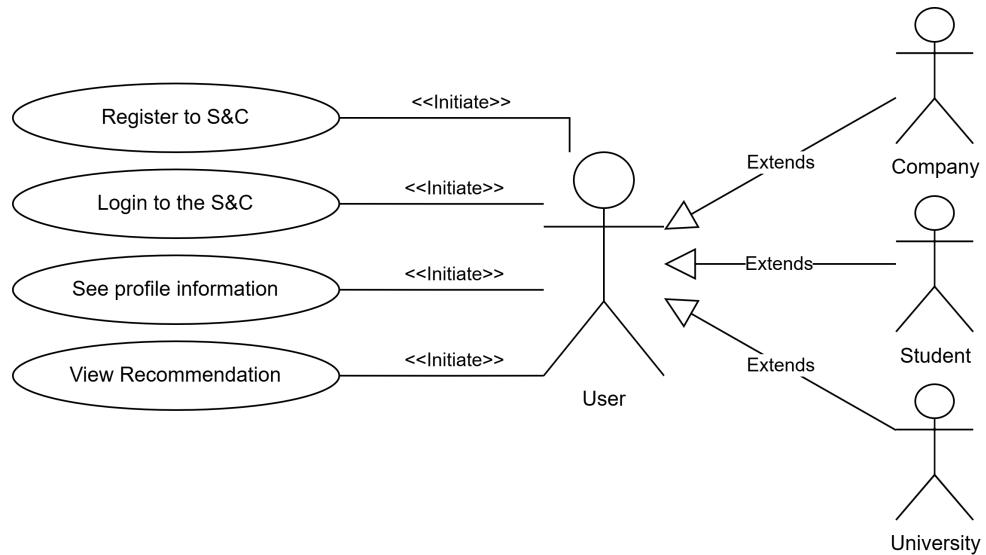


Figure 3.1: User's Use Case Diagram

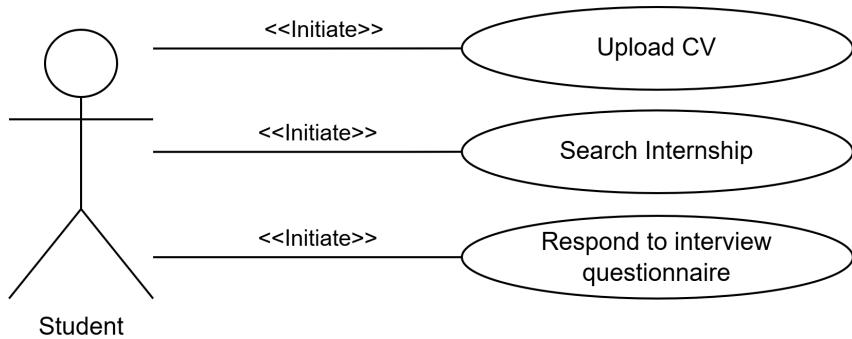


Figure 3.2: Student's Use Case Diagram

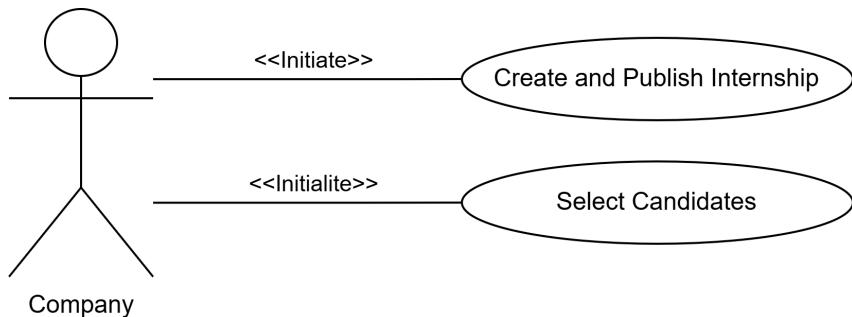


Figure 3.3: Company's Use Case Diagram

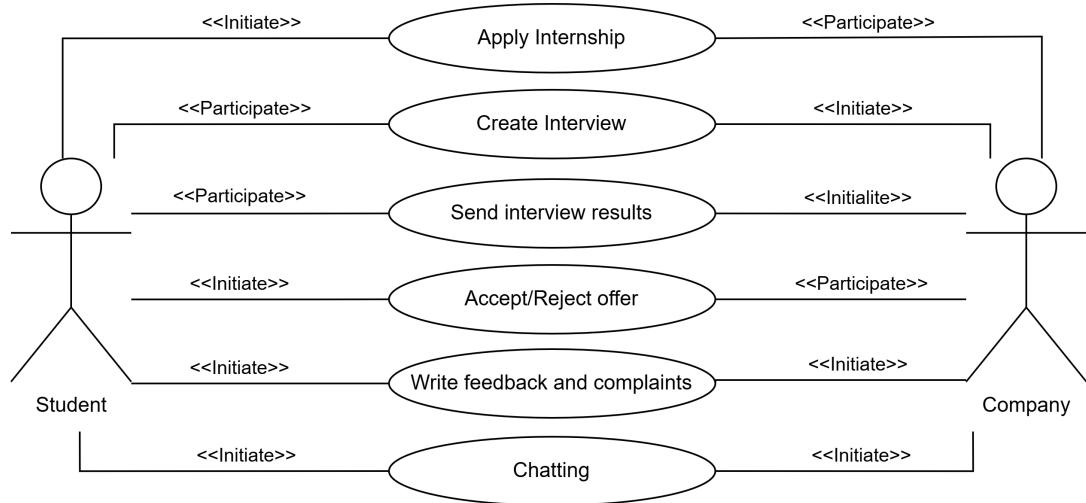


Figure 3.4: Student and Company Use Case Diagram

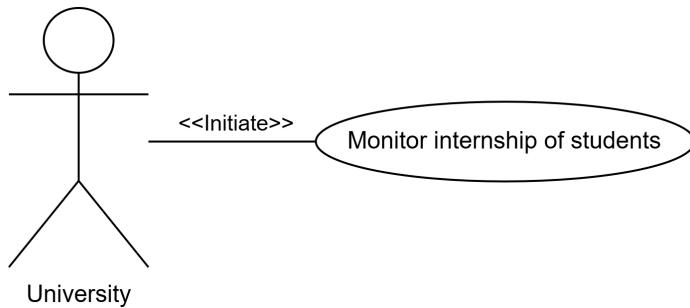


Figure 3.5: University's Use Case Diagram

## Use Case Description

Name	[U1]Register to S&C
Actors	Unregistered User
Entry Condition	<ul style="list-style-type: none"> <li>A user who does not have an account opens the platform.</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>S&amp;C shows the welcome page.</li> <li>The user clicks on the “Create account” button.</li> <li>S&amp;C shows the registration page.</li> <li>The user selects the type of account to create by clicking either on the Student or on the Company button.</li> <li>S&amp;C shows the respective registration form.</li> <li>The user fills in the form with the required information, like: email, password, name, surname, interests, university (for the student).</li> <li>The user clicks on the “Create account” button.</li> <li>S&amp;C checks the information provided by the user.</li> <li>S&amp;C updates the database in order to store all the provided information.</li> <li>S&amp;C sends a confirmation email to the user.</li> <li>The user clicks on the confirmation link in the email.</li> </ol>
Exit Condition	The user is now registered and is granted access to the platform.
Exceptions	In case the information provided by the user is not valid (step 8 fails), for instance the email is already linked to an account, the system will show an error message and ask to provide the correct information, effectively returning to step 5.

<b>Name</b>	[U2]Login to S&C
<b>Actors</b>	User
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• A user who does have an account opens the platform.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. S&amp;C shows the welcome page.</li> <li>2. The user fills in the form with the required information, like: email, password.</li> <li>3. The user clicks on the Login button.</li> <li>4. S&amp;C checks the information provided by the user.</li> <li>5. S&amp;C displays the Dashboard page.</li> </ol>
<b>Exit Condition</b>	The user is now logged in the platform.
<b>Exceptions</b>	In case the information provided by the user is invalid (step 4 fails), for instance the email isn't linked to any account or the password is wrong, an error message that asks to provide the correct information is shown, effectively returning to step 2.

<b>Name</b>	[U3]See profile information
<b>Actors</b>	User
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• User is logged in with the corresponding account.</li> <li>• S&amp;C shows the Dashboard page.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. User clicks on the Search tab or on the search bar in the dashboard page.</li> <li>2. User types the name of the user they want to search in the search bar.</li> <li>3. User click on the Search button.</li> <li>4. S&amp;C shows all the profiles that match the search.</li> <li>5. User clicks on the profile they want to see.</li> <li>6. S&amp;C shows the profile information.</li> </ol>
<b>Exit Condition</b>	S&C displayed the profile information requested by the user.
<b>Exceptions</b>	If no profile is found, a message will be displayed saying that the profile does not exist.

<b>Name</b>	[U4]View recommendation
<b>Actors</b>	Student, Company
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged.</li> <li>• Company is logged.</li> <li>• S&amp;C shows the Dashboard page.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. S&amp;C displays the recommendation section in the Dashboard page.</li> <li>2. The student or company browses the recommendations.</li> <li>3. The student or company clicks on the recommendation they want to see.</li> <li>4. S&amp;C shows the recommendation details.</li> <li>5. The student or company can either accept the recommendation (by applying for the internship, or inviting the student to take part in an internship) or ignore it.</li> </ol>
<b>Exit Condition</b>	S&C displays the recommendation details to the student or company and allows them to do the desired action.
<b>Exceptions</b>	If there are not enough data to provide a recommendation related to the student or company, S&C will show a message that no recommendation is available.

<b>Name</b>	[U5]Upload CV
<b>Actors</b>	Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged.</li> <li>• Student opens the profile modification page.</li> <li>• Student has a CV file to upload.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. The student clicks on the Edit profile button.</li> <li>2. S&amp;C shows the profile modification page.</li> <li>3. The student clicks on the Upload CV button.</li> <li>4. S&amp;C shows the file explorer.</li> <li>5. The student selects the CV file and uploads it.</li> <li>6. S&amp;C checks the file format and size.</li> <li>7. The student clicks on the Save button.</li> <li>8. S&amp;C stores the file in the database.</li> <li>9. S&amp;C displays the updated profile page.</li> </ol>
<b>Exit Condition</b>	The student's CV is uploaded and displayed in the profile page.
<b>Exceptions</b>	In case the file format or size is not valid (step 6 fails), the system will show an error message and ask to upload only files in pdf format, effectively returning to step 2.

<b>Name</b>	[U6]Search internship
<b>Actors</b>	Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged.</li> <li>• S&amp;C shows the Dashboard page.</li> <li>• The student wants to search for an internship.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. The student clicks on the Search tab or on the search bar in the Dashboard page.</li> <li>2. The student types the keyword in the search bar.</li> <li>3. The student clicks on the Search button.</li> <li>4. S&amp;C shows the internships that match the keyword.</li> <li>5. The student either clicks on an internship to see more details or scrolls down to see more internships.</li> </ol>
<b>Exit Condition</b>	The student has found the internship they were looking for. The student does not find what they were looking for, so they close the search page or retry a new search.
<b>Exceptions</b>	In case the keyword doesn't match any internship (step 4 fails), S&C shows the message: "No internship was found, please try with a different keyword", effectively returning to step 2.

<b>Name</b>	[U7]Apply internship
<b>Actors</b>	Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged.</li> <li>• S&amp;C shows the Internship details page.</li> <li>• The student wants to apply to the internship.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. The student clicks on the Apply button.</li> <li>2. S&amp;C gets the student's information from the database.</li> <li>3. S&amp;C checks that the student has not already applied for that internship or is starting an internship in the same period.</li> <li>4. S&amp;C creates and stores the application in the database.</li> <li>5. S&amp;C shows the details of the application in the My applications page.</li> </ol>
<b>Exit Condition</b>	The student has applied for the internship.
<b>Exceptions</b>	The student has already applied for that internship (step 3 fails), S&C will show an error message and ask to check the My applications page to see the status of the application.

<b>Name</b>	[U8]Create interview
<b>Actors</b>	Company, Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Company is logged in with the corresponding account.</li> <li>• Company has a list of students selected to create the interview for.</li> <li>• Student is selected by the company for the interview process.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Company sets up the questions that will be asked to the student in the questionnaire and other information about the interview.</li> <li>2. Company clicks on the “Submit” button.</li> <li>3. S&amp;C checks the list of students and the information filled by the company.</li> <li>4. S&amp;C stores the data in the database.</li> <li>5. S&amp;C sends a notification to Student about the interview.</li> </ol>
<b>Exit Condition</b>	Information related to the interview is received and stored by S&C, the interview information is updated and the notification is sent to the student who is waiting for the interview process to start.
<b>Exceptions</b>	The student selected by the company has already been accepted by another company and has started the internship at the same time. Then the system will notify the company that to select another student from the list of the candidates.

<b>Name</b>	[U9] Respond to interview questionnaire
<b>Actors</b>	Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged.</li> <li>• Student was selected by the company for the interview process.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Student clicks on the notification informing them about the interview or opens the interview page from the application page.</li> <li>2. S&amp;C shows the interview questionnaire.</li> <li>3. Student fills in the questionnaire.</li> <li>4. Student clicks on the “Submit” button.</li> <li>5. S&amp;C checks the answers provided by the student.</li> <li>6. S&amp;C stores the student’s answers in the database.</li> <li>7. S&amp;C updates the status of the student’s application.</li> </ol>
<b>Exit Condition</b>	S&C shows the student that the questionnaire has been submitted successfully.
<b>Exceptions</b>	The student does not correctly fill in the questionnaire (step 5 fails), S&C will ask the student to correctly fill the questionnaire, effectively returning to step 2.

<b>Name</b>	[U10]Send interview results
<b>Actors</b>	Company, Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged.</li> <li>• Company is logged.</li> <li>• Company has to decide who to accept for the internship.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. The Company opens the Internship Management page.</li> <li>2. The Company clicks on Interview Management page.</li> <li>3. S&amp;C shows the list of students who have completed the questionnaire.</li> <li>4. The Company selects the students to accept for the internship.</li> <li>5. The Company clicks on the Send results button.</li> <li>6. S&amp;C checks the list of students selected by the company.</li> <li>7. S&amp;C stores the data in the database.</li> <li>8. S&amp;C sends a notification to the students who have been accepted, asking them to confirm or reject the offer.</li> </ol>
<b>Exit Condition</b>	The company has sent the results of the interview to the students.
<b>Exceptions</b>	There is a problem with the company selection (step 6 fails), S&C will notify the company that the selection has not been saved and ask to try again, effectively returning to step 3.

