

# Students & Companies

Requirements Analysis and Specification Document

Francesco Ostidich, Matteo Salari, Francesco Rivitti

December 21, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.1.1	Goals . . . . .	2
1.2	Scope . . . . .	2
1.2.1	Features . . . . .	2
1.2.2	World phenomena . . . . .	3
1.2.3	Shared phenomena . . . . .	3
1.3	Definitions, acronyms, abbreviations . . . . .	4
1.3.1	Definitions . . . . .	4
1.3.2	Acronyms . . . . .	4
1.3.3	Abbreviations . . . . .	4
1.4	Revision history . . . . .	5
1.5	Reference documents . . . . .	5
1.6	Document structure . . . . .	5
<b>2</b>	<b>Overall description</b>	<b>6</b>
2.1	Product perspective . . . . .	6
2.1.1	Scenarios . . . . .	6
2.1.2	Class diagram . . . . .	7
2.1.3	State charts . . . . .	8
2.2	Product functions . . . . .	10
2.2.1	Key functions . . . . .	10
2.2.2	Requirements . . . . .	10
2.3	User characteristics . . . . .	11
2.4	Assumptions, dependencies and constraints . . . . .	11
2.4.1	Domain assumptions . . . . .	11
2.4.2	Dependencies . . . . .	12
<b>3</b>	<b>Specific requirements</b>	<b>13</b>
3.1	External interface requirements . . . . .	13
3.1.1	User Interfaces . . . . .	14
3.1.2	Hardware Interfaces . . . . .	15
3.1.3	Software Interfaces . . . . .	15
3.1.4	Communication Interfaces . . . . .	15
3.2	Functional requirements . . . . .	15

3.2.1	Use cases diagrams . . . . .	15
3.2.2	Use cases descriptions . . . . .	17
3.2.3	Use cases sequence diagrams . . . . .	25
3.2.4	Use cases mapping . . . . .	41
3.3	Performance requirements . . . . .	45
3.3.1	Specific requirements . . . . .	45
3.4	Design constraints . . . . .	46
3.4.1	Standards compliance . . . . .	46
3.4.2	Hardware limitations . . . . .	46
3.5	Software system attributes . . . . .	46
3.5.1	Availability . . . . .	46
3.5.2	Reliability . . . . .	46
3.5.3	Security . . . . .	47
3.5.4	Maintainability . . . . .	47
3.5.5	Portability . . . . .	47
<b>4</b>	<b>Formal analysis using Alloy</b>	<b>48</b>
4.1	Alloy Code . . . . .	48
4.2	Assertion Results . . . . .	58
4.3	World Generated . . . . .	59
<b>5</b>	<b>Effort spent</b>	<b>62</b>
<b>6</b>	<b>References</b>	<b>63</b>

# 1 Introduction

## 1.1 Purpose

Students&Companies (S&C) is a platform designed to connect university students with companies offering internships. It simplifies the internship searches of students and the projects advertisement for companies.

The platform employs recommendation mechanisms to match students and companies based on experience, skills, and project requirements. S&C also supports the selection process by managing interviews and collecting feedbacks. Additionally, it provides suggestions for improving CVs and project descriptions.

### 1.1.1 Goals

- G1 - Allow registered students to search and enroll for internship opportunities.
- G2 - Allow registered companies to advertise internship project opportunities.
- G3 - Allow registered universities to monitor their students ongoing internship and manage complaints.
- G4 - Support companies in the selection process by providing students with custom-made questionnaires.
- G5 - Ease matching by notifying students of relevant internships and companies for suitable candidates.
- G6 - Provide suggestions to both parties to refine their submissions.

## 1.2 Scope

### 1.2.1 Features

- F1 - **Recommendation mechanism:** students and companies are matched based on skills, experience and project requirements, exploiting keyword statistical methods.
- F2 - **Internship search:** students actively search and enroll for internships, while also being notified of opportunities that align with their profile.
- F3 - **Selection support:** companies are helped in the selection process by setting up interviews, creating custom questionnaires, and finalizing selections.
- F4 - **Suggestions system:** suggestions are provided to both parties in order to improve CVs and project descriptions, to enhance match potential.

**F5 - Complaint management:** universities can monitor internships of their students, handling complaints and addressing issues.

### 1.2.2 World phenomena

- WP1** - A user signs up in the system.
- WP2** - A user logs in the system.
- WP3** - A student looks for an internship to which enroll.
- WP4** - A company advertises an open internship project.
- WP5** - A student candidates himself for an internship.
- WP6** - A company asks a matched student to fill a questionnaire.
- WP7** - A company accepts a student for its internship project.
- WP8** - A student wants to visualize its ongoing internship information.
- WP9** - A company wants to visualize its ongoing internships information.
- WP10** - A university wants to visualize its student ongoing internship information.
- WP11** - A student or a company sends a complaint to the university.
- WP12** - A student or a company fills a feedback form when internship ends.

### 1.2.3 Shared phenomena

#### Controlled by the machine

- SP1** - The system shows the sign up page to a user.
- SP2** - The system shows the log in page to a user.
- SP3** - The system shows some internship advertisements to the student.
- SP4** - The system shows some eligible student information to the company.
- SP5** - The system notifies companies about students enrollment requests.
- SP6** - The system notifies students about companies candidature offers.
- SP7** - The system shows the company a panel in which to create a custom questionnaire.
- SP8** - The system shows the student the questionnaire to fill.
- SP9** - The system assigns the student to an internship project of a company.
- SP10** - The system shows the student its ongoing internship information.
- SP11** - The system shows the company its ongoing internships information.
- SP12** - The system notifies a student or a company with a suggestion.
- SP13** - The system notifies the university of a complaint.
- SP14** - The system ends the internship of a student.
- SP15** - The system shows a participant the feedback form to fill in.

#### Controlled by the world

- SP16** - A user fills the registration form to sign up.
- SP17** - A user fills the credentials fields to log in.
- SP18** - A student fills the form with its personal information and uploads its CV.
- SP19** - A company fills the form for advertising an internship project.
- SP20** - A student sends an enrollment request for an internship project.
- SP21** - A company sends an enrollment suggestion to a student.
- SP22** - A company creates a custom questionnaire.

- SP23 - A company sends a questionnaire to a student.
- SP24 - A student fills a questionnaire a company sent.
- SP25 - A company accepts a candidate student for its internship project.
- SP26 - A student or a company sends a complaint to the university.
- SP27 - A university ends the internship of its student.
- SP28 - A participant fills a feedback form.

## 1.3 Definitions, acronyms, abbreviations

### 1.3.1 Definitions

- **Internship project:** the description of the skills, technologies and roles the student will be working with during the internship, along with the set of tasks that will be covered
- **Internship advertisement:** the public post created by companies to promote available internships on the platform, aimed at attracting suitable candidates by highlighting its key aspects
- **Internship information:** general data about the (ongoing) internship, including the elapsed and remaining time, the compensation and the description of the project the student is working on
- **Enrollment request:** the submission of a student to indicate interest in a specific internship, initiating the selection process by formally applying
- **Enrollment suggestion:** the recommendation made by the platform to guide students in finding projects that best suit them
- **Custom questionnaire:** the tailored set of questions used by companies during interviews to assess a candidate fit for the internship
- **Candidate student:** a student who has applied for an internship and is currently under consideration by a company, moving forward in the selection process
- **Eligible student:** a student who meets the qualifications for an internship, making them viable candidates for recommendation and application
- **Suitable student:** a student who meets the qualifications for an internship, making them potential candidates to be recommended in the companies feed
- **Complaint:** a report submitted by a student or company to the university, regarding issues during the internship, such as unmet expectations, mistreatments, or procedural problems
- **Feedback form:** a structured form for students and companies used to provide feedback on their internship experience, enabling the platform to gather data for analysis, improvements, and recommendations

### 1.3.2 Acronyms

- **S&C:** Students&Companies

### 1.3.3 Abbreviations

- **Gn:** n-th goal
- **Fn:** n-th feature
- **WPn:** n-th world phenomena
- **SPn:** n-th shared phenomena

- **Sn**: n-th scenario
- **KFn**: n-th key function
- **Rn**: n-th requirement
- **Dn**: n-th domain assumption
- **UCn**: n-th use case

## 1.4 Revision history

- **Revised on**: December 21, 2024
- **Version**: 1.0
- **Description**: document initial release

## 1.5 Reference documents

- **Polimi Software Engineering 2 AY 2024/2025 assignment document**: goal, schedule and rules of the requirement engineering and design project
- **Polimi Software Engineering 2 AY 2024/2025 course slides**: the lecture slides provided during the course

## 1.6 Document structure

- **Chapter 1**: here is presented the problem statement and an outlining of the system objectives; in the scope subsection, insights into the various world and shared phenomena explain what the system addresses; here are also provided the essential resources for the readers, including definitions and abbreviations, to facilitate a comprehensive understanding of the document.
- **Chapter 2**: a comprehensive overview of the system is offered including insights into user profiles and their primary functions; the domain diagrams illustrate the system components and describe the various scenarios; the key domain assumptions are established, underpinning the system operations.
- **Chapter 3**: system requirements are delineated, encompassing both functional and non-functional aspects; follows the presentation of use case diagrams, illustrating the system interactions accompanied by their descriptions, with the related sequence diagrams; a clear mapping of the requirements is established, for a comprehensive understanding of both system goals and use cases.
- **Chapter 4**: here is given a formal analysis of the system with Alloy.
- **Chapter 5**: here is found an estimation of the effort spent by each group member.
- **Chapter 6**: here is provided a list of the references used in this document.

## 2 Overall description

### 2.1 Product perspective

The Students&Companies (S&C) platform serves as a matchmaking hub for university students seeking internships and companies offering them. It streamlines the process by matching student experiences, skills, and attitudes, with internships defined by specific projects, responsibilities, technologies, and benefits like mentorship and training. The platform is actively used by students to browse and apply for internships, while also supporting companies in showcasing available positions.

S&C includes an intelligent recommendation system to notify students of relevant opportunities and inform companies about suitable candidates, employing statistical analyses. Additionally, it facilitates the interview and selection process, offering tools for collecting candidate responses, in the form of custom questionnaires.

The platform gathers feedback from both students and companies, in order to better feed the recommendation system. Furthermore, it provides data-driven suggestions to enhance CVs and project descriptions.

Universities also engage with S&C to oversee internship progress, handle complaints, and, if necessary, address issues that may require intervention.