<b>Name</b>	[U11]Accept/reject offer
<b>Actors</b>	Company, Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student is logged in with the corresponding account.</li> <li>• Student has received an offer from the company.</li> <li>• Company is waiting for the student response.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Student click on the notification regarding the offer.</li> <li>2. S&amp;C displays the details to the student and asks to accept or reject the offer.</li> <li>3. Student clicks on the “accept” or “reject” button.</li> <li>4. S&amp;C checks the student’s choice and sends a notification to the related company.</li> <li>5. S&amp;C registers the student’s decision and updates the status of the student’s application.</li> </ol>
<b>Exit Condition</b>	The student’s decision is saved in the system, the status of the application of the student is updated, and the company who has sent the offer is notified about the student’s decision.
<b>Exceptions</b>	<p>The student has already accepted another internship. S&amp;C will notify the student that they cannot accept more than one offer if the two internships start in the same period.</p> <p>The deadline for accepting the offer has passed. The system will notify the student that it is too late to accept the offer.</p>

<b>Name</b>	[U12]Write feedback and complaints
<b>Actors</b>	Company, Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student and Company are logged in.</li> <li>• Student and company have taken part in an active internship.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Student or company clicks on the active internship which they want to leave comments on.</li> <li>2. S&amp;C displays the feedback and complaints page related to that internship.</li> <li>3. Student or company writes in the box dedicated to the comments.</li> <li>4. Student or company clicks on the “Submit” button.</li> <li>5. S&amp;C stores the comments and displays them on the evaluation page.</li> <li>6. S&amp;C notifies the Student and company that there is a new feedback or complaint regarding specific internship.</li> </ol>
<b>Exit Condition</b>	The comments are stored in the database and displayed on the internship evaluation page. The student and company are notified about the new feedback.
<b>Exceptions</b>	The student or company send an empty comment, the system will notify them that the comment can not be empty.

<b>Name</b>	[U13]Chatting
<b>Actors</b>	Company, Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student and Company are logged in.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Student or company clicks on the chat icon.</li> <li>2. S&amp;C displays the list of the available chats.</li> <li>3. Student or company selects the chat where they want write.</li> <li>4. Student or company write the message in the message box and click on the “Send” button.</li> <li>5. S&amp;C receives the message and stores it in the chat history database.</li> <li>6. S&amp;C sends it to the recipient and displays it in the chat.</li> <li>7. S&amp;C notifies the recipient that there is a new message.</li> </ol>
<b>Exit Condition</b>	The message is correctly sent and stored by S&C. The recipient and the sender's chat history are updated and the recipient is notified about the new message incoming, the message details are displayed successfully in the chat.
<b>Exceptions</b>	<p>If the content of the message is empty, S&amp;C will notify the sender that the message can not be empty.</p> <p>If there is no chat available in the chat list, S&amp;C will display a message saying the chats will be available when the student or company will have taken part in an internship at least once.</p>

<b>Name</b>	[U14]Create and Publish Internship
<b>Actors</b>	Company
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Company is logged in with the corresponding account.</li> <li>• Company has all necessary information to create an internship announcement.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Company opens the “Publish internship” page.</li> <li>2. S&amp;C displays the page with the form to fill in the internship announcement details.</li> <li>3. Company inserts the data needed to complete the form and clicks on the “Submit” button.</li> <li>4. S&amp;C checks the data inserted by the company and stores it in the database.</li> <li>5. S&amp;C displays updated list of the internship available related to this company.</li> <li>6. S&amp;C displays the internship announcement on the platform.</li> </ol>
<b>Exit Condition</b>	S&C successfully displays the internship announcement on the platform.
<b>Exceptions</b>	If there is a missing field in the form, S&C will notify the company that all fields are required to be filled.

<b>Name</b>	[U15]Select Candidates
<b>Actors</b>	Company, Student
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• Student and Company are logged in.</li> <li>• Company has published an internship and the application deadline has passed.</li> <li>• Student has applied for this internship.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. Company clicks on the internship announcement which they wants to select the candidates.</li> <li>2. S&amp;C displays the Internship management page.</li> <li>3. Company clicks on the “Select candidates” button.</li> <li>4. S&amp;C displays the list of the students who have submitted the application.</li> <li>5. Company clicks on the student’s profile to see the details, in particular the CV.</li> <li>6. S&amp;C displays the student’s profile and CV to the company.</li> <li>7. Company selects the students who will be invited for the interview.</li> <li>8. Company click on the “Submit” button.</li> <li>9. S&amp;C stores the list of the selected students and notifies them about the application result in this phase.</li> <li>10. S&amp;C updates the status of the applications of the students who have applied for this internship.</li> </ol>
<b>Exit Condition</b>	S&C stores the decision of the company and updates the status of the application of the students and notifies them about the result.
<b>Exceptions</b>	<p>If the company has selected a student who has already accepted an offer from another company, S&amp;C will notify the company that the student has already accepted an offer from another company, and ask to select another student from the list of the candidates.</p> <p>If the number of selected students is less or more than the number of the students specified in announcement, S&amp;C will notify the company to select more or fewer students.</p>

<b>Name</b>	[U16]Monitor internship of students
<b>Actors</b>	University
<b>Entry Condition</b>	<ul style="list-style-type: none"> <li>• A university is logged in with the corresponding account.</li> <li>• The university has students who have taken part in internships.</li> <li>• The university representative wants to monitor the internship process of the students.</li> </ul>
<b>Events Flow</b>	<ol style="list-style-type: none"> <li>1. S&amp;C shows the Dashboard page with the list of all registered students of the university.</li> <li>2. The University representative clicks on a student's profile.</li> <li>3. S&amp;C shows the student's profile page.</li> <li>4. The University representative clicks on one of the student's internships (current or past).</li> <li>5. S&amp;C shows the internship details page.</li> <li>6. The University representative can see the feedback and complaints from both parties involved related to that internship.</li> </ol>
<b>Exit Condition</b>	The university representative has monitored all the internship processes of the students that they were interested in.
<b>Exceptions</b>	If the student has not taken part in any internship, S&C will communicate this to the representative with a message that says "This student has not taken part in any internship yet".

## Sequence Diagrams

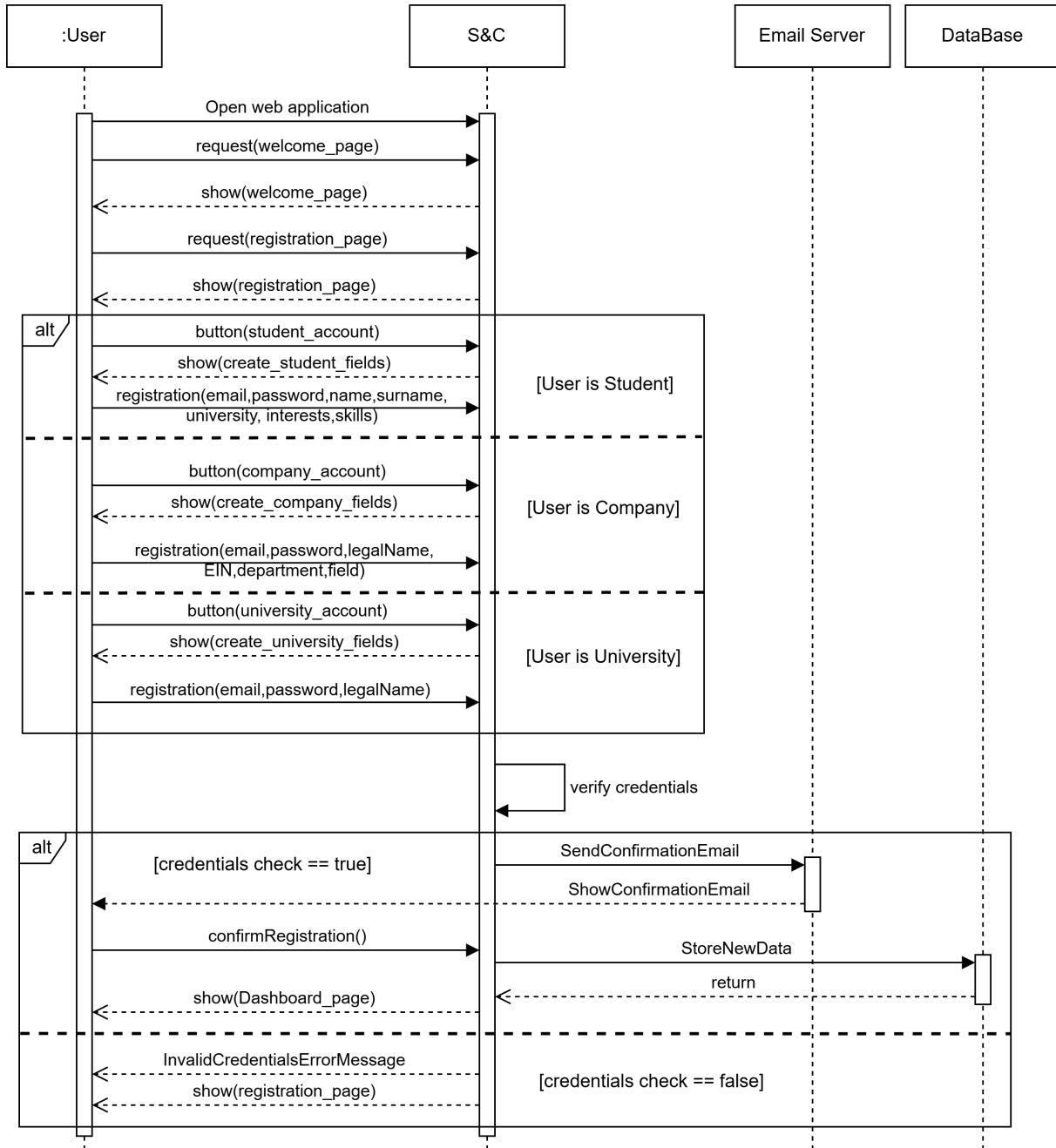


Figure 3.6: Registration to S&C Sequence Diagram

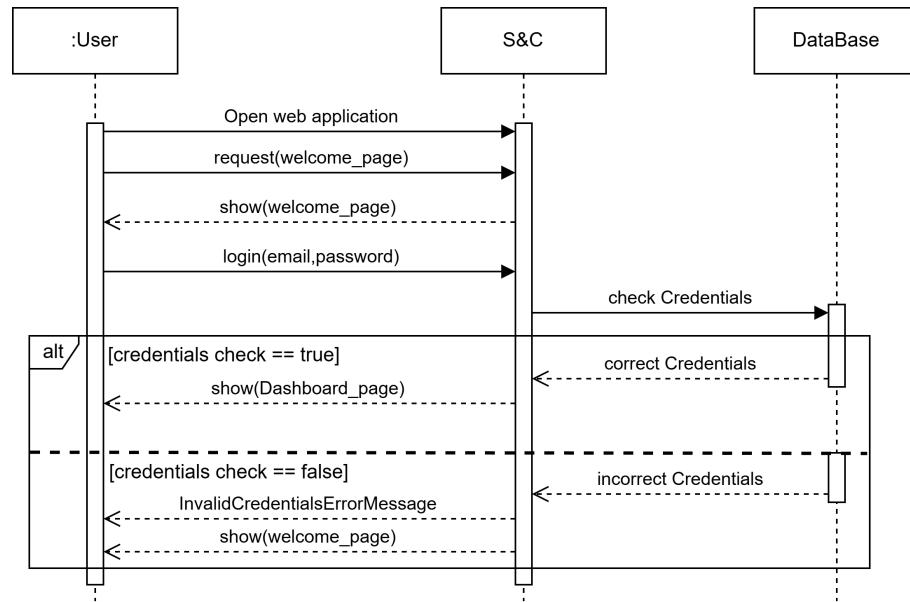


Figure 3.7: Login to S&amp;C Sequence Diagram

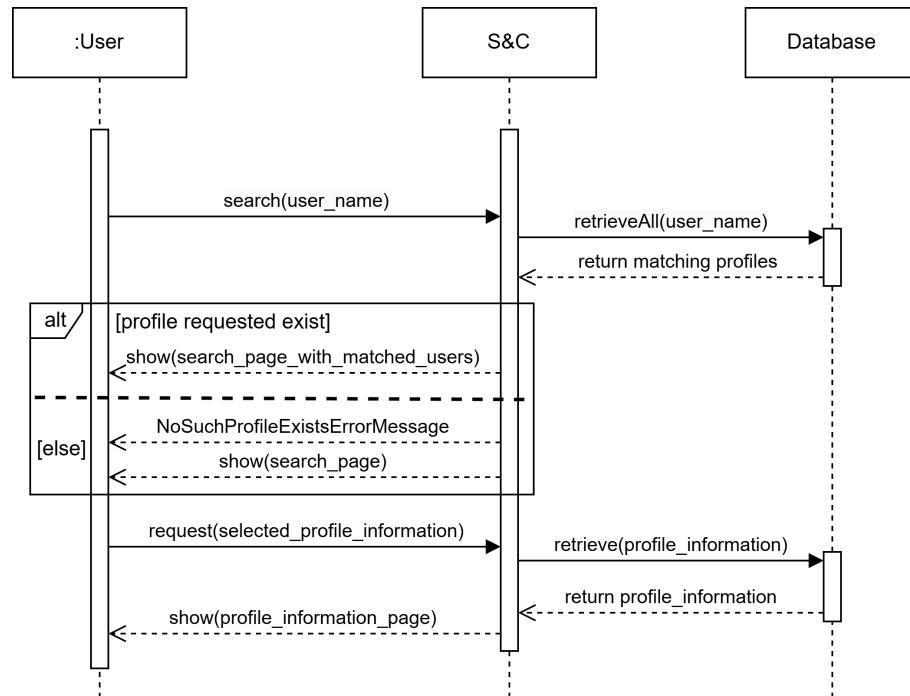


Figure 3.8: User sees profile information Sequence Diagram

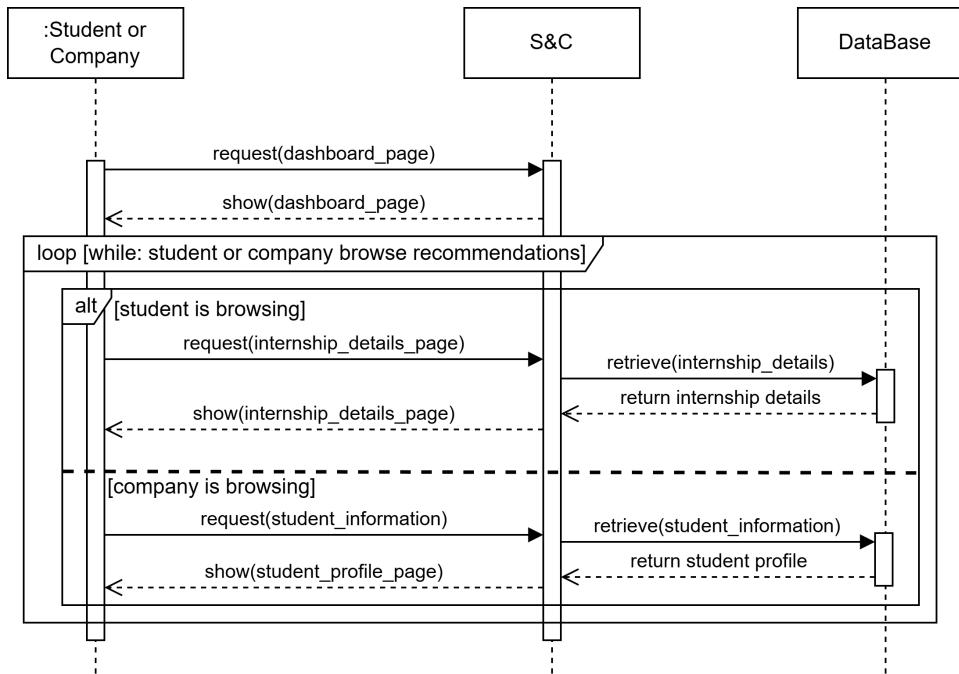


Figure 3.9: Student or Company views recommendation Sequence Diagram

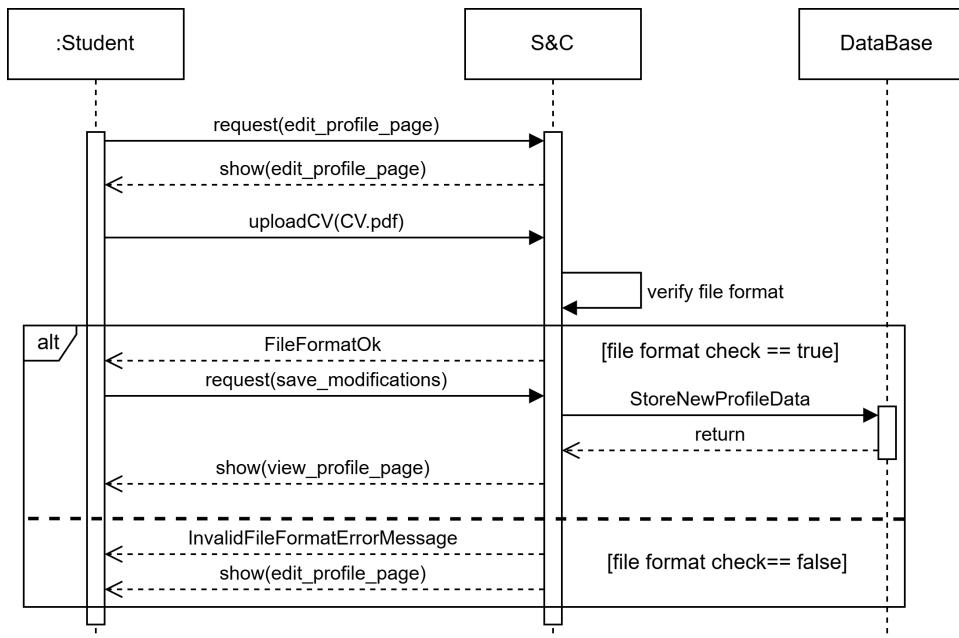


Figure 3.10: Student uploads CV to his profile Sequence Diagram

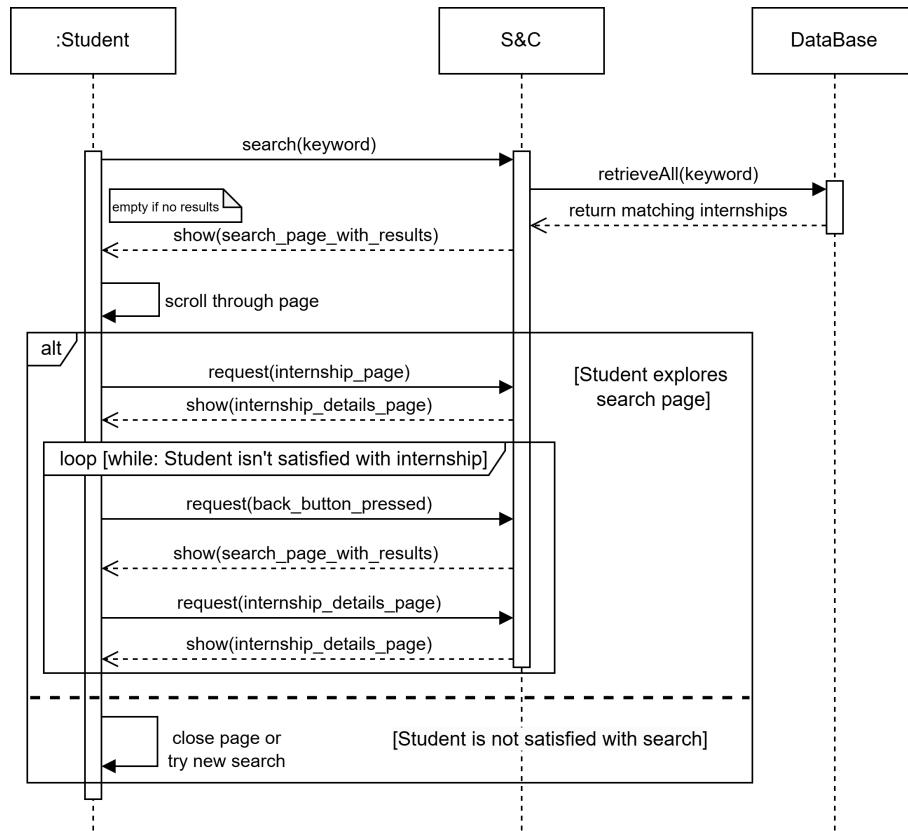


Figure 3.11: Student searches for an internship Sequence Diagram

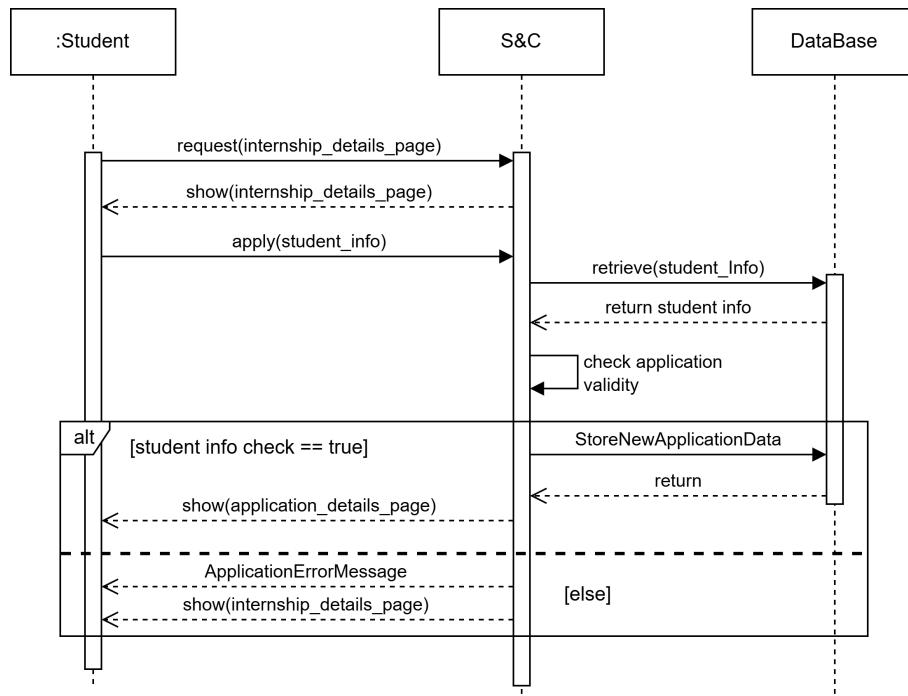


Figure 3.12: Student applies for an internship Sequence Diagram

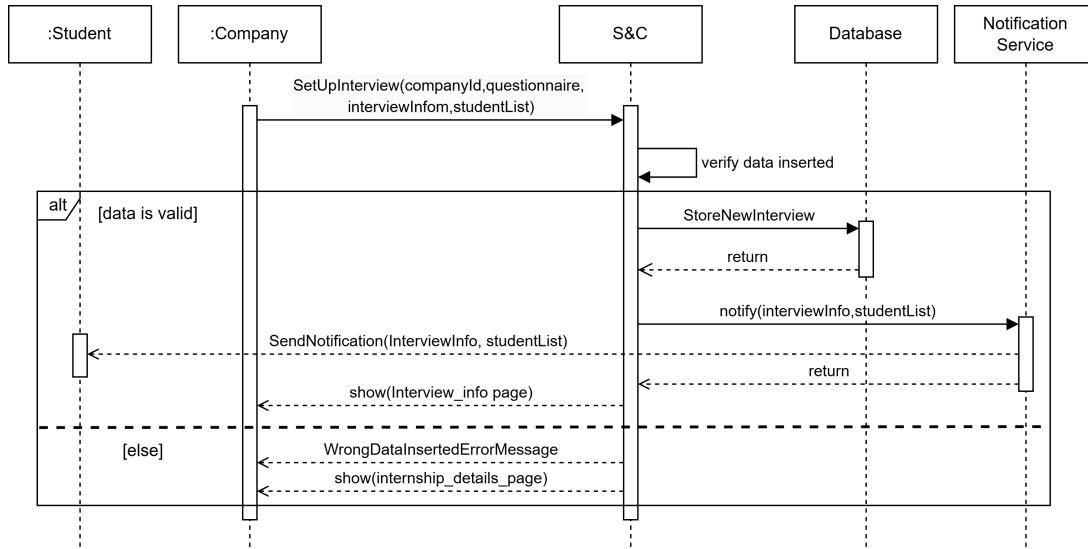


Figure 3.13: Company creates an interview Sequence Diagram

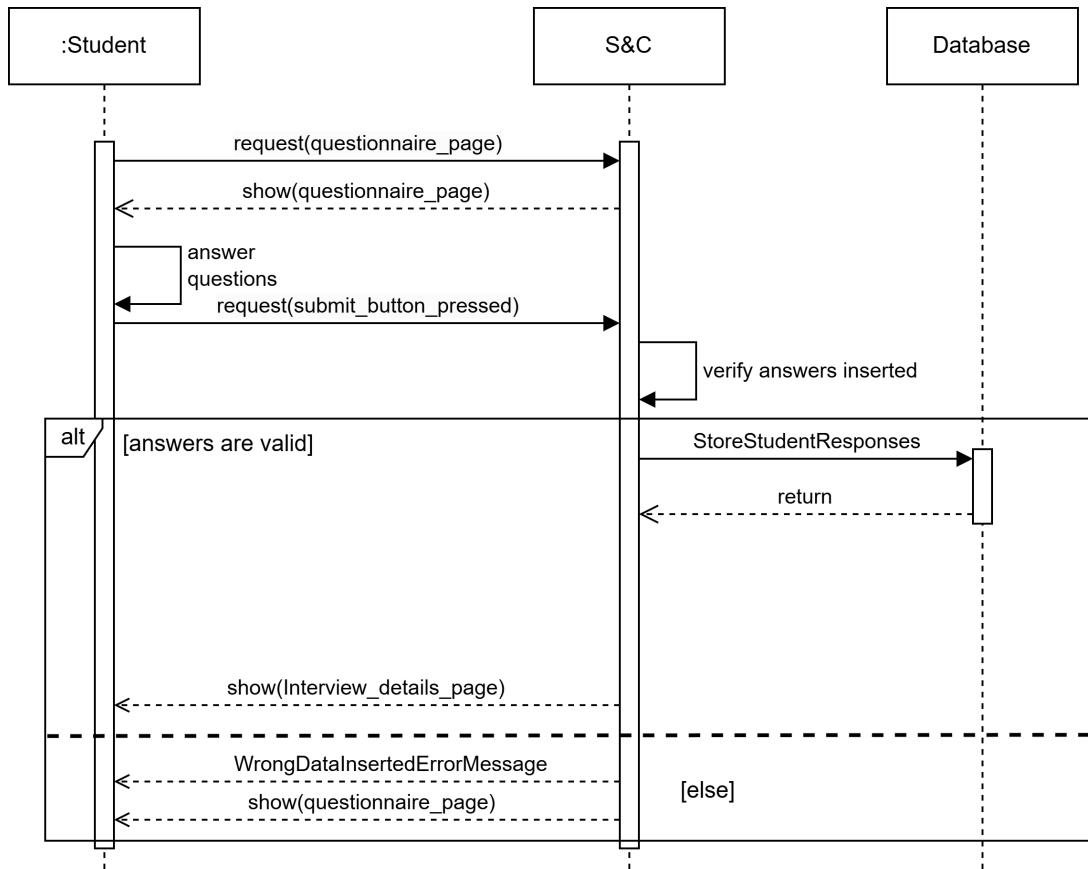