#### 2.1.1 Scenarios

- S1 - Signing up and logging in** - User Marco opens the platform and starts the sign-up procedure. He fills in the required information and completes the sign-up process. Then he logs out to make sure everything went smoothly, and a few seconds later, he logs in back again using the same credentials.
- S2 - Filling in personal information** - Maria logs into the platform and accesses the profile section. She fills in the personal data fields, such as contact details, degree program and competence skills. After completing the desired fields, she clicks the "Save" button, and the system confirms that her information has been successfully updated.
- S3 - Uploading the CV** - Luca decides to upload his CV, to be ready to apply for internship opportunities. He goes to the "Upload CV" section, selects the PDF file from his computer, and clicks "Upload". The system verifies the file validity and confirms the upload. Luca now has his CV associated with his profile, ready to be included with applications.
- S4 - Creating an internship project advertisement** - A consulting firm logs into the platform to publish a new internship advertisement. Anna, the recruitment officer, fills in the required fields in the creation form, including the project description, required skills, and internship



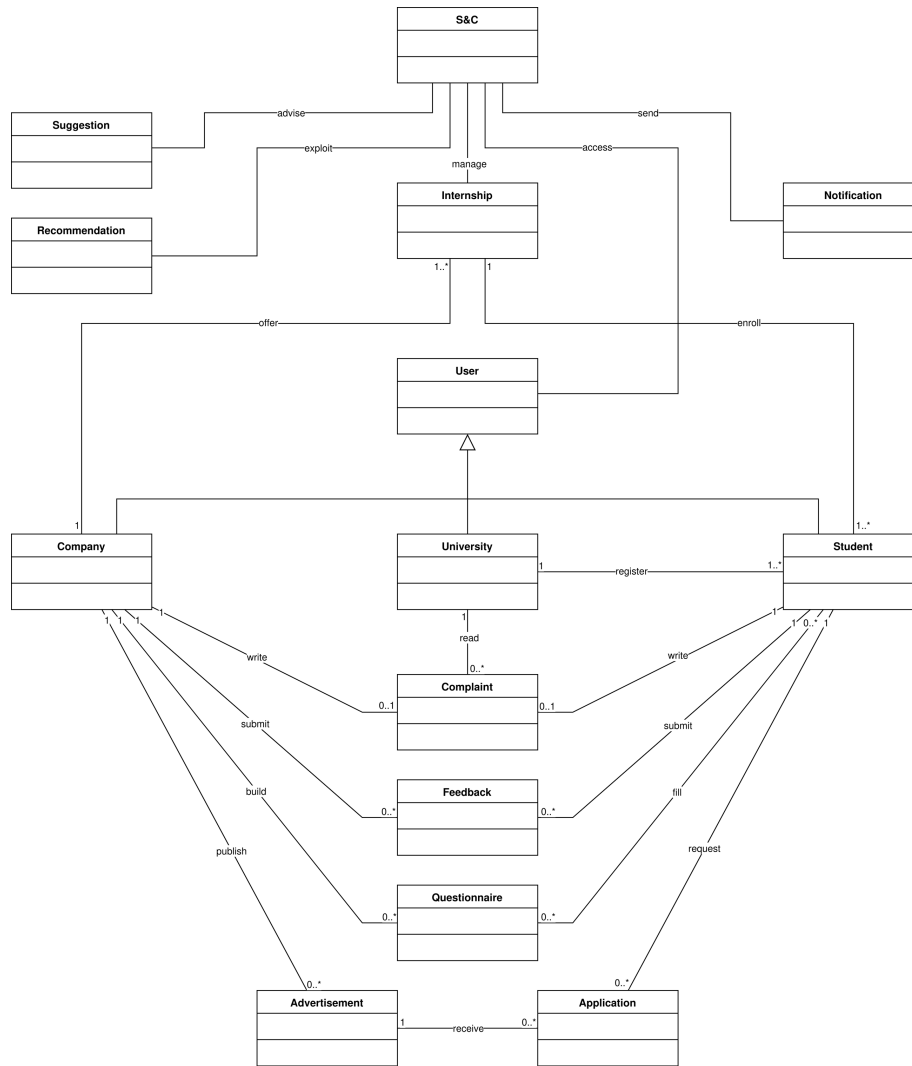
duration. After verifying the details, Anna selects “Publish”. The system confirms the publication, and the advertisement immediately becomes visible to students on the platform.

- S5 - Notifying the availability of an internship** - Sara has set her preferences to receive notifications about internship opportunities in marketing. When a new company posts a relevant advertisement, the system automatically pushes a notification in her notification section. By opening this section page, Sara can quickly access the advertisement and decide whether to apply.
- S6 - Selecting an internship project** - Alessandro is looking for an internship experience in computer engineering. He accesses the internship opportunities section and browses the available advertisements. When he finds an interesting offer at a startup, he views the details and decides to apply. The system confirms the submission of the application, and Alessandro receives a notification that the application was successfully sent.
- S7 - Creating a custom questionnaire** - A technology company decides to filter candidates with a custom questionnaire on programming skills. Laura, the person in charge, accesses the platform, selects the option to create a questionnaire, and adds specific questions about algorithms and programming languages. After completing the questionnaire, Laura associates it with the internship advertisement, ready to be sent and completed by applicants.
- S8 - Filling a questionnaire** - Martina is applying for an internship as a graphic designer at an advertising agency. During the application process, she is asked to complete a questionnaire evaluating her skills in graphic software like Photoshop and Illustrator. Martina answers all the questions and submits the questionnaire. The system confirms the submission, and Martina has completed her application.
- S9 - Starting a new internship** - Giovanni successfully completes the selection process with the company. Once he has been confirmed, the system notifies both parties and updates the status to “Internship Started”. Giovanni can now access the platform to monitor his progress during the internship.
- S10 - Viewing internship information** - During her internship, Laura accesses the “Internship Information” section to monitor project details, including goals, deadlines, and her supervisor name. The platform allows her to view updates and track her progress, ensuring she meets the requirements and expectations of the internship.
- S11 - Sending a complaint** - Paolo, a student on an internship, encounters issues with the support provided by the company. He decides to report the problem to the university through the platform. He accesses the complaints section, describes the issue, and submits the report. The system notifies the university administration, which reviews the case. After evaluating the situation, the university decides on the appropriate action, such as contacting the company for clarification or providing direct support to Paolo.
- S12 - Receiving feedback** - After completing the internship, both the student and the company provide feedback about their experience. The student evaluates the company regarding working conditions, support received, and the relevance of the assigned activities, while the company provides feedback on the skills and effort demonstrated by the student. These feedback reports are collected by the platform and made available to the university for an overall evaluation of the internship experience.

### 2.1.2 Class diagram

The UML class diagram below represents a high-level conceptual model of the software.

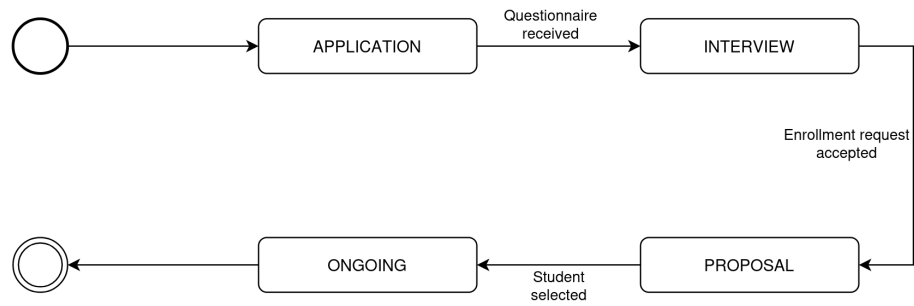
Due to the highly changing nature of the project, the UML may model objects that will not be implemented in the actual system. Moreover, at this level, any reference to methods and other low-level details will not be included, as detailed in the design phase.



### 2.1.3 State charts

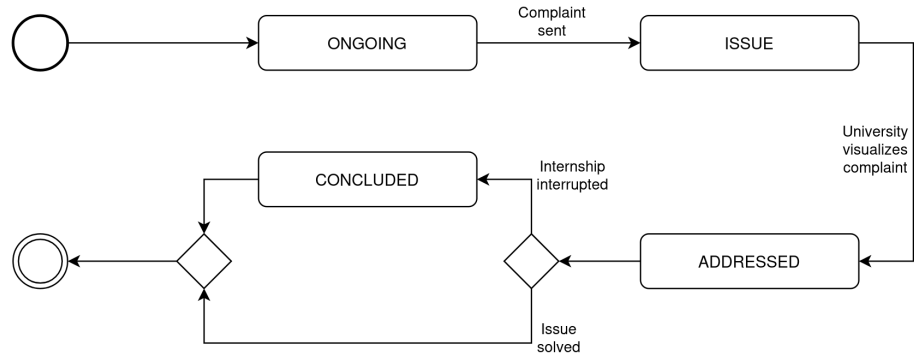
#### Selection process

The selection process comprehends all the actions that the student and the company take, from looking for advertisements to filling in and submitting the questionnaire.



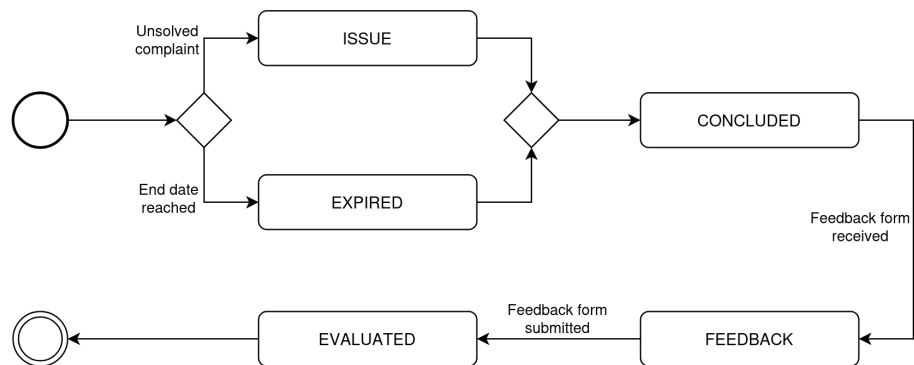
### Complaints addressing

Complaints are sent to the university by students and companies, and if the issue cannot be addressed and solved, the ongoing internship is interrupted.



### Feedback collection

After the internship is concluded, either because it expired or due to arising issues, the system prompts the participants to fill in and submit a feedback form, in order to collect information regarding the internship.



## 2.2 Product functions

### 2.2.1 Key functions

- KF1 - Internship project advertisements** - Companies can post detailed internship project descriptions, including tasks, required skills, technologies, and terms like compensation and benefits. This provides students clear insights about the available opportunities.
- KF2 - CV upload** - Students can upload and manage their CVs, listing skills, experiences, and interests. This helps the system match students with suitable internships and allows companies to view candidate profiles.
- KF3 - Interviews via custom questionnaires** - S&C supports the interview process by allowing companies to set up structured questionnaires, enabling them to gather specific information from candidates and assess their compatibility with the internship.
- KF4 - Complaints management** - The platform includes a complaints management feature where students and companies can report issues to the university, that can then address is accordingly, ensuring smooth and fair handling of internship-related problems.
- KF5 - Recommendation system** - S&C employs a recommendation system that connects students with internships based on CV content and project requirements, using keyword matching statistical analyses, to optimize matches.
- KF6 - Suggestion system** - The platform provides guidance for students and companies on improving their CVs and project descriptions, enhancing their appeal and increasing the likelihood of successful matches.

### 2.2.2 Requirements

- R1** - The system must allow an unregistered student to sign up.
- R2** - The system must allow an unregistered company to sign up.
- R3** - The system must allow an unregistered university to sign up.
- R4** - The system must allow a registered user to log in.
- R5** - The system must allow a registered user to fill in and edit its personal information.
- R6** - The system must allow a registered student to upload its CV.
- R7** - The system must allow a registered company to post an internship project.
- R8** - The system must allow a registered student to visualize a list of open internship projects.
- R9** - The system must allow a registered company to visualize a list of eligible students.
- R10** - The system must allow a registered student to make an enrollment request to an internship project.
- R11** - The system must allow a registered company to build custom made questionnaires.
- R12** - The system must allow a registered company to send questionnaires to students.
- R13** - The system must allow a registered student to fill in the questionnaire.
- R14** - The system must allow a registered company to accept students enrollment requests.
- R15** - The system must allow a registered student to see their ongoing internship information.
- R16** - The system must allow a registered company to see their ongoing internships information.
- R17** - The system must allow a registered university to see their students ongoing internship information.
- R18** - The system must allow a registered student to send complaints to the university.
- R19** - The system must allow a registered company to send complaints to the university.

- R20** - The system must allow a registered university to visualize complaints it received.
- R21** - The system must allow a registered university to end an ongoing internship of its student.
- R22** - The system must allow a registered student to fill in a feedback form when the internship ends.
- R23** - The system must allow a registered company to fill in a feedback form when the internship ends.
- R24** - The system must allow a registered student to visualize a list of suggested internships.
- R25** - The system must allow a registered company to visualize a list of suggested students.
- R26** - The system must allow a registered student to be notified about recommended internship.
- R27** - The system must allow a registered company to be notified about recommended students.

## 2.3 User characteristics

**Student** - Students actively seek internships to gain practical experience. They create profiles and upload CVs, detailing their skills, experiences, and interests. Students can proactively search for internships, receive recommendations for suitable roles, and engage in the selection process. They can also provide feedback on internships and report any issues through the platform.

**Company** - Organizations look for interns to hire for specific projects. They use the platform to post internship opportunities, specifying requirements like skills, tasks, and technologies. Companies receive recommendations for matching candidates, review CVs, conduct interviews via custom questionnaires, and can provide feedback on the internship experience. They can also provide feedback or report issues with interns.

**University** - Academic institutions oversee student internships to ensure they align with educational standards. Universities monitor internship statuses, handle critical complaints that may require intervention, providing support if internship issues arise. They play a supervisory role to ensure students benefit from safe, constructive, and academically relevant internship experiences.

## 2.4 Assumptions, dependencies and constraints

This section serves as a comprehensive overview of critical factors which must be considered during the implementation of the platform. It consolidates the foundational assumptions made during project planning and highlights eventual dependencies.

### 2.4.1 Domain assumptions

- D1** - The user must have a working internet connection.
- D2** - The user must have provided valid personal information.
- D3** - The student must be registered to a university.
- D4** - The university must have provided an organization mail to the student.
- D5** - The university must have been registered in the system directly by a staff member.

### **2.4.2 Dependencies**

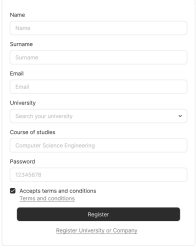
For the registration process, a verification email must be sent by the system to the user, to successfully sign up. This action requires the integration of an email service.

## **3 Specific requirements**

### **3.1 External interface requirements**

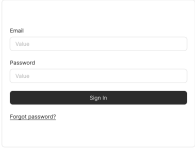
Users are able to utilize all the main functions that the system provides, making them able to access all the information they are granted permission for. Given such purposes, any device is suitable to make use of all the S&C functionalities, allowing convenient access through any web browser.

### 3.1.1 User Interfaces



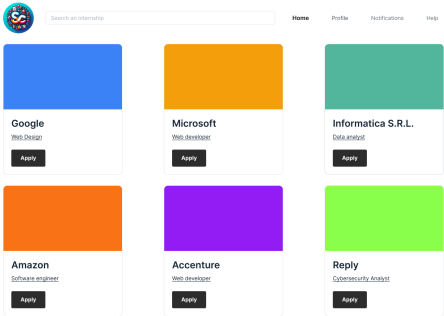
Form for signing up, including fields for Name, Surname, Email, University, Course of studies, Password, and a checkbox for accepting terms and conditions. A Register button is at the bottom.

(a) Sign up page



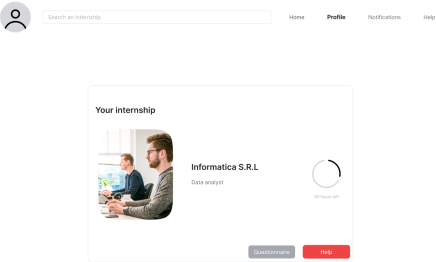
Form for logging in, including fields for Email and Password, a Sign in button, and a Forgot password? link.

(b) Log in page



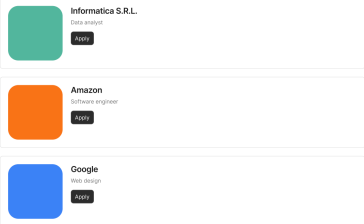
Home page layout showing a search bar, navigation links (Home, Profile, Notifications, Help), and a grid of company cards (Google, Microsoft, Informatica S.R.L., Amazon, Accenture, Reply) with Apply buttons.

(c) Home page



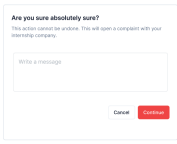
Profile page layout showing a user profile icon, search bar, navigation links, and a section titled "Your internship" with a company card for Informatica S.R.L. and buttons for Questions and Help.

(d) Profile page



Notifications page layout showing a search bar, navigation links, and a section titled "Companies that find you interesting" with a list of company cards (Informatica S.R.L., Amazon, Google) and Apply buttons.

(e) Notifications page



Complaint page form showing a confirmation message: "Are you sure absolutely sure? This action is irreversible. This will open a complaint with your internship company." and buttons for Cancel and Confirm.

(f) Complaint page



### 3.1.2 Hardware Interfaces

In order to be accessed, the platform requires a suitable device with a web browser.

### 3.1.3 Software Interfaces

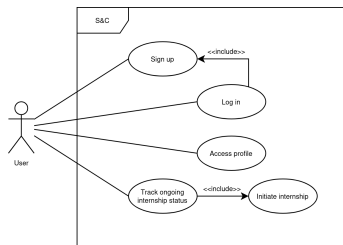
The system should integrate an email service, used for the registration process.

### 3.1.4 Communication Interfaces

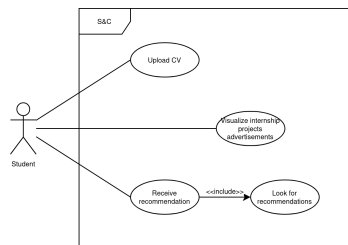
The system requires a stable internet connection to work properly. The connection is used to exchange data between users and the web server, which queries the requested information in a DB.

## 3.2 Functional requirements

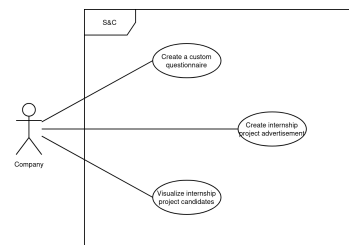
### 3.2.1 Use cases diagrams



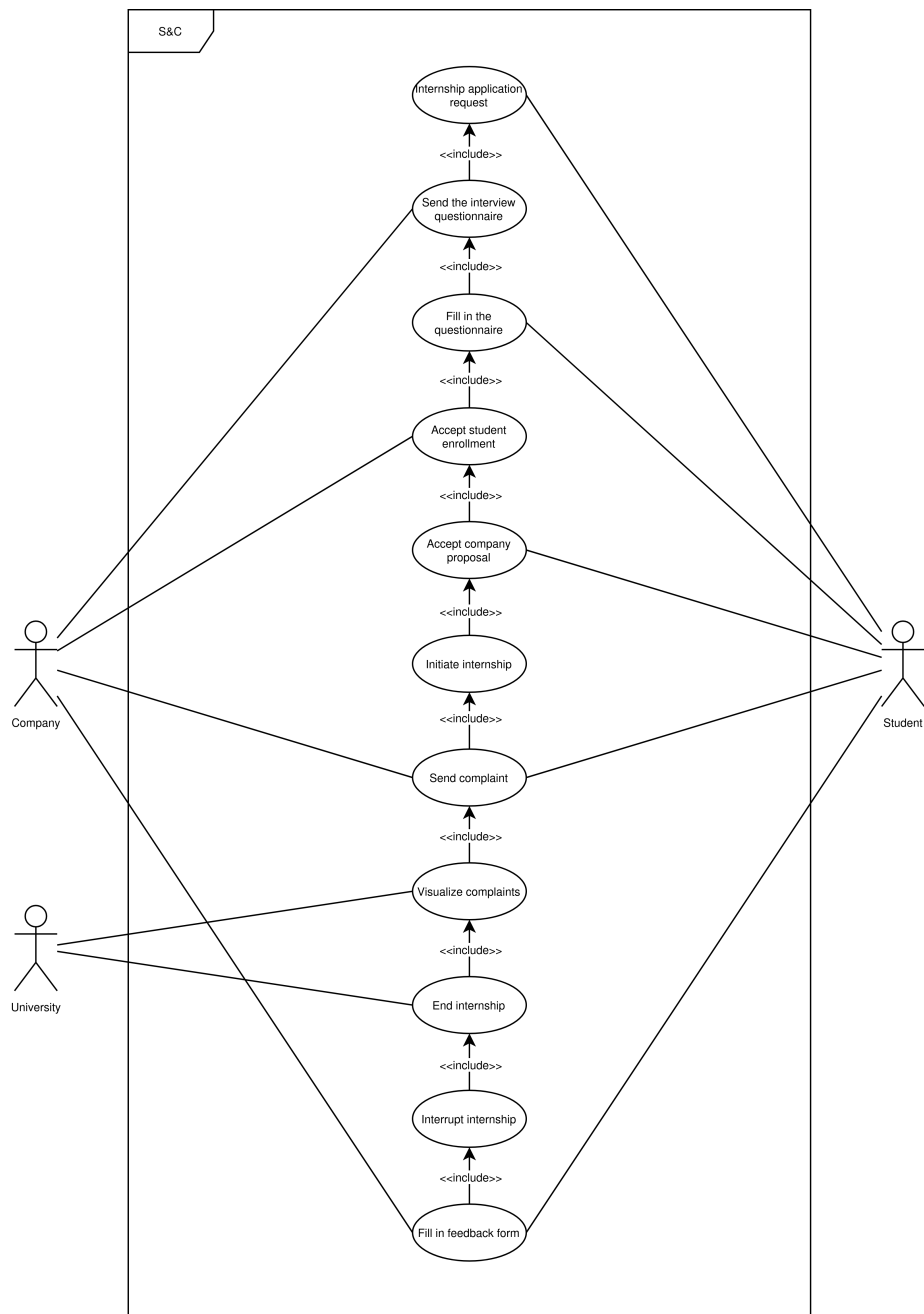
User common use cases



Student unique use cases



Company unique use cases



Internship iter use cases

### 3.2.2 Use cases descriptions

#### UC1 - StudentSignsUp

StudentSignsUp	
<b>Actor</b>	Student, Students&Companies, EmailService
<b>Entry condition</b>	The student is not already registered
<b>Event flow</b>	<ol style="list-style-type: none"><li>1 The student opens the sign up page</li><li>2 The student fills in the required information and clicks the sign up button</li><li>3 The platform checks that the information provided is valid</li><li>4 The platform sends an email through the email service to verify the student account</li><li>5 The student verifies the account via the link found in the email</li><li>6 The platform registers the new student account</li><li>7 The platform shows the student feed page</li></ol>
<b>Exit condition</b>	The student is registered
<b>Exceptions</b>	The student is already registered (3)

#### UC2 - CompanySignsUp

CompanySignsUp	
<b>Actor</b>	Company, Students&Companies, EmailService
<b>Entry condition</b>	The company is not already registered
<b>Event flow</b>	<ol style="list-style-type: none"><li>1 The company opens the sign up page</li><li>2 The company fills in the required information and clicks the sign up button</li><li>3 The platform checks that the information provided is valid</li><li>4 The platform sends an email through the email service to verify the company account</li><li>5 The company verifies the account via the link found in the email</li><li>6 The platform registers the new company account</li><li>7 The platform shows the company feed page</li></ol>
<b>Exit condition</b>	The company is registered
<b>Exceptions</b>	The company is already registered (3)

### UC3 - UserLogIn

UserLogIn	
Actor	Student, Company, University, Students&Companies
Entry condition	User is registered
Event flow	<ol style="list-style-type: none"><li>1 The user opens the log in page</li><li>2 The user enters its credentials and clicks the log in button</li><li>3 The platform checks that the account exists and the credentials are correct</li><li>4 The platform shows the home page</li></ol>
Exit condition	The user is logged in
Exceptions	The student is not registered (3)

### UC4 - StudentUploadsCV

StudentUploadsCV	
Actor	Student, Students&Companies
Entry condition	The student is logged in
Event flow	<ol style="list-style-type: none"><li>1 The student opens the profile page</li><li>2 The student clicks the upload CV button</li><li>3 The student chooses the CV file and lets it upload</li><li>4 The platform receives the file and stores it</li><li>5 The platform shows the student profile page</li></ol>
Exit condition	The student profile page shows the new CV
Exceptions	None