Figure 3.14: Student responds to an interview questionnaire Sequence Diagram

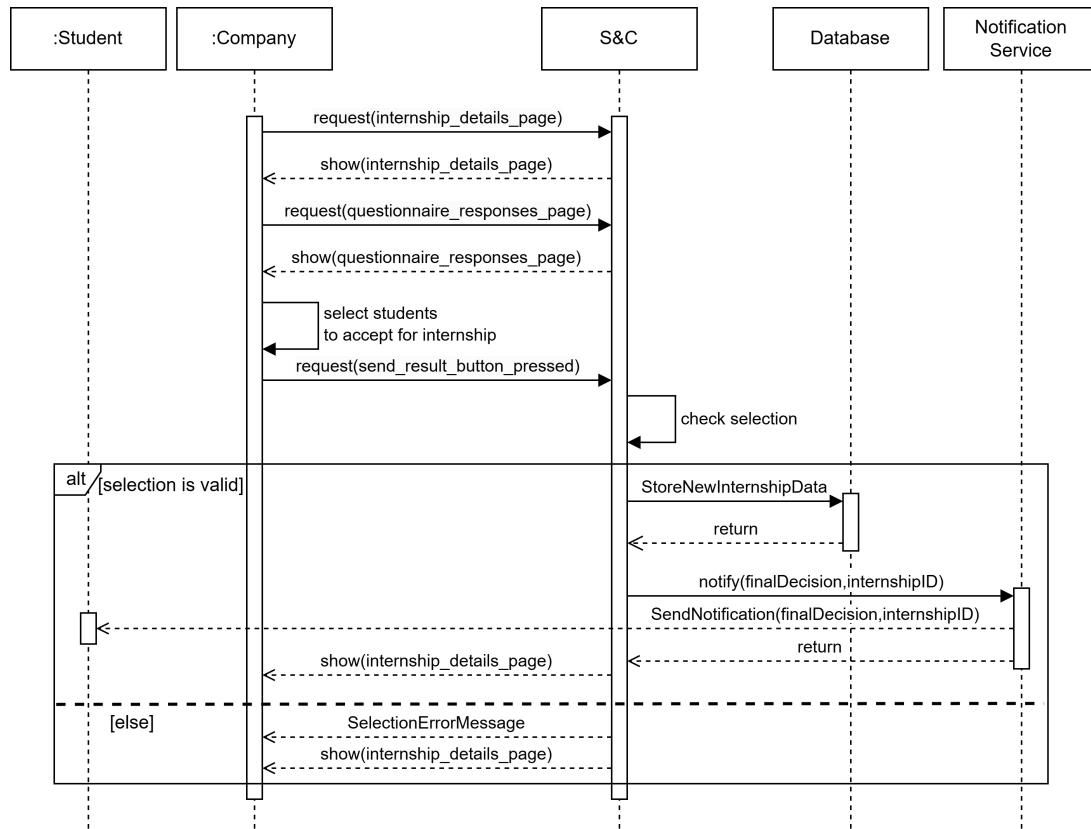


Figure 3.15: Company sends interview results Sequence Diagram

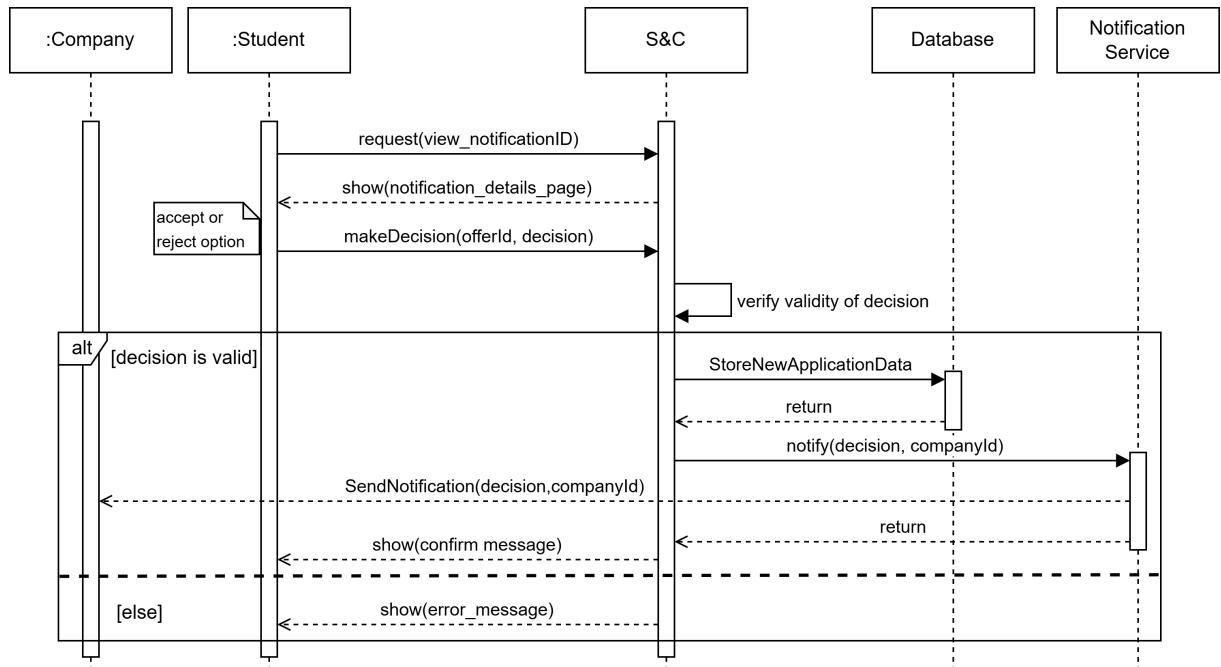


Figure 3.16: Student accepts/rejects an offer Sequence Diagram

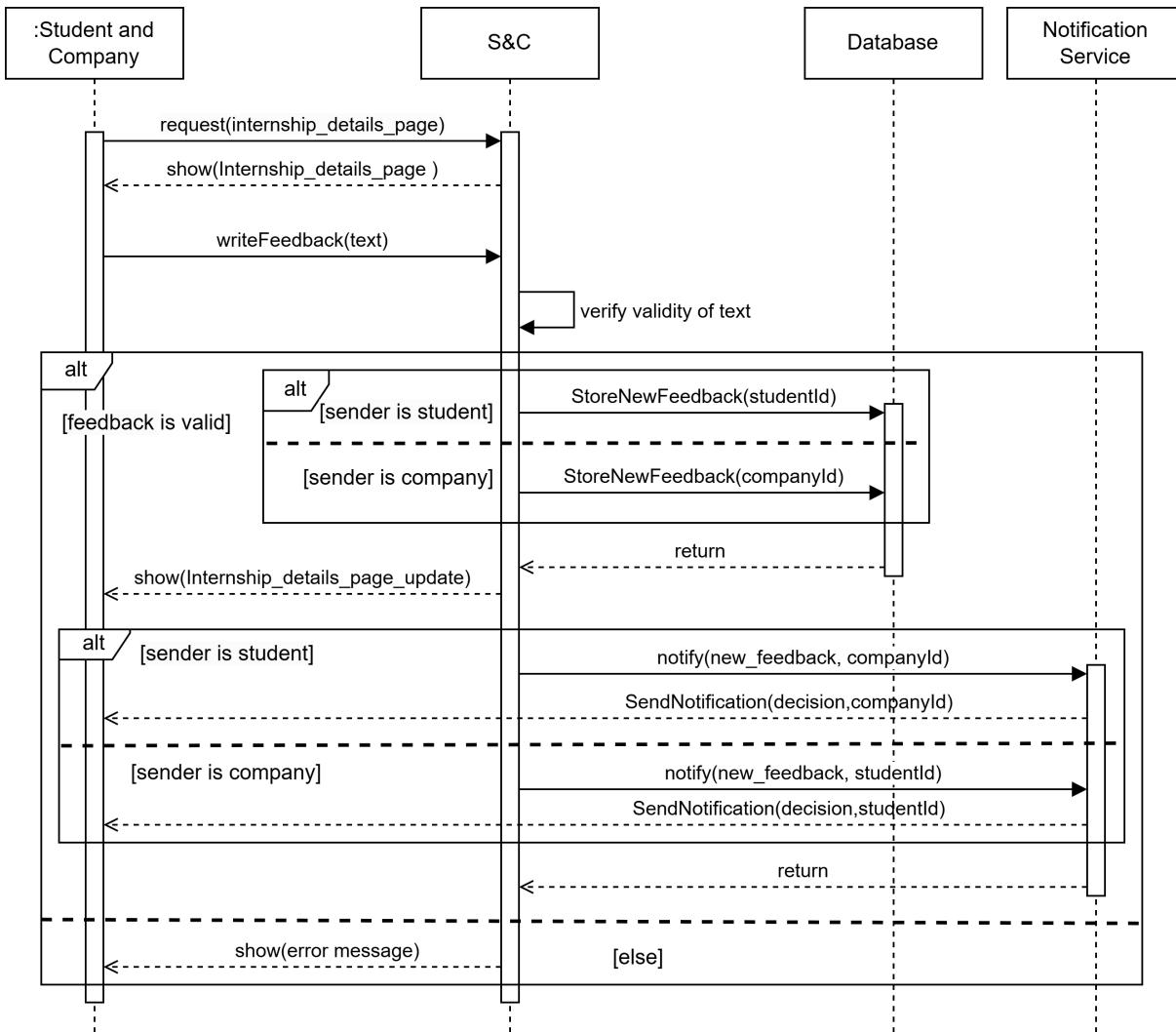


Figure 3.17: Student or Company writes feedback Sequence Diagram

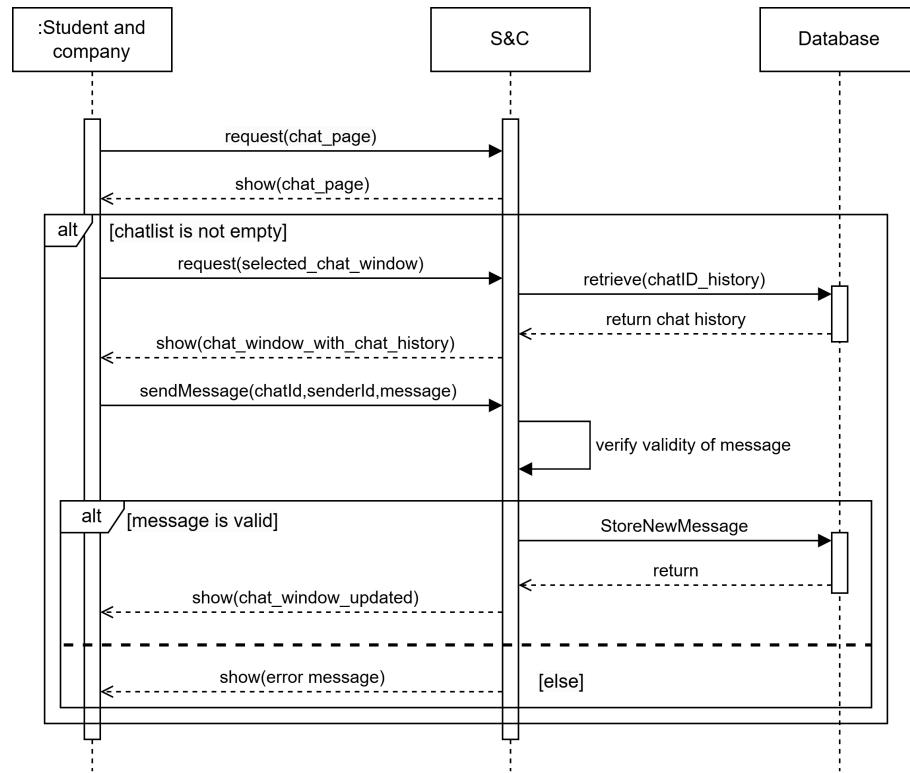


Figure 3.18: Student or Company chats with each other Sequence Diagram

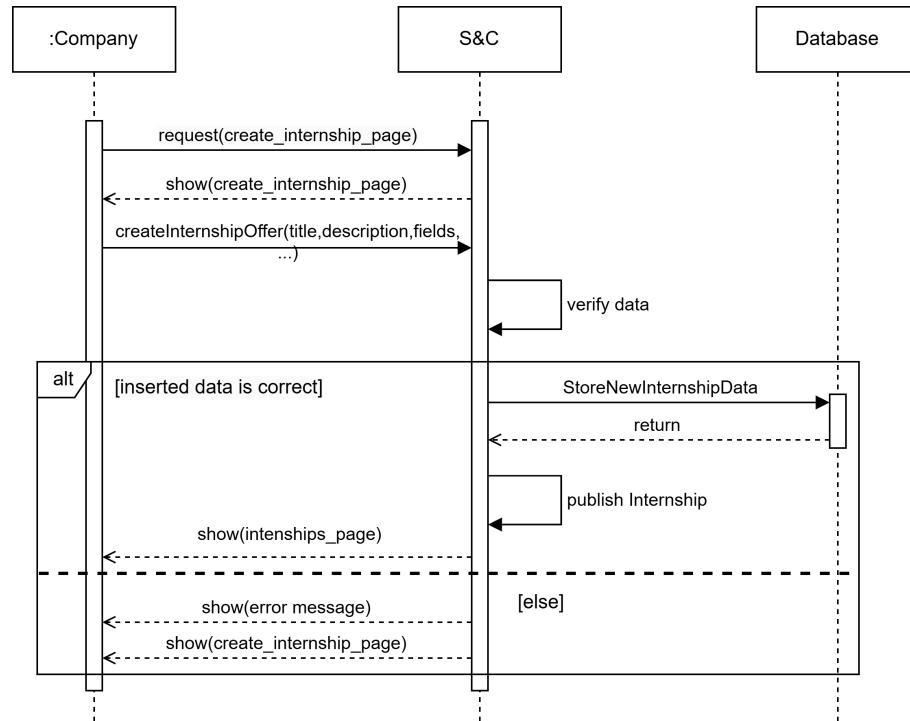


Figure 3.19: Company creates and publishes an internship Sequence Diagram

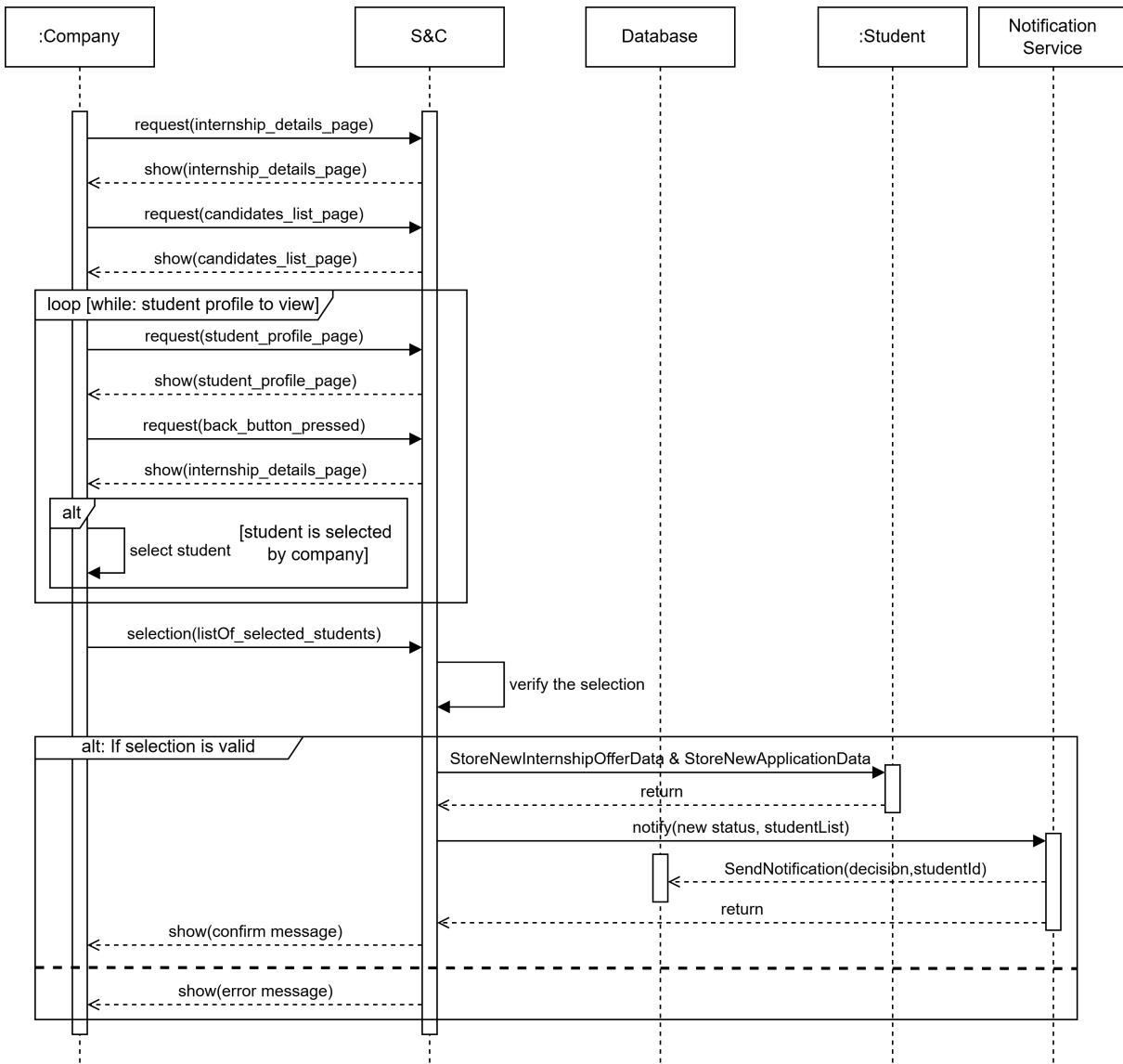


Figure 3.20: Company selects candidates Sequence Diagram

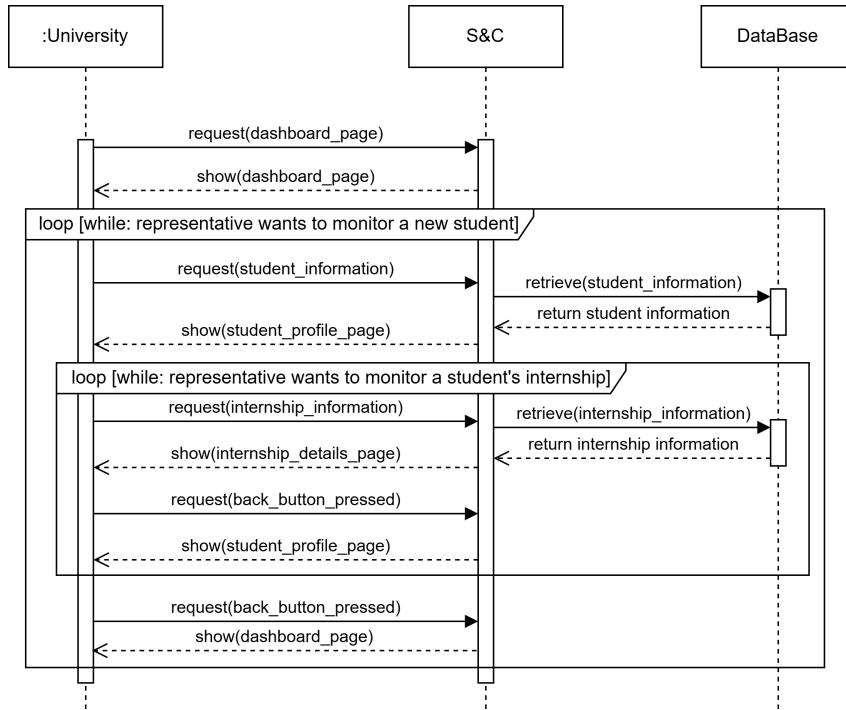


Figure 3.21: University monitors the internship processes of students Sequence Diagram

#### B.4. Traceability Matrix

Mapping the requirements and goals to the use cases using the following matrix:

Note: *Some requirements are mapped to multiple use cases because they relate to multiple functionalities. However, in the matrix, we show only the one use case that is most relevant to the requirement.*

Raw ID	Req. ID	Goal ID	Principal Use Case
1	R1	G1	U1.Register to S&C
2	R2	G1	U2.Login to S&C
3	R3	G2	U5.Upload CV
4	R4	G3	U6.Search internship
5	R5	G4	U14.Create and Publish Internship
6	R6	G4	U14.Create and Publish Internship
7	R7	G4	U14.Create and Publish Internship
8	R8	G5	U15.Select Candidates
9	R9	G5	U4. View recommendation

*Continued on next page*

<b>Raw ID</b>	<b>Req. ID</b>	<b>Goal ID</b>	<b>Principal Use Case</b>
<b>10</b>	R10	G5	U10.Send interview results
<b>11</b>	R11	G5	U15.Select Candidates
<b>12</b>	R12	G5	U11.Accept/reject offer
<b>13</b>	R13	G5, G6	U4.View recommendation
<b>14</b>	R14	G5	U15.Select Candidates
<b>15</b>	R15	G5	U11.Accept/reject offer
<b>16</b>	R16	G5	U11.Accept/reject offer
<b>17</b>	R17	G5	U13.Chatting
<b>18</b>	R18	G5, G6	U4.View recommendation
<b>19</b>	R19	G6	U4.View recommendation
<b>20</b>	R20	G7	U7.Apply internship
<b>21</b>	R21	G3, G7	U6.Search internship
<b>22</b>	R22	G8	U15.Select Candidates
<b>23</b>	R23	G8	U15.Select Candidates
<b>24</b>	R24	G8	U8.Create interview
<b>25</b>	R25	G9	U9.Respond to interview questionnaire
<b>26</b>	R26	G10	U10.Send interview results
<b>27</b>	R27	G10	U12.Send interview results
<b>37</b>	R28	G11	U14.Accept/reject offer
<b>29</b>	R29	G11	U14.Accept/reject offer
<b>30</b>	R30	G12	U12.Write feedback and complaints
<b>31</b>	R31	G12	U12.Write feedback and complaints
<b>32</b>	R32	G13	U13.Chatting
<b>33</b>	R33	G14	U16.Monitor internship of students
<b>34</b>	R34	G14	U16.Monitor internship of students
<b>35</b>	R35	G14	U3.See profile information
<b>36</b>	R36	G2	U5.Upload CV
<b>37</b>	R37	G5, G12	U1.Register to S&C

## C. Performance Requirements

- **Concurrent access of users and resource utilization:** A platform like Students&Companies is expected to have a large number of users, so it must be able to handle multiple requests simultaneously. By searching online for platforms that offer similar services we have found that a good estimate for the number of concurrent users that the platform should be able to manage is roughly a few thousands, peaking at a few tens of thousands during peak internship publishing periods (like at the end of universities semesters).

The platform should be able to optimize the resource utilization to ensure that the system can handle the load without any performance issues. This includes optimizing the database queries, caching data, and using load balancing techniques to distribute the load across multiple servers.

- **Data processing and storage:** The system should be able to process and store a large amount of data efficiently. From what we were able to find online, similar platforms estimate that the number of registered users can reach a few hundreds of thousands, and the number of internships published can reach a few tens of thousands. For this reason, the database should be able to easily handle a few terabytes of data.

The platform should also be able to process data quickly, especially when performing statistical analyses in order to generate recommendations for students and companies. This includes optimizing the queries made by recommendation algorithm to ensure that it can provide recommendations in real-time.

- **Time of Response:** From the users' perspective, the system should be responsive, meaning the response to any of his request should appear instantly. In order to achieve this, the response time for most operations, such as loading a page, submitting an application, or performing a search, should be at most a few seconds during peak usage and in the domain of milliseconds in normal conditions.

Particular attention should be given to the recommendation algorithm, which should be able to provide recommendations in real-time, to the chat functionality, which should allow users to communicate in real-time, and to the notification system, who must ensure that updates are delivered to the user before relevant deadlines expire. The response time of other operations such as the ones that involve the mailing system cannot be guaranteed by S&C.

## D. Design Constraints

### D.1. Standards Compliance

The platform should comply with the REST API standard in order to correctly process user inputs.

The system must be compliant with the European Union's General Data Protection Regulation (GDPR), which is a set of regulations that is designed to protect the privacy and personal data of individuals within the European Union. This means that the platform must ensure that user data is collected and processed in a lawful and transparent manner, and that users have the right to access, correct, and delete their data.

The platform should also comply with the Web Content Accessibility Guidelines (WCAG) to ensure that the platform is accessible to users with disabilities. This includes providing multiple ways to navigate the platform, for example by showing alternative text for images.

Since the users accessing the platform could be from different countries and timezones, the platform should use a time standard like UTC to ensure that all dates and times are consistent across different regions and that deadlines can be communicated and handled without ambiguity.

### D.2. Hardware Limitations

The platform is a web-based application, so it should be able to run on any device with an internet connection and a compatible web browser. Furthermore, it should not require high level or specific hardware.

### D.3. Any Other Constraint

S&C is intended for students, companies and universities only, so the platform should not be accessible to users who do not belong to these categories.

The target of the platform is the European market. Since users may speak different languages, it should be designed completely in English, as it is the most widely spoken language and is commonly used in the business and academic world.

## E. Software System Attributes

## E.1. Reliability

The platform should be reliable and ensure that the data is always available and consistent. Particular attention should be given to the recommendation algorithm, which should be programmed with the utmost care so that it is always working properly and recommendations are always accurate. This is because uninteresting and unfit recommendations could lead to a decrease in the platform's popularity and a loss of users.

The system should also be able to cope with partial failures through replication and recover from failures quickly and without data loss. This includes implementing regular backups of the database and the state.

## E.2. Availability

The system should have a required uptime of at least 99.8%, which means that the platform should be available 99.8% of the time. This is equivalent to a downtime of less than 18 hours per year. To achieve this, the platform should be designed with fault tolerance in mind to ensure that the system can handle failures without affecting the availability.

During the downtime period, a maintenance page should be displayed to inform users that the platform is being updated or is experiencing technical difficulties and is currently unavailable. In order to avoid the expiration of a deadline during the downtime, planned maintenance should be scheduled during off-peak hours, such as late at night or early in the morning.

## E.3. Security

In order to guarantee a secure system, the platform should control the access rights of the users, ensuring both authentication, meaning that the identity of users that attempt to login must be verified, and authorization, meaning that the permission of users to perform specific actions must be verified. As an example, a student should not be allowed to create an internship.

To comply with the GDPR, the platform should encrypt all sensitive data, such as passwords and personal information, using secure communication protocols, like HTTPS, TLS, and specific algorithms.

## E.4. Scalability

The system should be designed to be scalable, meaning that it shouldn't sacrifice performance as the number of concurrent users and stored data grows. This includes the ability to scale horizontally by adding more servers in order to distribute the load and the ability to scale vertically by upgrading the backend hardware to increase the system's capacity. In particular, the recommendation algorithm must be able to scale well. It should be optimized to handle a larger number of users and data, and ensure that it can provide recommendations in real-time.

## E.5. Maintainability

The platform should be easy to maintain and update. This includes writing clean, understandable and well-documented code, using version control systems to track changes, and following best practices for software development, like writing unit tests and using continuous integration and deployment tools.

The platform should also be designed to be modular, meaning that different components should be decoupled and independent from each other, so that they can be updated or replaced by different development teams without affecting the rest of the system.

## E.6. Portability

The system should be compatible with any kind of device that has an internet connection and a compatible web browser. This includes desktops, laptops, tablets, and smartphones.

# 4 | Formal Analysis Using Alloy

In this section we are going to provide a formal description of the system-to-be using the Alloy 6 language. We have separated the description in two parts: the first one is the static part, which describes the structure of the system, and the second one is the dynamic part, which describes the behavior of the system and how it evolves in time.

## A. Static part

This model aims to describe the structure of the system, focusing on the main entities and their relationships. In particular, we are going to describe the relationships between internships, applications, interviews and users of the platform.

### Signatures

The following signatures describe the main entities of the system:

---

```

abstract sig User {}           // Abstract signature for all users
// Student class to model its submitted applications and participations in
// internships
sig Student extends User {
    submits: set Application,   // Applications submitted by student
    participates: set Internship // Internships student has participated in
}
// Company signature to model companies' published internships
sig Company extends User {
    publishes: set Internship   // Internships published by company
}
// University signature to model universities' students
sig University extends User {
    enrolls: set Student        // Students enrolled in university
}
// Internship signature to model internships' applications and feedbacks

```

```

sig Internship {
    submissions: set Application, // Applications submitted by students for
        internship
    submittedFeedbacks: set Feedback // Feedbacks submitted by student who
        participated in internship
}
// Application signature to model applications' interviews
sig Application {
    interview: lone Interview // Interview scheduled for application
}
// Feedback signature to model feedbacks submitted by students who
// participated in internships and companies who published them
sig Feedback {}
// Interview signature to model interviews scheduled for applications
sig Interview {}

```

---

## Facts

The following facts describe the constraints between the entities and relationships of the system:

```

// Fact to ensure that each application is submitted by only one student (no
// application can be submitted by multiple students)
fact oneStudentPerApplication{
    all s1,s2: Student | no a: Application | (a in s1.submits and a in
        s2.submits and s1 != s2)
}

// Fact to ensure that all applications have been submitted by a student (no
// orphan applications)
fact allApplicationsInStudent{
    all a: Application | some s: Student | a in s.submits
}

// Fact to ensure that a student cannot submit multiple applications for the
// same internship
fact noRepeatedApplicationsForSameInternship{
    all s: Student | no i: Internship |
        (some a1,a2: s.submits | (a1 in i.submissions and a2 in i.submissions
            and a1 != a2))
}
```

```

// Fact to ensure that all internships that a student has participated in have
// an application submitted by the student
fact allStudentInternshipsHaveApplication{
    all s: Student, i: s.participates | some a: s.submits | a in i.submissions
}

// Fact to ensure that each internship is published by only one company (no
// internship can be published by multiple companies)
fact oneCompanyPerInternship{
    all c1,c2: Company | no i: Internship | (i in c1.publishes and i in
        c2.publishes and c1 != c2)
}

// Fact to ensure that all internships have been published by a company (no
// orphan internships)
fact allInternshipsInCompany{
    all i: Internship | some c: Company | i in c.publishes
}

// Fact to ensure that each student is enrolled in only one university (no
// student can be enrolled in multiple universities)
fact oneUniPerStudent{
    all u1,u2: University | no s: Student | (s in u1.enrolls and s in
        u2.enrolls and u1 != u2)
}

// Fact to ensure that all students are enrolled in a university (no orphan
// students)
fact allStudentsInUni{
    all s: Student | some u: University | s in u.enrolls
}

// Fact to ensure that each application is submitted for only one internship
// (no application can be submitted for multiple internships)
fact oneInternshipPerApplication{
    all i1,i2: Internship | no a: Application | (a in i1.submissions and a in
        i2.submissions and i1 != i2)
}

// Fact to ensure that all applications have been submitted for an internship
// (no orphan applications)
fact allSelectedStudentsApplicationsInInternship{
    all a: Application | some i: Internship | a in i.submissions
}

// Fact to ensure that each feedback is submitted for only one internship (no

```

```

    feedback can be submitted for multiple internships)
fact oneInternshipPerFeedback{
    all i1,i2: Internship | no f: Feedback | (f in i1submittedFeedbacks and f
        in i2submittedFeedbacks and i1 != i2)
}

// Fact to ensure that all feedbacks have been submitted for an internship (no
// orphan feedbacks)
fact allFeedbacksInInternship{
    all f: Feedback | some i: Internship | f in isubmittedFeedbacks
}

// Fact to ensure that each interview is scheduled for only one application
// (no interview can be scheduled for multiple applications)
fact oneApplicationPerInterview{
    all a1,a2: Application | no intv: Interview | (intv in a1interview and
        intv in a2interview and a1 != a2)
}

// Fact to ensure that all interviews have been scheduled for an application
// (no orphan interviews)
fact allInterviewsInApplication{
    all intv: Interview | some a: Application | intv in ainterview
}

// Fact to ensure that a feedback is submitted only if at least a student has
// participated in the internship
fact feedbackOnlyIfStudentInInternship{
    all f: Feedback, i: Internship | f in isubmittedFeedbacks implies (some
        s: Student | i in sparticipates and
            (some a: Application | a in ssubmits and a in isubmissions))
}

// Fact to ensure that all internships where a student has participated have
// an interview scheduled for the application
fact allStudentInternshipsHaveInterviewInApplication{
    all s: Student, i: Internship | i in sparticipates implies
        (some a: Application | a in ssubmits and a in isubmissions and
            (some intv: Interview | intv in ainterview))
}

```

---

## Scenario1 (Predicate 0)

The following predicate describes a simple scenario with 1 university, 2 students, 2 internships, and 1 company. Only 1 student participates in an internship (the other one does not), both students submit different number of applications, there are no bounds on the number of interviews and feedbacks.