### UC5 - CompanyCreatesAdvertisement

CompanyCreatesAdvertisement	
Actor	Company, Students&Companies
Entry condition	The company is logged in
Event flow	<ol style="list-style-type: none"><li>1 The company opens the home page and clicks the create advertisement button</li><li>2 The company writes out the advertisement details and clicks the post button</li><li>3 The platform receives the advertisements details and stores it</li><li>4 The platform shows the company home page</li></ol>
Exit condition	The advertisements can be found in the students feed
Exceptions	None

**UC6 - StudentVisualizesAdvertisements**

<b>StudentVisualizesAdvertisements</b>	
<b>Actor</b>	Student, Students&Companies
<b>Entry condition</b>	The student is logged in
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The student opens the feed page</li> <li>2 The platform searches suitable advertisements to show</li> <li>3 The platform sends the advertisements list to the student</li> <li>4 The student feed page shows the received list</li> </ol>
<b>Exit condition</b>	The student visualizes interesting advertisements in the feed
<b>Exceptions</b>	None

**UC7 - CompanyVisualizesCandidates**

<b>CompanyVisualizesCandidates</b>	
<b>Actor</b>	Company, Students&Companies
<b>Entry condition</b>	The company is logged in
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The company opens the feed page</li> <li>2 The platform searches suitable students to show</li> <li>3 The platform sends the students list to the company</li> <li>4 The company feed page shows the received list</li> </ol>
<b>Exit condition</b>	The company visualizes potential students in the feed
<b>Exceptions</b>	None

**UC8 - StudentRequestsApplication**

<b>StudentRequestsApplication</b>	
<b>Actor</b>	Student, Company, Students&Companies
<b>Entry condition</b>	The student is logged in
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The student clicks an advertisement in the feed page</li> <li>2 The student clicks the apply button</li> <li>3 The platform receives the application request and stores it</li> <li>4 The platform notifies the company of the application request</li> </ol>
<b>Exit condition</b>	The company is notified of the application request
<b>Exceptions</b>	The student meanwhile accepted another internship (3)

#### UC9 - CompanyCreatesQuestionnaire

CompanyCreatesQuestionnaire	
<b>Actor</b>	Company, Students&Companies
<b>Entry condition</b>	The company is logged in
<b>Event flow</b>	<ol style="list-style-type: none"><li>1 The company opens the home page and clicks the create questionnaire button</li><li>2 The company builds the questionnaire form and clicks the save button</li><li>3 The platform receives the questionnaire form and stores it</li><li>4 The platform shows the company home page</li></ol>
<b>Exit condition</b>	The questionnaire can be sent to students who have applied
<b>Exceptions</b>	None

#### UC10 - StudentFillsQuestionnaire

StudentFillsQuestionnaire	
<b>Actor</b>	Student, Students&Companies
<b>Entry condition</b>	The student has received a questionnaire to fill
<b>Event flow</b>	<ol style="list-style-type: none"><li>1 The student opens the notifications page</li><li>2 The student opens the questionnaire that the company has sent</li><li>3 The student fills the questionnaire form and clicks the submit button</li><li>4 The platform sends the filled questionnaire to the company</li></ol>
<b>Exit condition</b>	The company receives the filled questionnaire
<b>Exceptions</b>	None

**UC11 - CompanyAcceptsStudentEnrollment**

<b>CompanyAcceptsStudentEnrollment</b>	
<b>Actor</b>	Student, Company, Students&Companies
<b>Entry condition</b>	The student has sent the questionnaire back to the company
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The company opens the page with the student questionnaire</li> <li>2 The company reviews the questionnaire and clicks the accept student button</li> <li>3 The platform notifies the student that it has been accepted</li> <li>4 The student opens the notifications page and clicks the accept button</li> <li>5 The platform checks that all information is valid</li> <li>6 The platform initiates the internship between student and company</li> <li>7 The platform shows the internship page to the student</li> <li>8 The platform notifies the company that the internship has started</li> </ol>
<b>Exit condition</b>	An internship is started between student and company
<b>Exceptions</b>	The student meanwhile accepted another internship (5)

**UC12 - StudentVisualizesInternshipInformation**

<b>StudentVisualizesInternshipInformation</b>	
<b>Actor</b>	Student, Students&Companies
<b>Entry condition</b>	Student is enrolled in an internship
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The student opens the profile page</li> <li>2 The student selects its internship project panel</li> <li>3 The platform shows the student internship information</li> </ol>
<b>Exit condition</b>	The student visualizes its ongoing internship information
<b>Exceptions</b>	None

**UC13 - CompanyVisualizesInternshipsInformation**

<b>CompanyVisualizesInternshipsInformation</b>	
<b>Actor</b>	Company, Students&Companies
<b>Entry condition</b>	Company has students enrolled in its internships
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The company opens the profile page</li> <li>2 The company selects one of its active internship projects</li> <li>3 The platform shows the enrolled students and the internship information</li> </ol>
<b>Exit condition</b>	The company visualizes its ongoing internship information
<b>Exceptions</b>	None

**UC14 - StudentSendsComplaint**

<b>StudentSendsComplaint</b>	
<b>Actor</b>	Student, University, Students&Companies
<b>Entry condition</b>	The student is currently enrolled in an internship
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The student opens the complaints page</li> <li>2 The student fills in the complaint text box and clicks the send button</li> <li>3 The platform notifies the university of the complaint</li> </ol>
<b>Exit condition</b>	The university is notified of the complaint
<b>Exceptions</b>	None

**UC15 - CompanySendsComplaint**

<b>CompanySendsComplaint</b>	
<b>Actor</b>	Company, University, Students&Companies
<b>Entry condition</b>	The company has a student currently enrolled in its internship
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The company opens the complaints page</li> <li>2 The company fills in the complaint text box and clicks the send button</li> <li>3 The platform notifies the university of the complaint</li> </ol>
<b>Exit condition</b>	The university is notified of the complaint
<b>Exceptions</b>	None



**UC16 - UniversityVisualizesComplaints**

<b>UniversityVisualizesComplaints</b>	
<b>Actor</b>	University, Students&Companies
<b>Entry condition</b>	A complaint has been sent by a student or a company
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The university opens the notifications/complaint page</li> <li>2 The university selects one complaint notification</li> <li>3 The university visualizes the full complaint message and information</li> </ol>
<b>Exit condition</b>	The university visualizes the complaint
<b>Exceptions</b>	None

**UC17 - UniversityEndsInternship**

<b>UniversityEndsInternship</b>	
<b>Actor</b>	Student, Company, University, Students&Companies
<b>Entry condition</b>	Student and company are currently linked in an internship
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The university opens the ongoing internships page and clicks the interruption button</li> <li>2 The platform registers the internship as concluded</li> <li>3 The company is notified of the internship conclusion</li> <li>4 The student is notified of the internship conclusion</li> </ol>
<b>Exit condition</b>	The internship is registered as concluded
<b>Exceptions</b>	None

**UC18 - InternshipExpires**

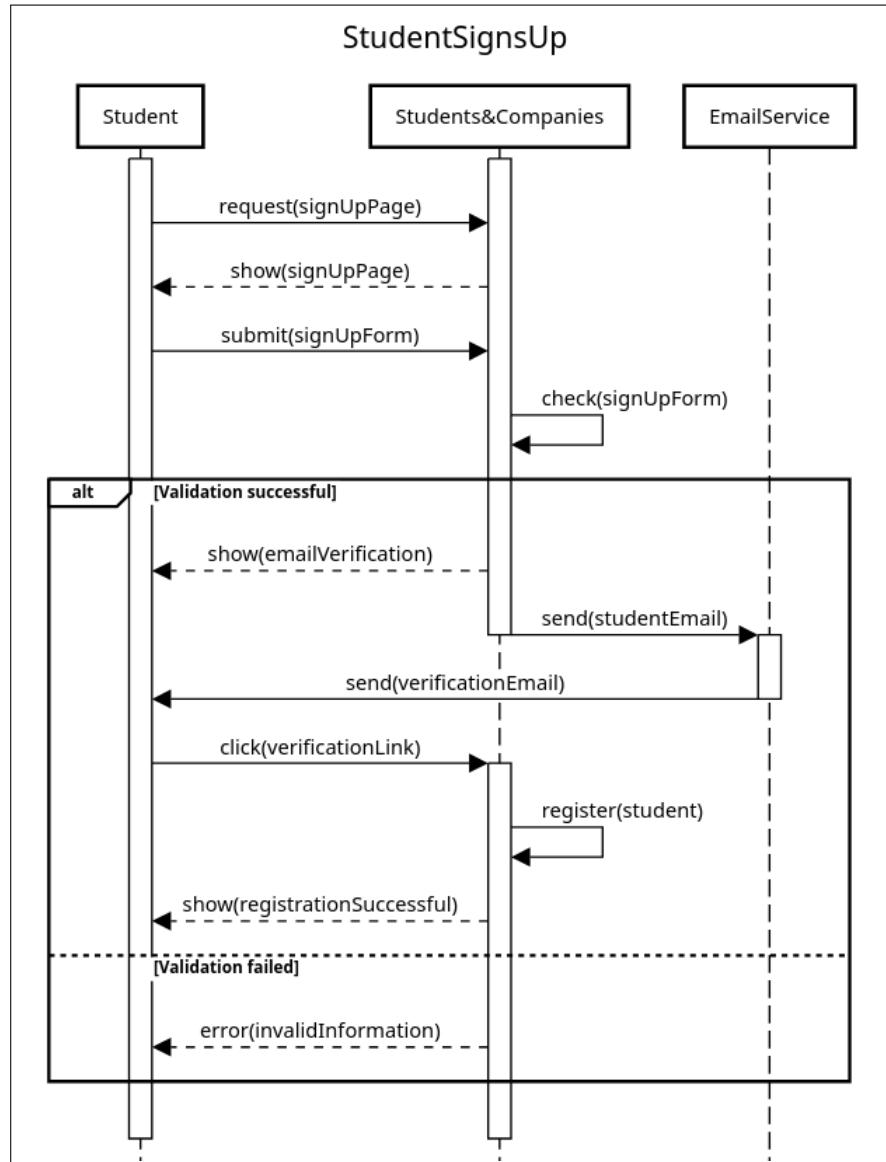
<b>InternshipExpires</b>	
<b>Actor</b>	Student, Company, Students&Companies
<b>Entry condition</b>	An ongoing internship reaches its conclusion date
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1 The platform sees that the internship has reached its conclusion date</li> <li>2 The platform registers the internship as concluded</li> <li>3 The company is notified of the internship conclusion</li> <li>4 The student is notified of the internship conclusion</li> </ol>
<b>Exit condition</b>	The internship is registered as concluded
<b>Exceptions</b>	None