---

```

pred example0 {
    #University = 1
    #Student = 2
    #Internship = 2
    #Company = 1

    one s: Student | #s.submits = 1
    one s: Student | #s.submits = 2
    one s: Student | #s.participates = 1
    one s: Student | #s.participates = 0
}

run example0 for 4

```

---

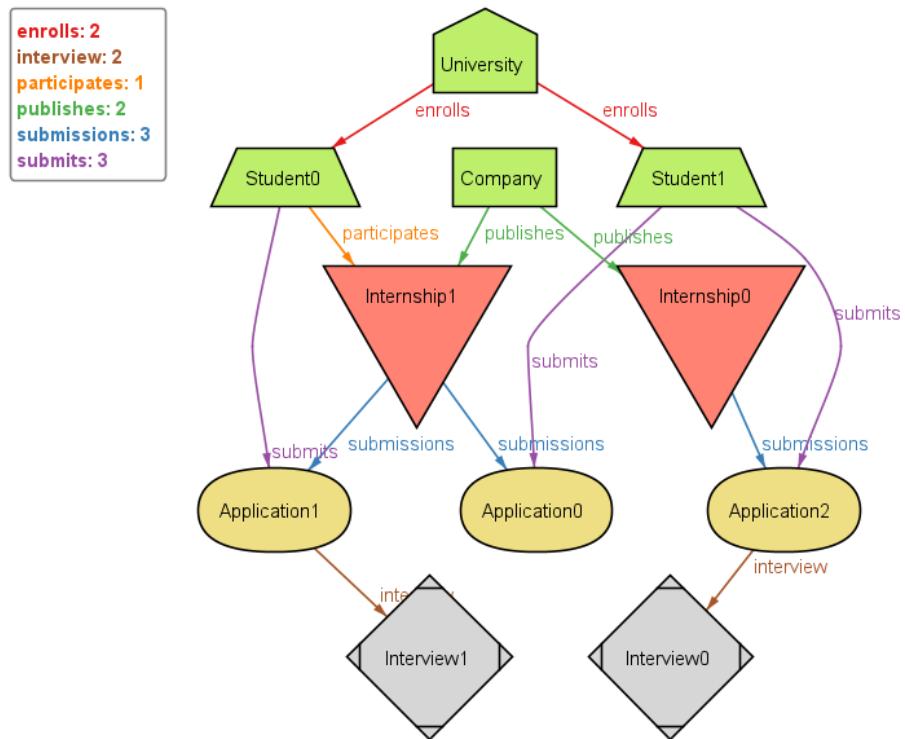


Figure 4.1: World generated by example 0

## Scenario2 (Predicate 1)

The following predicate describes a more structured model with 2 universities, 5 students, 3 internships, and 2 companies. All students submit more than 1 application, all students participate in less internships than the number of applications they submitted, at least 1 student participates in an internship, applications and feedbacks are more than 0, and the number of interviews is less than the number of applications.

---

```

pred example1 {
    #University = 2
    #Student = 5
    #Internship = 3
    #Company = 2
    some s: Student | #s.submits > 1
    all s: Student | #s.participates < #s.submits
    some s: Student | #s.participates > 0
    #Application > 0
    #Feedback > 0
    #Interview < #Application
}
run example1 for 9

```

---

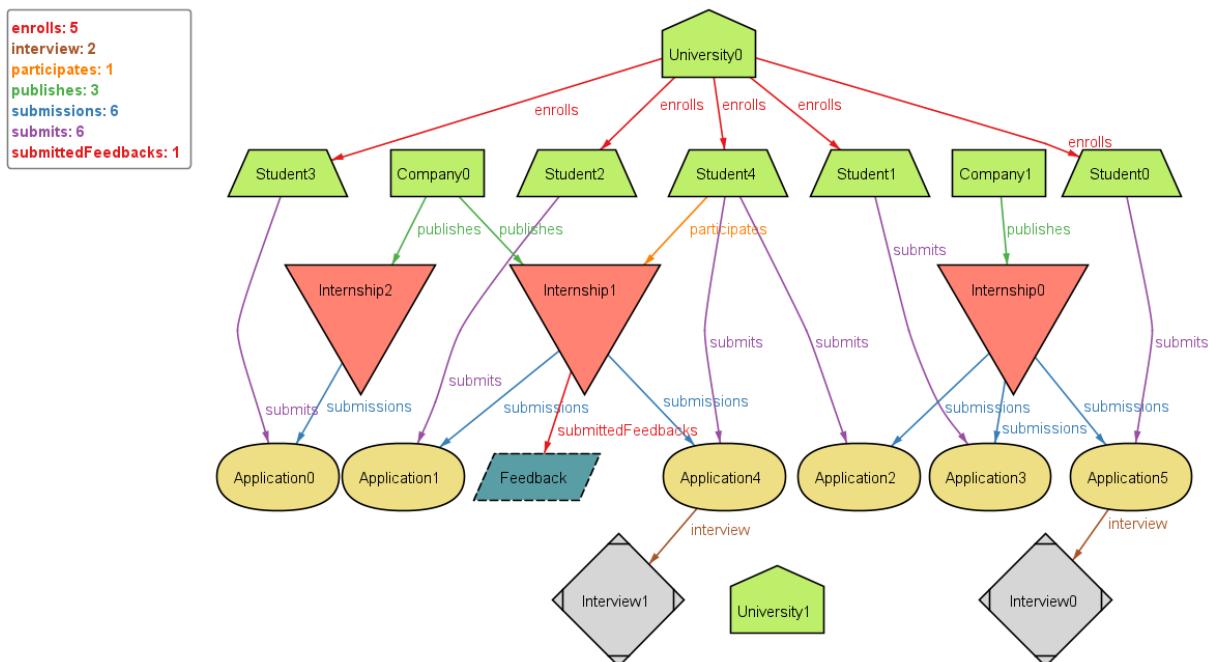


Figure 4.2: World generated by example 1

## Scenario3 (Predicate 2)

The following predicate describes a more complex model with 2 universities, 5 students, 5 internships, and 2 companies. All universities have less than 3 students enrolled, at least 1 student submits more than 1 application, at least 1 student participates in an internship, at least 1 internship has no submissions, all internships have less than 3 submitted feedbacks, and the number of submissions is greater than the number of students participating. Each company has less than 3 internships published.

---

```

pred example2 {
    #University = 2    #Student = 5      #Internship = 5 #Company = 2
    #Application > 0  #Feedback > 0     #Interview < #Application
    all u: University | #u.enrolls <= 3
    all u: University | (some s: Student | s in u.enrolls and #s.submits > 0
        and #s.participates > 0)
    some s: Student | #s.submits > 1
    all s: Student | #s.submits < 3
    some s: Student | #s.participates > 0
    some s: Student | #s.participates > 1
    some i: Internship | #i.submissions = 0
    all i: Internship | #i.submittedFeedbacks < 3
    all c: Company | #c.publishes <= 3
}
run example2 for 13

```

---

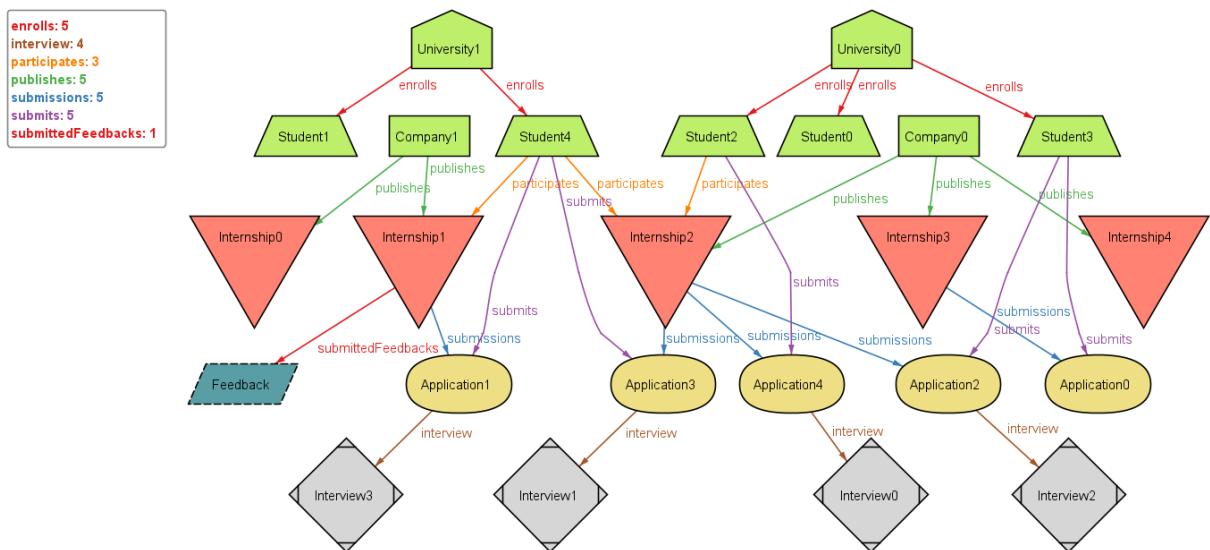


Figure 4.3: World generated by example 2

## B. Dynamic part

This model aims to describe the behavior of the system, focusing on the evolution of the system in time. In particular, we are going to describe the evolution of the system when a student submits an application, when a company publishes an internship, when an interview is scheduled, when a feedback is submitted, and when a student participates in an internship.

### Signatures

The following signatures describe the main entities of the system:

---

```

abstract sig User {}

sig Student extends User {
    var submits: set Application,
    var participates: set Internship
}

sig Company extends User {
    var publishes: set Internship
}

sig University extends User {
    var enrolls: set Student
}

var sig Internship {
    var submissions: set Application,
    var submittedFeedbacks: set Feedback,
    var internStatus: one InternshipStatus
}

var sig Application {
    var interview: lone Interview,
    var AppStatus: one ApplicationStatus
}

var sig Feedback {} {

```

```

    one i: Internship | this in i.submittedFeedbacks
}
var sig Interview {} {
    one a: Application | this in a.interview
}
//Model the status of an application
abstract sig ApplicationStatus{}
one sig SubmissionWaiting, SelectedToInterview, AcceptedToInternship,
AcceptedOffer, RejectedOffer, Rejected extends ApplicationStatus{}

//Model the status of internship:
abstract sig InternshipStatus{}
one sig InPublishing, InSelection, InProgress, Completed extends
InternshipStatus{}

```

---

## Facts

The following facts describe the constraints between the entities, relationships, and evolution of the system:

```

//Facts used before in Static model
// Fact to ensure that a student cannot submit multiple applications for the
// same internship
fact noRepeatedApplicationsForSameInternship{
    always (all s: Student, i: Internship | no disj a1,a2: s.submit | (a1 in
        i.submissions and a2 in i.submissions) )
}

// Fact to ensure that a feedback is submitted only if at least a student has
// participated in the internship
fact feedbackOnlyIfStudentInInternship{
    always( all f: Feedback, i: Internship | f in i.submittedFeedbacks
        implies (some s: Student | i in s.participates and
            (some a: Application | a in s.submit and a in i.submissions))
    )
}

// Fact to ensure that all internships where a student has participated have
// an interview scheduled for the application
fact allStudentInternshipsHaveInterviewInApplication{
    always( all s: Student, i: Internship | i in s.participates implies
        (some a: Application | a in s.submit and a in i.submissions and

```

```

        (some intv: Interview | intv in a.interview))
    )
}

//Facts for the dynamic model
// Application can only be in one status at a time
fact ApplicationStatusCanOnlyHaveOnePhaseAtATime{
    always all a: Application | (one a.AppStatus and
    (a.AppStatus = SubmissionWaiting or
    a.AppStatus = SelectedToInterview or
    a.AppStatus = AcceptedToInternship or
    a.AppStatus = AcceptedOffer or
    a.AppStatus = Rejected or
    a.AppStatus = RejectedOffer))
}

//Internship can only be in one possible status at a time
fact InternshipStatusCanOnlyHaveOnePhaseAtATime {
    always all i: Internship | one i.internStatus and
    (i.internStatus = InPublishing or
    i.internStatus = InSelection or
    i.internStatus = InProgress or
    i.internStatus = Completed)
}

// Fact to describe the possible statuses of the applications in each state of
// the internship
fact allowedAppStatusInEachInternshipStatus {
    always( all i: Internship |
        (i.internStatus = InPublishing implies
            (all a: i.submissions | a.AppStatus = SubmissionWaiting)) and
        (i.internStatus = InSelection implies
            (all a: i.submissions | a.AppStatus = SelectedToInterview or
            a.AppStatus = Rejected or a.AppStatus = AcceptedToInternship)) and
        (i.internStatus = InProgress implies
            (all a: i.submissions | a.AppStatus = AcceptedOffer or a.AppStatus
            = Rejected or a.AppStatus = RejectedOffer) and (some a:
            i.submissions | a.AppStatus = AcceptedOffer)) and
        (i.internStatus = Completed implies
            (all a: i.submissions | a.AppStatus = AcceptedOffer or a.AppStatus
            = Rejected or a.AppStatus = RejectedOffer)))
}
}

```

```

        = Rejected or a.AppStatus = RejectedOffer) and (some a:
        i.submissions | a.AppStatus = AcceptedOffer))
    )
}

// Variation of Interview in every state of the application
fact InterviewInEveryStateOfApplication{
    always all a: Application |
        (a.AppStatus = SubmissionWaiting implies a.interview = none) and
        (a.AppStatus = SelectedToInterview implies a.interview != none) and
        (a.AppStatus = AcceptedToInternship implies a.interview != none) and
        (a.AppStatus = AcceptedOffer implies a.interview != none) and
        (a.AppStatus = RejectedOffer implies a.interview != none) and
        ((a.AppStatus = Rejected and a.interview != none) implies after always
            a.interview != none) and
        ((a.AppStatus = Rejected and a.interview = none) implies after always
            a.interview = none)
    }
    // When an interview is scheduled for an application, it should never be
    // removed
fact InInterviewCanNotBeRemoved{
    always all a: Application | a.interview != none implies (a.interview' =
        a.interview)
}
// Student cannot change the university that he is enrolled in
fact studentUniversityInvariant{
    always (all s: Student, u:University | s in u.enrolls implies (after
        always s in u.enrolls))
}
// Internship cannot change the company that published it
fact internshipCompanyInvariant{
    always (all i: Internship, c:Company | i in c.publishes implies (after
        always i in c.publishes))
}
// Application cannot change the student that submitted it
fact applicationStudentInvariant{
    always (all a: Application, s:Student | a in s.submits implies (after
        always a in s.submits))
}
// Student cannot remove the internship that he has participated in

```

```

fact studentInternshipInvariant{
    always (all s: Student, i:Internship | i in s.participates implies (after
        always i in s.participates))
}

// Application cannot change the internship that it is submitted for
fact applicationInternshipInvariant{
    always (all a: Application, i:Internship | a in i.submissions implies
        (after always a in i.submissions))
}

// Feedback cannot change the internship that it is submitted for
fact feedbackInternshipInvariant{
    always (all f: Feedback, i:Internship | f in i.submittedFeedbacks implies
        (after always f in i.submittedFeedbacks))
}

// Fact to describe the possible status evolution of the internship
fact internshipStatusAllowedEvolution {
    always( no i: Internship |
        i.internStatus = InPublishing and (i.internStatus' = InProgress or
            i.internStatus' = Completed)
        or
        i.internStatus = InSelection and (i.internStatus' = InPublishing or
            i.internStatus' = Completed)
        or
        i.internStatus = InProgress and (i.internStatus' = InSelection or
            i.internStatus' = InPublishing)
        or
        i.internStatus = Completed and (i.internStatus' = InSelection or
            i.internStatus' = InProgress or i.internStatus' = InPublishing)
    )
}

// Fact to describe the possible status evolution of the application
fact allowedAppEvol {
    always (no a: Application |
        a.AppStatus = SubmissionWaiting and (a.AppStatus' = AcceptedOffer or
            a.AppStatus' = RejectedOffer or a.AppStatus' = AcceptedToInternship)
        or
        a.AppStatus = SelectedToInterview and (a.AppStatus' = AcceptedOffer or
            a.AppStatus' = RejectedOffer or a.AppStatus' = SubmissionWaiting)
        or
    )
}