#### UC19 - ParticipantFillsFeedbackForm

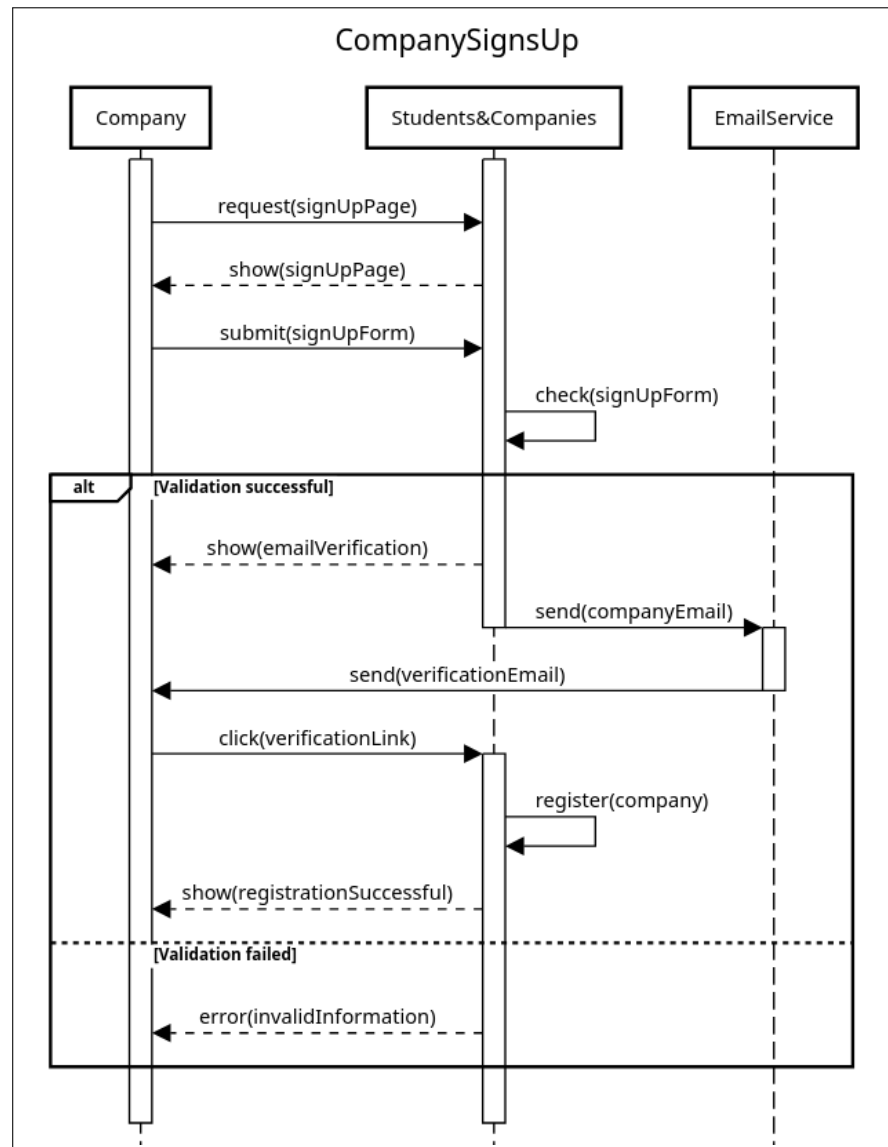
StudentFillsFeedbackForm	
<b>Actor</b>	Student, Company, Students&Companies
<b>Entry condition</b>	The internship has ended
<b>Event flow</b>	<ol style="list-style-type: none"><li>1 The platform notifies the student/company that a feedback form can be filled in</li><li>2 The student/company opens the notifications page</li><li>3 The student/company opens the feedback form page</li><li>4 The student/company fills in the form and submits it</li><li>5 The platform receives the feedback form and stores it</li></ol>
<b>Exit condition</b>	The platform receives feedback data about the internship
<b>Exceptions</b>	None

### 3.2.3 Use cases sequence diagrams

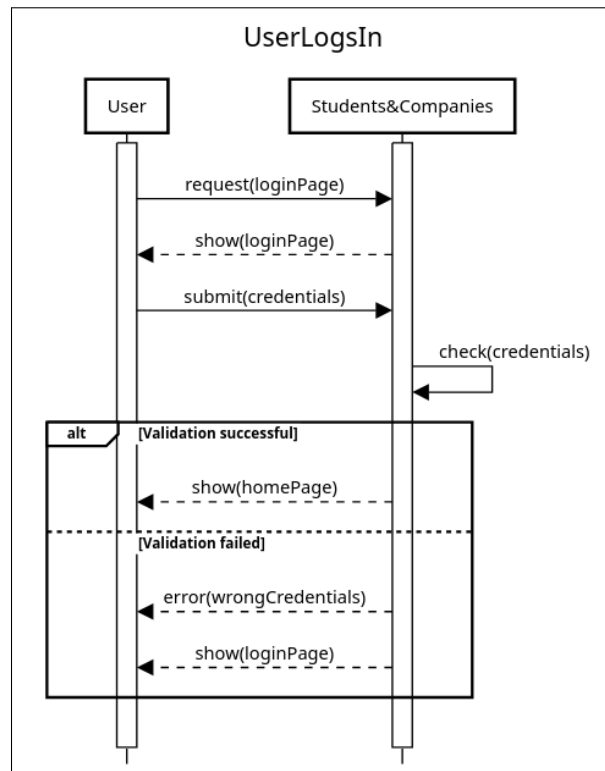
#### UC1 - StudentSignsUp



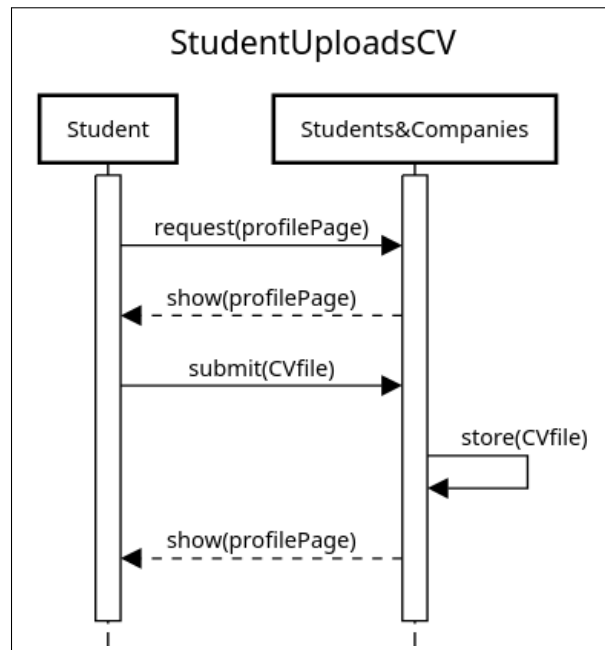
## UC2 - CompanySignsUp



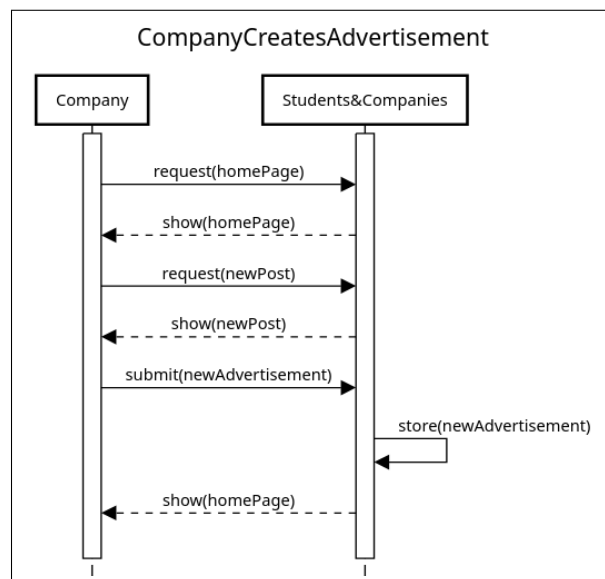
### UC3 - UserLogsIn



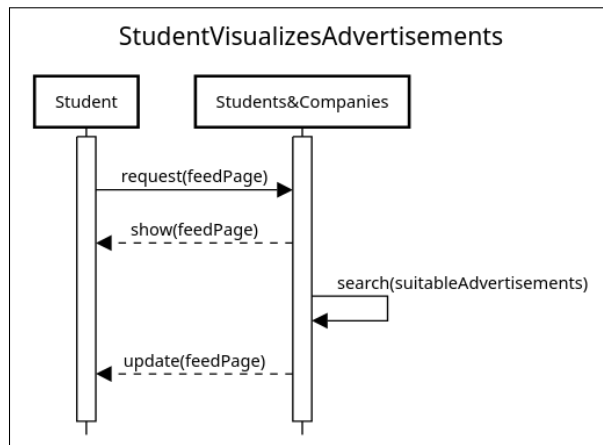
#### UC4 - StudentUploadsCV



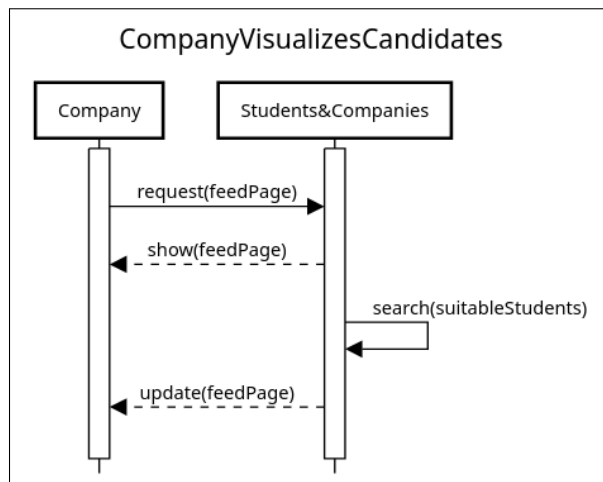
#### UC5 - CompanyCreatesAdvertisement



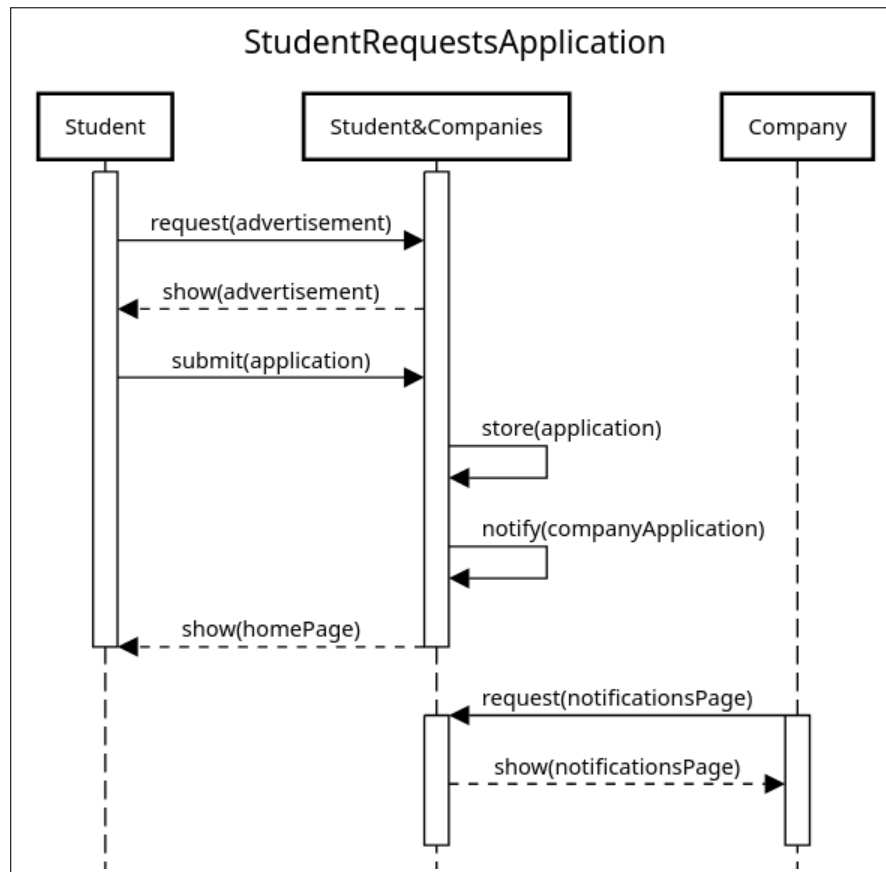
## UC6 - StudentVisualizesAdvertisements



## UC7 - CompanyVisualizesCandidates

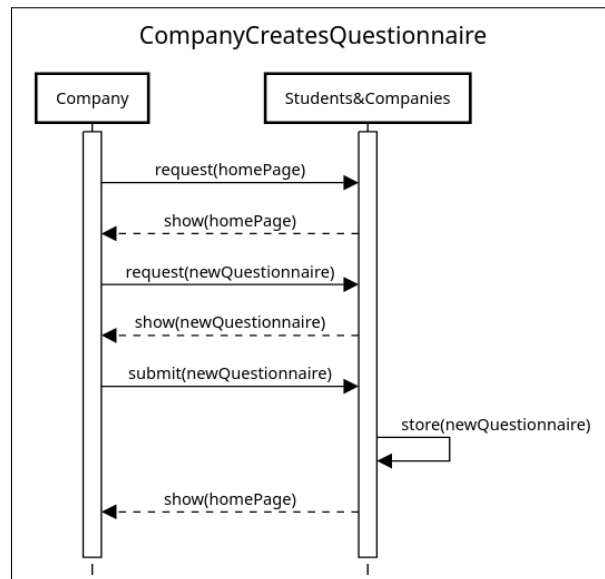


## UC8 - StudentRequestsApplication

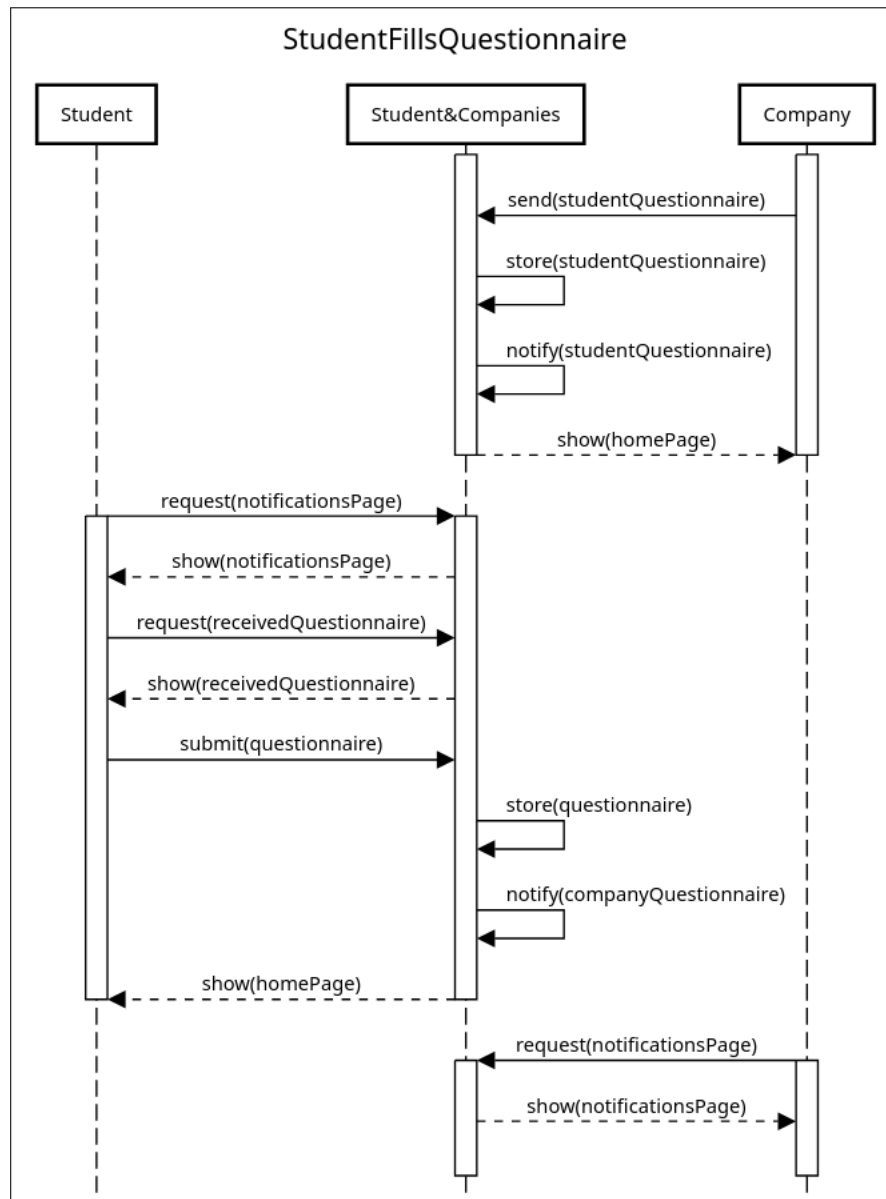




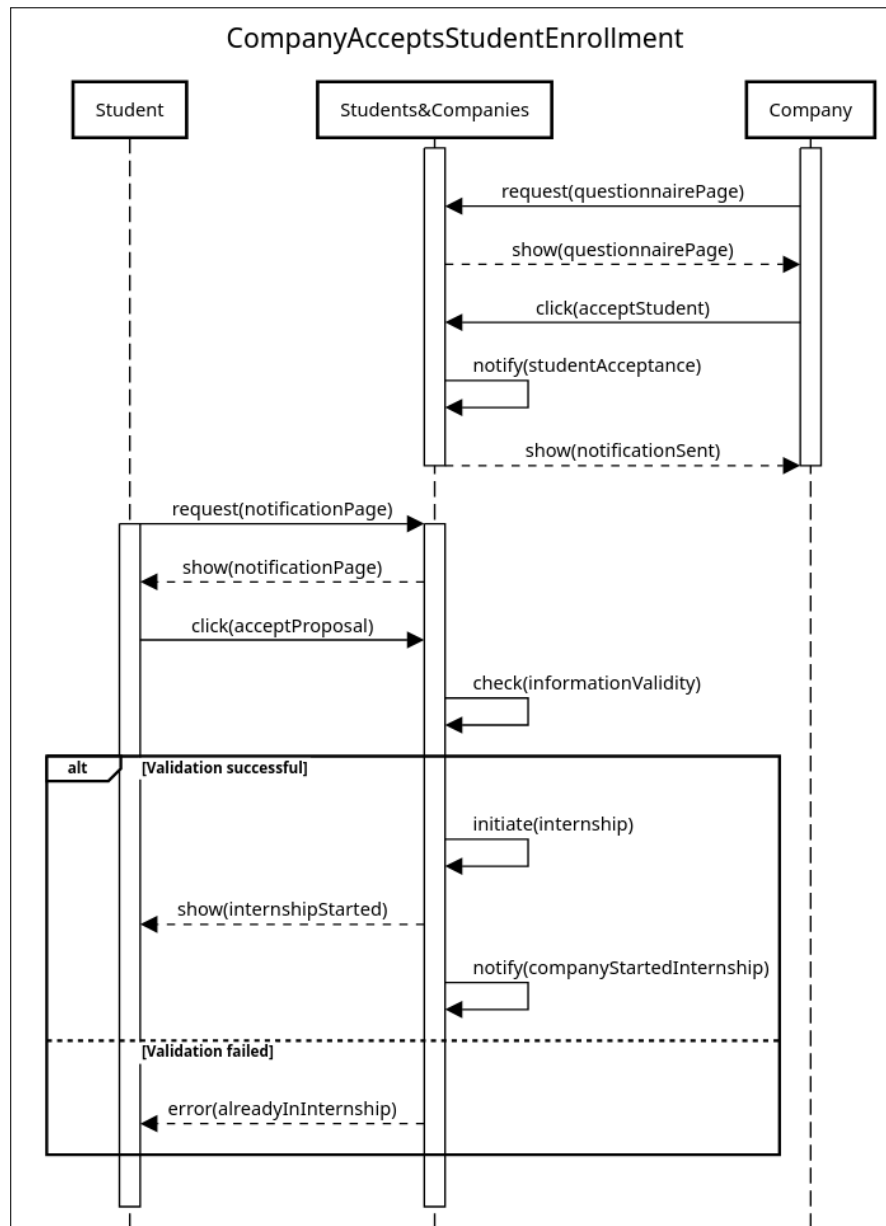
## UC9 - CompanyCreatesQuestionnaire



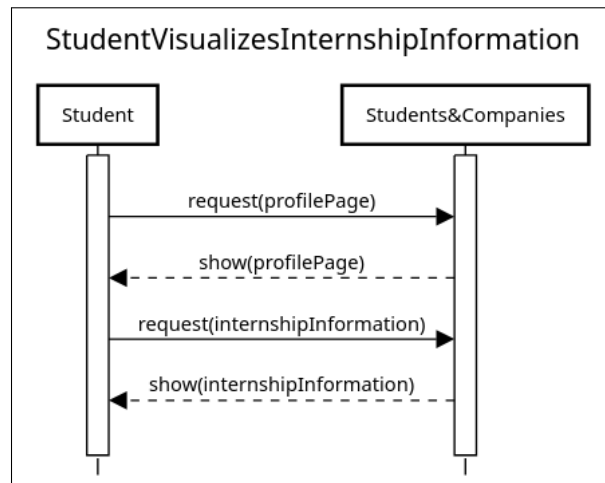
## UC10 - StudentFillsQuestionnaire



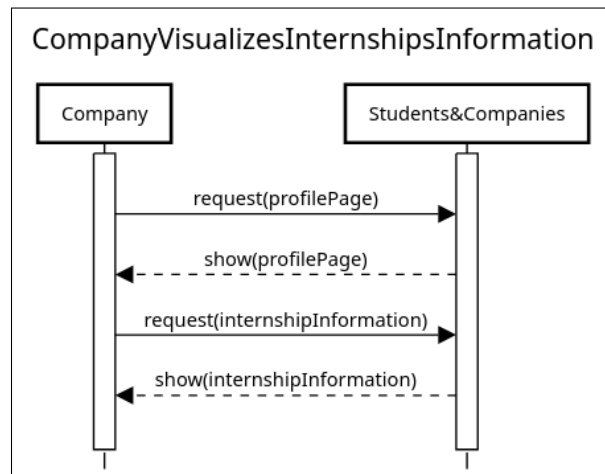
## UC11 - CompanyAcceptsStudentEnrollment



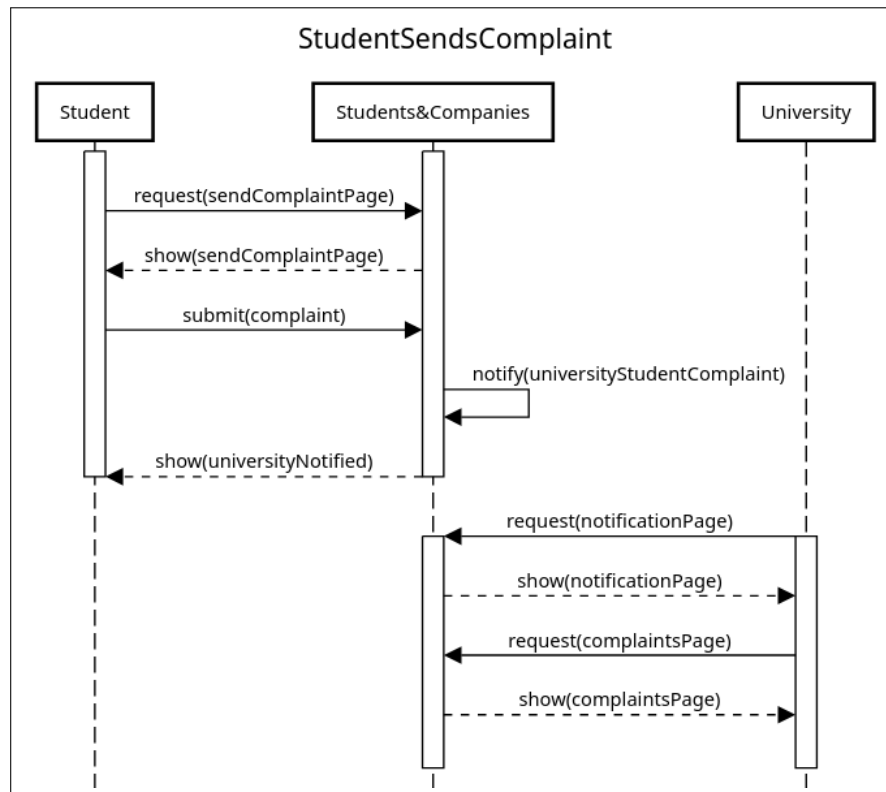
### UC12 - StudentVisualizesInternshipInformation