```

```

a.AppStatus = AcceptedToInternship and (a.AppStatus' = Rejected or
    a.AppStatus' = SelectedToInterview or a.AppStatus' =
    SubmissionWaiting)
or
a.AppStatus = AcceptedOffer and (a.AppStatus' = Rejected or
    a.AppStatus' = RejectedOffer or a.AppStatus' = AcceptedToInternship
    or a.AppStatus' = SelectedToInterview or a.AppStatus' =
    SubmissionWaiting)
or
a.AppStatus = RejectedOffer and (a.AppStatus' = Rejected or
    a.AppStatus' = AcceptedOffer or a.AppStatus' = AcceptedToInternship
    or a.AppStatus' = SelectedToInterview or a.AppStatus' =
    SubmissionWaiting)
or
a.AppStatus = Rejected and (a.AppStatus' = AcceptedOffer or
    a.AppStatus' = RejectedOffer or a.AppStatus' = AcceptedToInternship
    or a.AppStatus' = SelectedToInterview or a.AppStatus' =
    SubmissionWaiting)
)
}

// An internship in publishing or selection cannot have students participating
// in it and feedbacks submitted
fact NoStudentsOrFeedbacksInInternshipInPublishingOrSelection{
    always all i: Internship | (i.internStatus = InPublishing or
        i.internStatus = InSelection) implies
        (no s: Student | i in s.participates) and (no f: Feedback | f in
            i.submittedFeedbacks)
}

// When internship is in publishing status, no applications submitted should
// have an interview scheduled
fact noInterviewsWhileInternshipInPublishing {
    always all i: Internship | i.internStatus = InPublishing implies (no a:
        i.submissions | a.interview != none)
}

// An application can be submitted to the internship only if the internship is
// in publishing status
fact SubmissionsMutableOnlyInPublishing {
    always all i: Internship | (i.internStatus != InPublishing or
        i.internStatus' != InPublishing) implies (i.submissions' =

```

```

    i.submissions)
}

// An internship can be in selection status only if at least one application
// has been submitted
fact InternshipInSelectionOnlyIfApplicationSubmitted{
    always all i: Internship | i.internStatus = InSelection implies (some a:
        Application | a in i.submissions)
}

// All applications of an internship in selection move from state
// SelectedToInterview to AcceptedToInternship (or Rejected) at the same step
fact NoSimultaneousSelectedAndAccepted {
    always all i: Internship | (i.internStatus = InSelection and some a:
        i.submissions | a.AppStatus = AcceptedToInternship) implies (all a:
        i.submissions | a.AppStatus = AcceptedToInternship or a.AppStatus =
        Rejected)
}

// If a student participates to an internship, then that internship is in
// progress or completed
fact NoStudentParticipatesInInternshipIfNotInProgressOrCompleted{
    always all s: Student, i: Internship | i in s.participates implies
        (i.internStatus = InProgress or i.internStatus = Completed)
}

// An internship can be in progress or completed only if at least one student
// participates in it
fact InternshipInProgressOrCompletedOnlyIfStudentParticipates{
    always all i: Internship | (i.internStatus = InProgress or i.internStatus =
        Completed) implies (some s: Student | i in s.participates)
}

// When an internship is in progress status or completed status, all
// applications should be in the final status: accepted offer, rejected offer
// or rejected
fact InternshipInProgressOrCompletedApplicationsInFinalStatus{
    always all i: Internship | (i.internStatus = InProgress or i.internStatus =
        Completed) implies
        (all a: i.submissions | (a.AppStatus = AcceptedOffer or a.AppStatus =
            RejectedOffer or a.AppStatus = Rejected))
}

// Every Student can only do one internship at a time
fact StudentCanDoOneInternshipsAtATime{

```

```

    always all s: Student | let concurrentParticipations = #(s.participates &
        {i: s.participates | i.internStatus = InProgress}) |
        concurrentParticipations <= 1
    }

    // Student application number should be more or equals to the number of
    // internships they participated in
    fact StudentApplicationsMoreThanParticipations{
        always all s: Student | #s.submits >= #s.participates
    }

    // Student should have the offer accepted equal or less than the number of
    // applications submitted and equal to the number of participations
    fact StudentOffersAccepted{
        always all s: Student |
        let numOffersAccepted = #(s.submits & {a: s.submits | a.AppStatus =
            AcceptedOffer}) |
        let numApplications = #s.submits |
        let numParticipations = #s.participates |
        numOffersAccepted <= numApplications and numOffersAccepted =
            numParticipations
    }

    // Student has participated if and only if he accepted the application related
    // to the internship
    fact StudentParticipatesInInternship{
        always all s: Student, i: Internship | i in s.participates implies
        (some a: Application | a in s.submits and a in i.submissions and
        a.AppStatus = AcceptedOffer)
    }

    // When an internship is in progress or completed, no more interviews should
    // be scheduled
    fact noChangeInInterviewsWhenInternshipInProgressOrCompleted{
        always all i: Internship | (i.internStatus = InProgress or i.internStatus =
            Completed) implies
            i.submissions.interview' = i.submissions.interview
    }

    // Once Internship is completed, no more feedbacks can be submitted
    fact InternshipCompletedNoMoreFeedbacks{
        always all i: Internship | i.internStatus' = Completed implies
            (i.submittedFeedbacks' = i.submittedFeedbacks)
    }
}

```

```

// Once Internship is in progress, feedbacks can be submitted
fact InProgressForFeedbacks{
    always all i: Internship | (#i.submittedFeedbacks' >
        #i.submittedFeedbacks) implies i.internStatus = InProgress
}

// Once Internship is completed, there should be at least 2 feedbacks
fact AtLeast2FeedbackIfCompleted{
    always all i: Internship | i.internStatus' = Completed implies
        (#i.submittedFeedbacks > 1)
}

//Initialization
// Forcing the initial state of the system
fact init {
    #University = 1
    #Company = 1
    #Student = 4
    #Internship = 1
    all i: Internship | i.internStatus = InPublishing
}