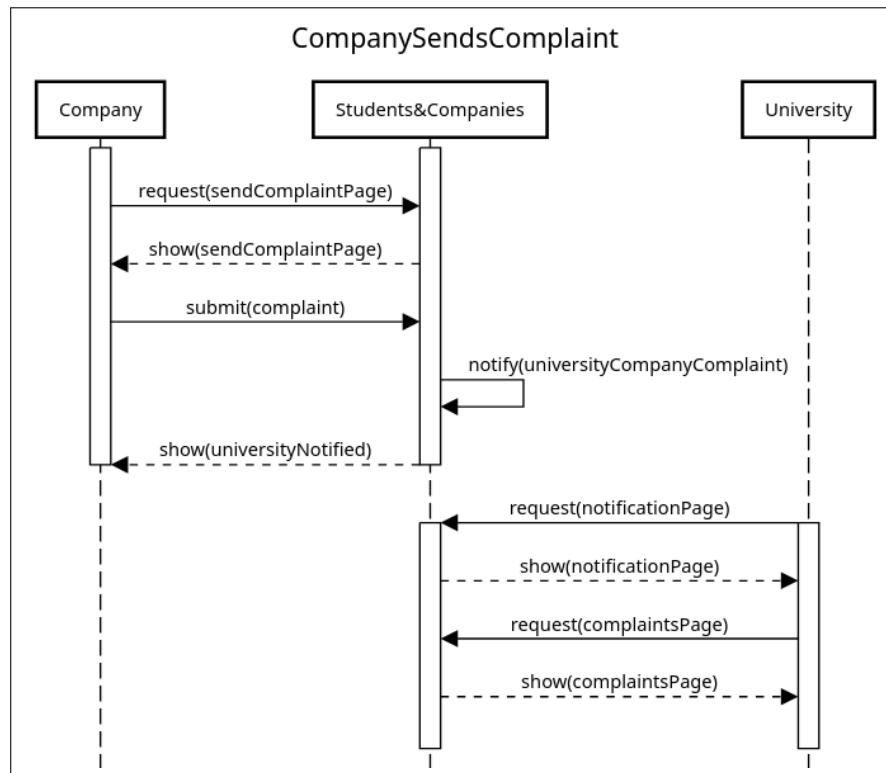
### UC13 - CompanyVisualizesInternshipsInformation