```

---

## Predicates and Scenario

In this section we are going to describe the evolution of the system in time, focusing on the main actions that can be performed by the users of the system.

```

// Predicates to model the creation of an internship
pred internshipCreation {
    eventually (#Internship = 2 and all i: Internship | i.internStatus =
        InPublishing and after always #Internship = 2) and
    always (all a: Application, i: Internship | (a in i.submissions and
        a.AppStatus = SubmissionWaiting) implies before i.internStatus =
        InPublishing)
}

// Predicates to model the submission of an application
pred applicationSubmission {
    eventually (some a: Application, s: Student, i: Internship | a in
        s.submits and a in i.submissions and a.AppStatus = SubmissionWaiting)
    and eventually #Application > 2
}
```

```

}

// Predicates to model the rejection of an application
pred applicationRejection {
    eventually (some s: Student, i: Internship, a: Application | a in
        s.submits and a in i.submissions and a.AppStatus = Rejected)
}

// Predicates to model the selection of an application
pred applicationSelection {
    eventually (some s: Student, i: Internship, a: Application | a in
        s.submits and a in i.submissions and a.AppStatus = SelectedToInterview)
}

// Predicates to model the acceptance of an application
pred applicationAcceptance {
    eventually (some s: Student, i: Internship, a: Application | a in
        s.submits and a in i.submissions and a.AppStatus = AcceptedToInternship)
}

// Predicates to model the acceptance of an internship
pred internshipAcceptance {
    eventually (some s: Student, i: Internship, a: Application | a in
        s.submits and a in i.submissions and a.AppStatus = AcceptedOffer)
}

// Predicates to model the rejection of an internship
pred internshipRejection {
    eventually (some s: Student, i: Internship, a: Application | a in
        s.submits and a in i.submissions and a.AppStatus = RejectedOffer)
}

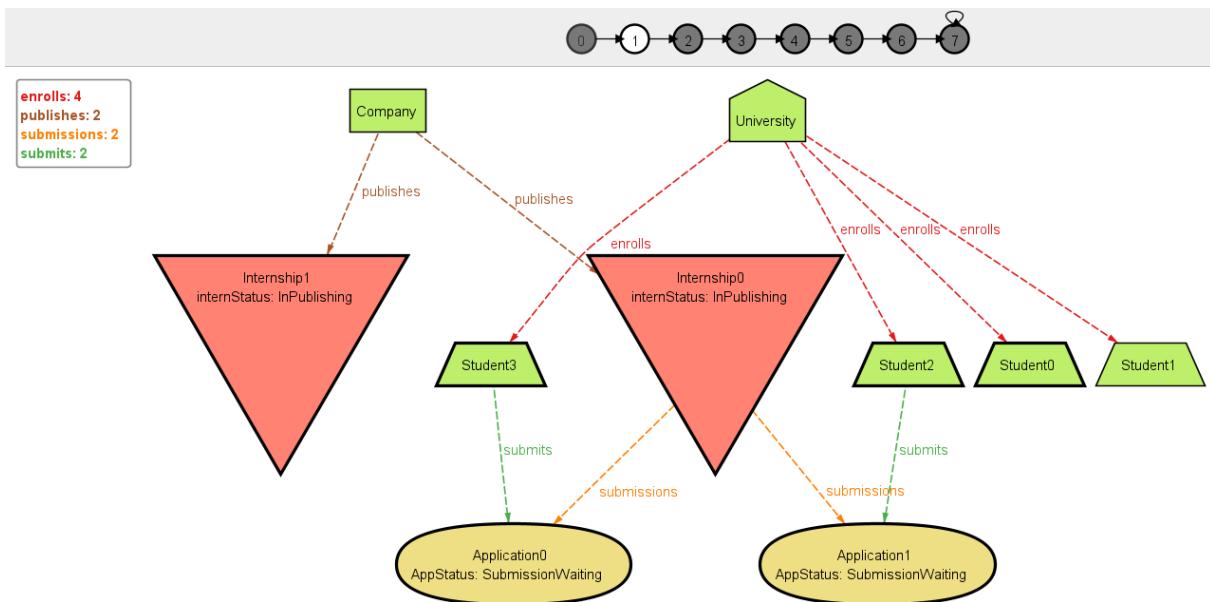
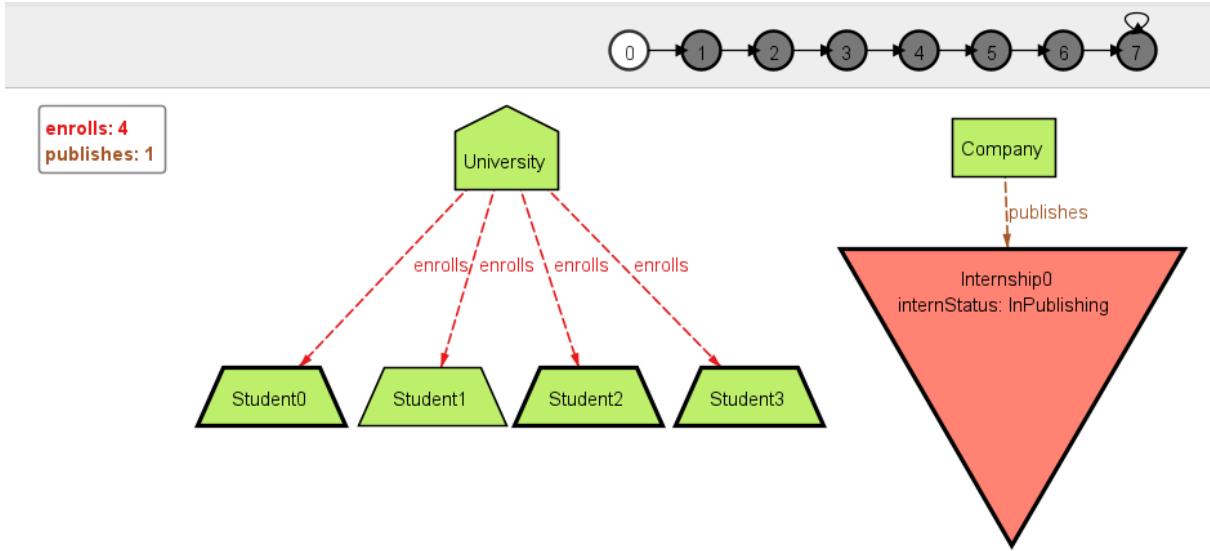
// Predicates to model the submission of a feedback
pred feedbackSubmission {
    eventually (some f: Feedback, i: Internship | f in i.submittedFeedbacks)
}

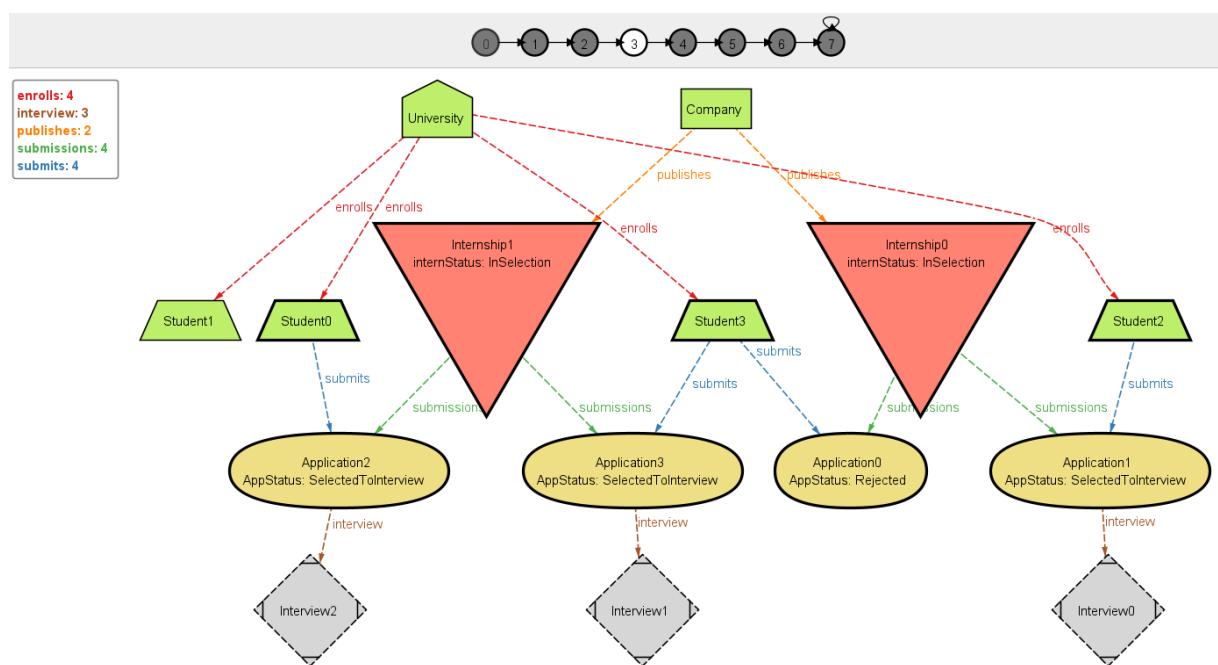
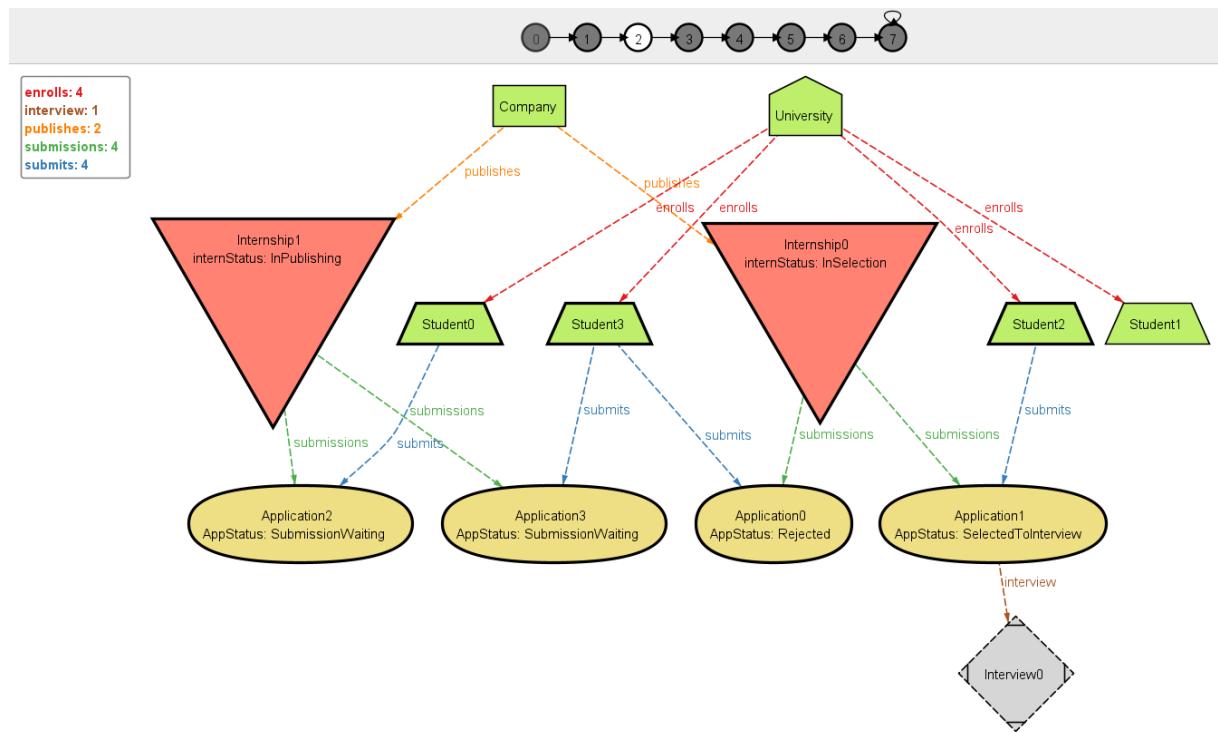
// Predicates to model the completion of all internships
pred allInternshipCompleted {
    eventually (all i: Internship | i.internStatus = Completed)
}

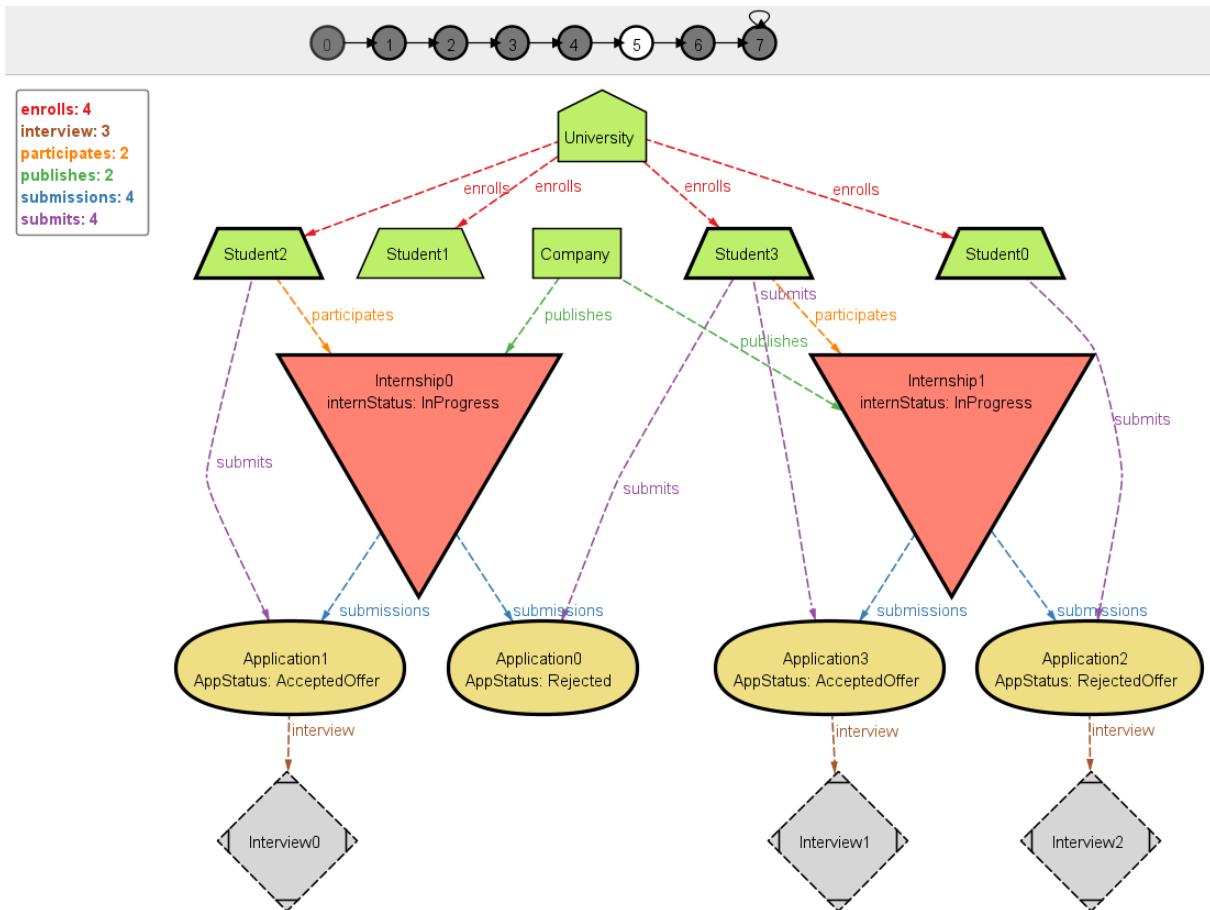
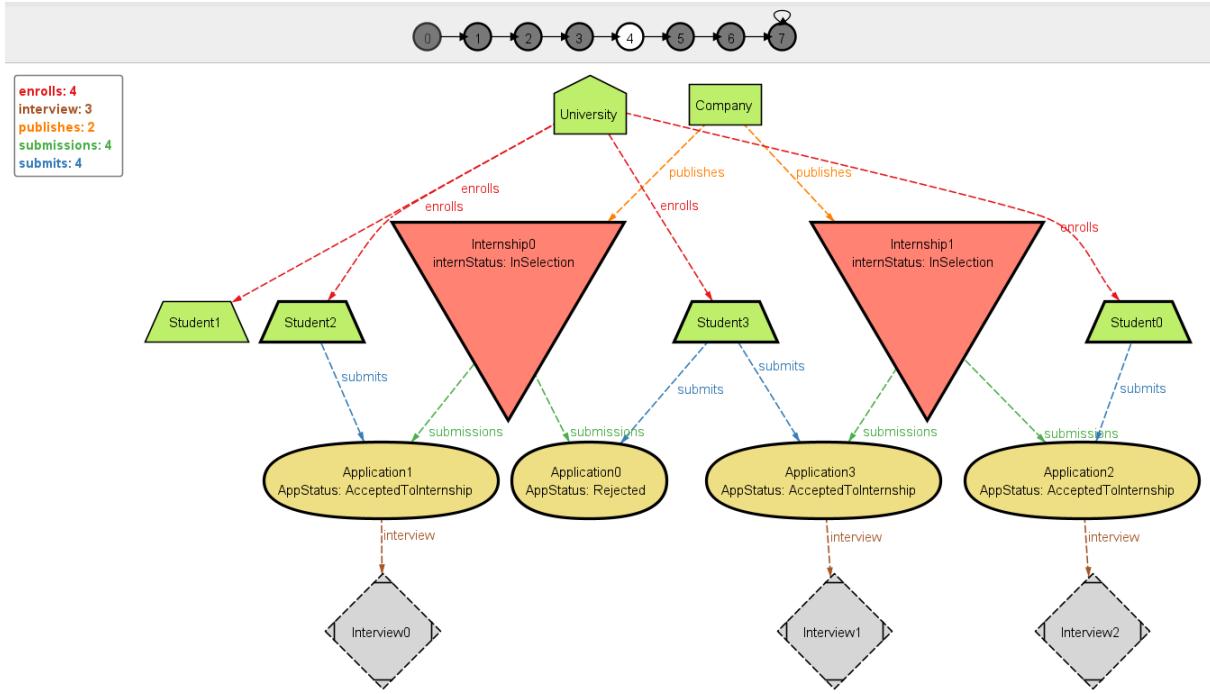
// RUN THE SYSTEM
run { internshipCreation;applicationSubmission;applicationRejection;
    applicationSelection;applicationAcceptance;internshipAcceptance;
    internshipRejection;feedbackSubmission;allInternshipCompleted
} for 7

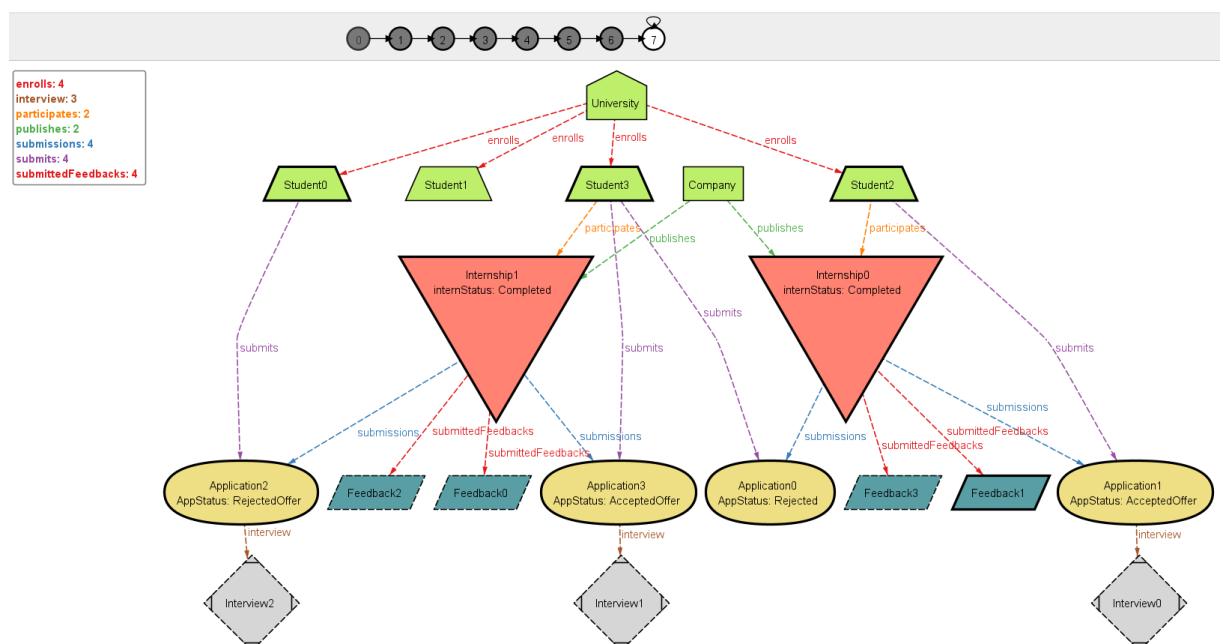
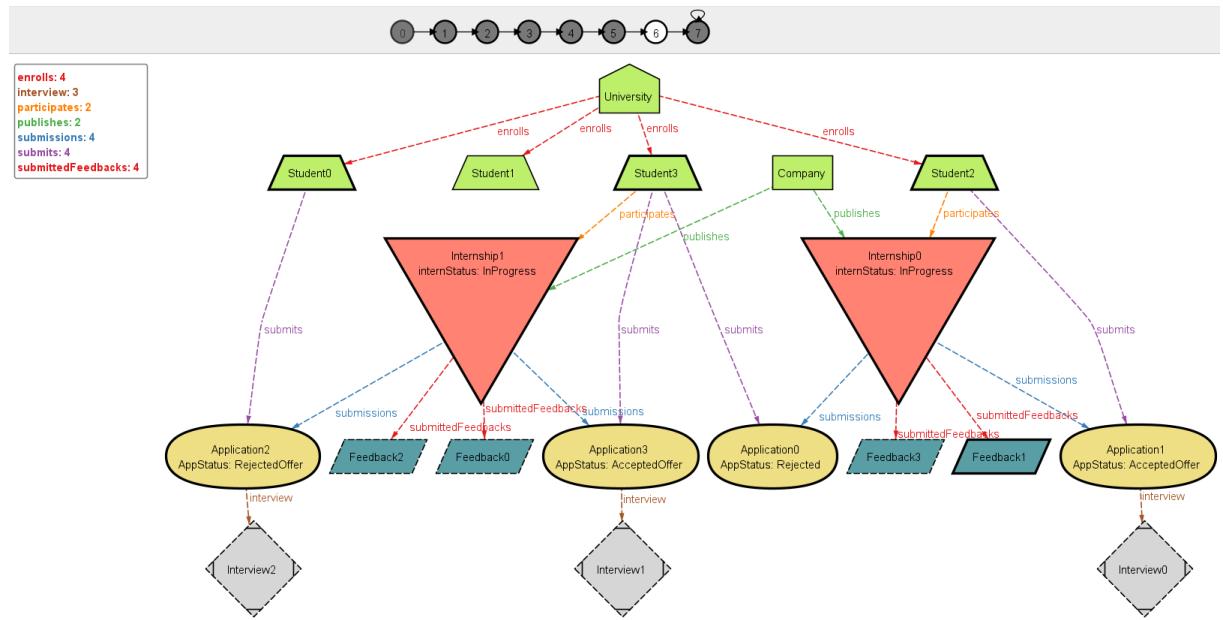
```

---











# 5 | Effort Spent

In following table we provide the tracking of the effort spent by each group member in the development of this document.

Section	Jie Chen	Riccardo Bonfanti
<b>1 – Introduction</b>	10 hours	6 hours
<b>2 – Overall Description</b>	20 hours	12.5 hours
<b>3 – Specific Requirements</b>	22.5 hours	21 hours
<b>4 – Formal Analysis Using Alloy Generic</b>	12 hours 5 hours	16 hours 11 hours
<b>Total</b>	69.5 hours	66.5 hours

Table 5.1: Effort spent for each section



# 6 | References

- The names of *SpongeBob SquarePants* characters referenced in this document are the intellectual property of *Nickelodeon and Viacom International Inc.* We do not claim any ownership of the copyrighted material. The use of these names is intended solely for purposes such as commentary, criticism, analysis, or education, and falls under the “fair use” provisions outlined in Section 107 of the Copyright Act of 1976 (Articolo 70 della Legge sul Diritto d’Autore italiana (Legge n. 633/1941)). This use is non-commercial and transformative in nature, with no intention of infringing upon the copyright holders’ rights.
- Lecture Slides of the course "Software Engineering 2", AA 2024/2025, by professor E. Di Nitto (Politecnico di Milano).
- We used Draw.io for the creation of the UML diagrams - <https://www.draw.io/>.
- We used GitHub for version control - <https://github.com/>.
- We used Visual Studio Code IDE for development of the LaTeX document - <https://www.visualstudio.com/>.
- We followed the Politecnico di Milano thesis template for the structure and style of the document - <https://www.overleaf.com/latex/templates/classical-format-thesis-scuola-di-ingegneria-industriale-e-dellinformazione-politecnico-di-milano/dkmvtndqkyxg>.