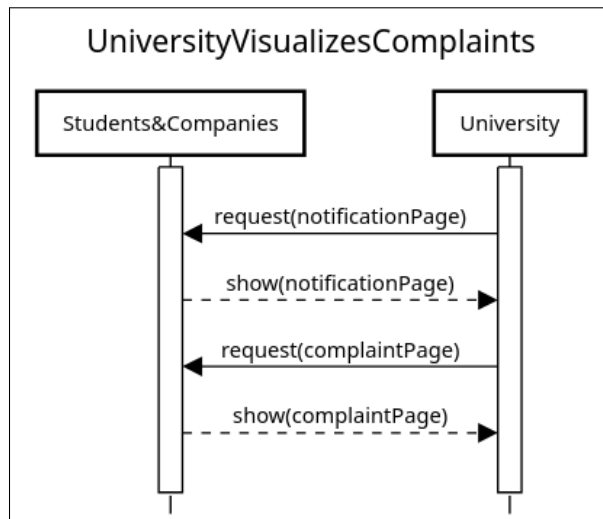
## UC14 - StudentSendsComplaint



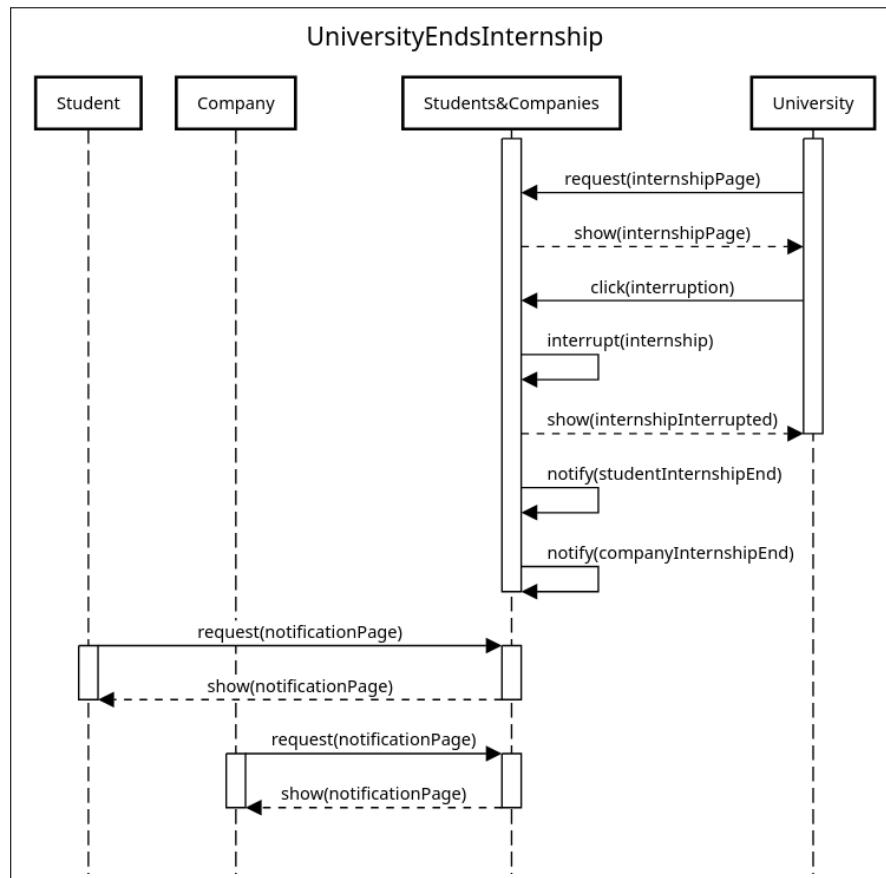
## UC15 - CompanySendsComplaint



## UC16 - UniversityVisualizesComplaints

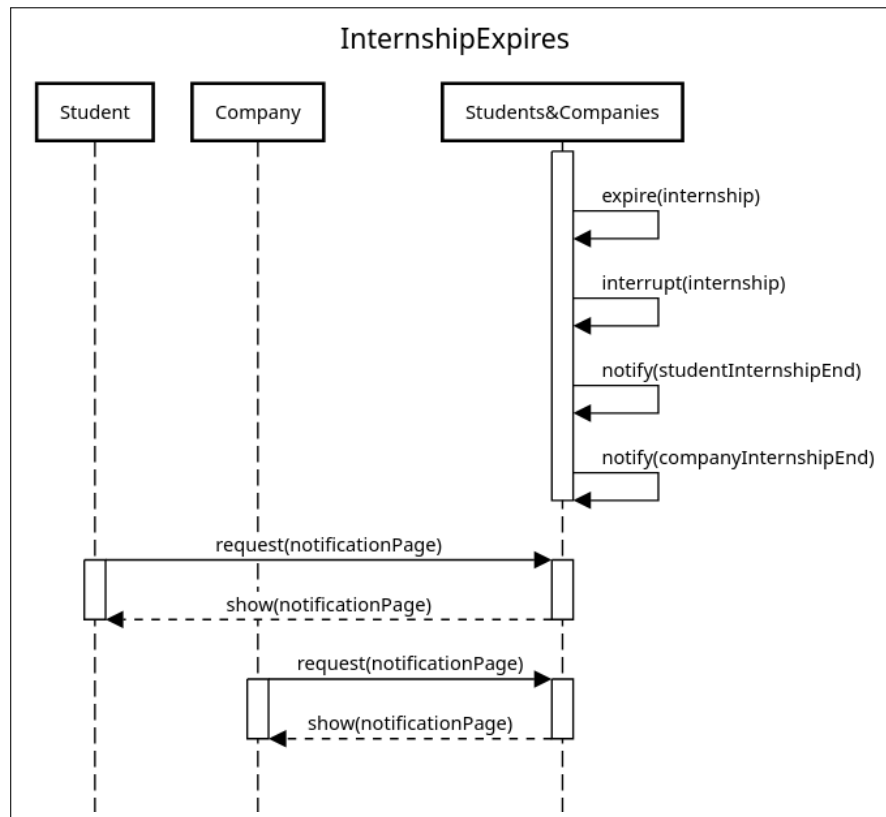


## UC17 - UniversityEndsInternship

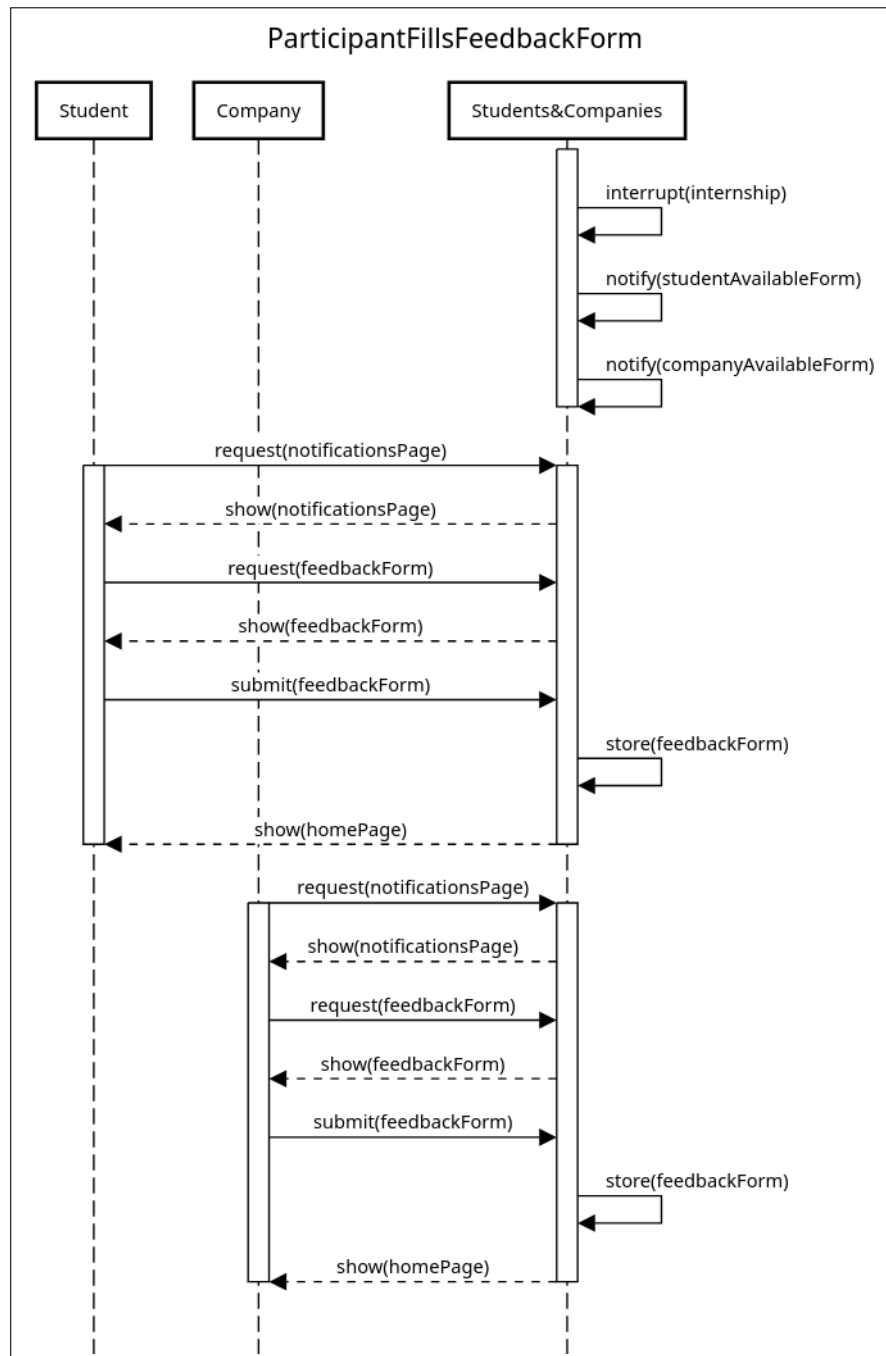




## UC18 - InternshipExpires



## UC19 - ParticipantFillsFeedbackForm



### 3.2.4 Use cases mapping

Goal 1	
G1	Allow registered students to search and enroll for internship opportunities.
D1	The user must have a working internet connection.
D2	The user must have provided valid personal information.
D5	The university must have been registered in the system directly by a staff member.
R4	The system must allow a registered user to log in.
R5	The system must allow a registered student to fill in and edit its personal information.
R6	The system must allow a registered student to upload its CV.
R7	The system must allow a registered company to post an internship project.
R8	The system must allow a registered student to visualize a list of open internship projects.
R10	The system must allow a registered student to make and enrollment request to an internship project.
R11	The system must allow a registered company to build custom made questionnaires.
R12	The system must allow a registered company to send questionnaires to students.
R13	The system must allow a registered student to fill in the questionnaire.
R14	The system must allow a registered company to accept students enrollment requests.
R22	The system must allow a registered student to fill in a feedback form when the internship ends.
R26	The system must allow a registered student to be notified about recommended internship.

<b>Goal 2</b>	
<b>G2</b>	Allow registered companies to advertise internship project opportunities.
<b>D1</b>	The user must have a working internet connection.
<b>D2</b>	The user must have provided valid personal information.
<b>R4</b>	The system must allow a registered user to log in.
<b>R7</b>	The system must allow a registered company to post an internship project.
<b>R9</b>	The system must allow a registered company to visualize a list of eligible students.
<b>R24</b>	The system must allow a registered student to visualize a list of suggested internships.
<b>R25</b>	The system must allow a registered company to visualize a list of suggested students.
<b>R26</b>	The system must allow a registered student to be notified about recommended internship.
<b>R27</b>	The system must allow a registered company to be notified about recommended students.

<b>Goal 3</b>	
<b>G3</b>	Allow registered universities to monitor their students ongoing internship and manage complaints.
<b>D1</b>	The user must have a working internet connection.
<b>D2</b>	The user must have provided valid personal information.
<b>D3</b>	The student must be registered to a university.
<b>D4</b>	The university must have provided an organization mail to the student.
<b>D5</b>	The university must have been registered in the system directly by a staff member.
<b>R1</b>	The system must allow an unregistered student to sign up.
<b>R2</b>	The system must allow an unregistered company to sign up.
<b>R4</b>	The system must allow a registered user to log in.
<b>R17</b>	The system must allow a registered university to see their students' ongoing internship information.
<b>R18</b>	The system must allow a registered student to send complaints to the university.
<b>R19</b>	The system must allow a registered company to send complaints to the university.
<b>R20</b>	The system must allow a registered university to visualize complaints it received.
<b>R21</b>	The system must allow a registered university to end an ongoing internship of its student.
<b>R22</b>	The system must allow a registered student to fill in a feedback form when the internship ends.
<b>R23</b>	The system must allow a registered company to fill in a feedback form when the internship ends.

<b>Goal 4</b>	
<b>G4</b>	Support companies in the selection process by providing students with custom-made questionnaires.
<b>D1</b>	The user must have a working internet connection.
<b>D2</b>	The user must have provided valid personal information.
<b>D5</b>	The university must have been registered in the system directly by a staff member.
<b>R4</b>	The system must allow a registered user to log in.
<b>R9</b>	The system must allow a registered company to visualize a list of eligible students.
<b>R11</b>	The system must allow a registered company to build custom made questionnaires.
<b>R12</b>	The system must allow a registered company to send questionnaires to students.
<b>R14</b>	The system must allow a registered company to accept students enrollment requests.

<b>Goal 5</b>	
<b>G5</b>	Ease matching by notifying students of relevant internships and companies for suitable candidates.
<b>D1</b>	The user must have a working internet connection.
<b>D2</b>	The user must have provided valid personal information.
<b>D3</b>	The student must be registered to a university.
<b>D4</b>	The university must have provided an organization mail to the student.
<b>D5</b>	The university must have been registered in the system directly by a staff member.
<b>R1</b>	The system must allow an unregistered student to sign up.
<b>R2</b>	The system must allow an unregistered company to sign up.
<b>R4</b>	The system must allow a registered user to log in.
<b>R5</b>	The system must allow a registered student to fill in and edit its personal information.
<b>R6</b>	The system must allow a registered student to upload its CV.
<b>R7</b>	The system must allow a registered company to post an internship project.
<b>R8</b>	The system must allow a registered student to visualize a list of open internship projects.
<b>R9</b>	The system must allow a registered company to visualize a list of eligible students.
<b>R10</b>	The system must allow a registered student to make and enrollment request to an internship project.
<b>R11</b>	The system must allow a registered company to build custom made questionnaires.
<b>R12</b>	The system must allow a registered company to send questionnaires to students.
<b>R13</b>	The system must allow a registered student to fill in the questionnaire.
<b>R14</b>	The system must allow a registered company to accept students enrollment requests.
<b>R15</b>	The system must allow a registered student to see their ongoing internship information.
<b>R16</b>	The system must allow a registered company to see their ongoing internships information.
<b>R22</b>	The system must allow a registered student to fill in a feedback form when the internship ends.
<b>R23</b>	The system must allow a registered company to fill in a feedback form when the internship ends.
<b>R24</b>	The system must allow a registered student to visualize a list of suggested internships.
<b>R25</b>	The system must allow a registered company to visualize a list of suggested students.
<b>R26</b>	The system must allow a registered student to be notified about recommended internship.
<b>R27</b>	The system must allow a registered company to be notified about recommended students.

<b>Goal 6</b>	
<b>G6</b>	Provide suggestions to both parties to refine their submissions.
<b>D1</b>	The user must have a working internet connection.
<b>D2</b>	The user must have provided valid personal information.
<b>R4</b>	The system must allow a registered user to log in.
<b>R5</b>	The system must allow a registered student to fill in and edit its personal information.
<b>R6</b>	The system must allow a registered student to upload its CV.
<b>R7</b>	The system must allow a registered company to post an internship project.
<b>R8</b>	The system must allow a registered student to visualize a list of open internship projects.
<b>R9</b>	The system must allow a registered company to visualize a list of eligible students.
<b>R24</b>	The system must allow a registered student to visualize a list of suggested internships.
<b>R25</b>	The system must allow a registered company to visualize a list of suggested students.
<b>R26</b>	The system must allow a registered student to be notified about recommended internship.
<b>R27</b>	The system must allow a registered company to be notified about recommended students.

### 3.3 Performance requirements

Given the non-critical nature of the system, there are no overly stringent performance requirements. However, the system strives to satisfy the following requirements in order to offer the best user experience.

#### 3.3.1 Specific requirements

- SR1** - The system should handle at least 1000 user requests per second.
- SR2** - The system should respond to web page requests in under 2 seconds.
- SR3** - The system should let users sign up within 5 seconds.
- SR4** - The system should let users log in within 2 seconds.
- SR5** - The system should let users access their profile information within 2 seconds.
- SR6** - The system should let students upload their CV within 5 seconds.
- SR7** - The system should let companies post advertisements within 2 seconds.
- SR8** - The system should let students retrieve an advertisement project list within 2 seconds.
- SR9** - The system should let companies retrieve a suitable student list within 2 seconds.
- SR10** - The system should let students send an application request within 2 seconds.
- SR11** - The system should let companies store a newly built custom questionnaire within 2 seconds.
- SR12** - The system should let students submit a filled questionnaire within 2 seconds.
- SR13** - The system should let companies accept a student enrollment within 2 seconds.
- SR14** - The system should let students accept a company proposal within 2 seconds.

- SR15** - The system should initiate an internship instance in under 5 seconds.
- SR16** - The system should let users visualize ongoing internship information within 2 seconds.
- SR17** - The system should let students and companies send complaints in under 2 seconds.
- SR18** - The system should let universities visualize complaints within 2 seconds.
- SR19** - The system should let universities end an internship within 2 seconds.
- SR20** - The system should finish an internship instance in under 5 seconds.
- SR21** - The system should let students and companies submit a filled feedback form within 2 seconds.
- SR22** - The system should allow users to receive notifications in under 10 seconds.
- SR23** - The system should find valid internship projects recommendations in under 5 seconds.
- SR24** - The system should find valid suitable students in under 5 seconds.
- SR25** - The system should find valid profile suggestions in under 5 seconds.

## **3.4 Design constraints**

This section outlines the constraints impacting design choices, including regulatory standards, technical limitations, and specific requirements. These constraints guide compliance and design feasibility.

### **3.4.1 Standards compliance**

Specifications described in this document must be respected by the system. The source code of the application must be commented and documented adequately.

Moreover, the system should respect the laws about privacy and data protection of the country in which it is used.

### **3.4.2 Hardware limitations**

To access the system the user must have a device with a web browser and an internet connection.

## **3.5 Software system attributes**

This section describes essential software attributes like performance, security, and maintainability, which define the quality and user experience of the system.

### **3.5.1 Availability**

The system must operate continuously, ensuring 24/7 access with downtime below 1%. Any failure must be resolved within 1 hour to minimize service disruption.

### **3.5.2 Reliability**

To maintain consistent performance, the system should be designed with high redundancy and fault tolerance, reducing the risk of critical failures.



### **3.5.3 Security**

Data encryption is crucial to protect sensitive user information, including personal details and passwords. All such data must be encrypted both in transit and at rest, ensuring it remains secure against unauthorized access.

Access control is enforced through a robust authentication and authorization mechanism that restricts data and functionality access according to user roles. Passwords are securely hashed and stored, further safeguarding user credentials.

The system ensures responsible handling of personal data, requiring explicit user consent and implementing clear procedures for data protection. Users have the right to access their data and request its deletion, promoting transparency and control over personal information.

To uphold high security standards, regular security audits and vulnerability assessments are scheduled. These help identify and mitigate any emerging risks, continuously enhancing system security.

### **3.5.4 Maintainability**

The system must be designed with maintainability in mind, ensuring that future updates, bug fixes, and feature enhancements can be implemented efficiently.

### **3.5.5 Portability**

The system is designed to be platform-independent, accessible from any device with a web browser and an internet connection. The user interface is responsive and optimized for various screen sizes and resolutions.

## 4 Formal analysis using Alloy

In our formal analysis, we have focused on modeling a dynamic system for internship management, incorporating both static and temporal constraints. Our model aims to represent the interplay between students, universities, companies, advertisements, and internships, while ensuring the consistency of assignments, status transitions, and relationships among the entities.

We leveraged Alloy temporal logic features to capture the evolution of internships and advertisements over time, allowing us to model preconditions, invariants, and dynamic behavior in a rigorous manner. The design emphasizes clarity and precision, ensuring that key system properties, such as unique assignment of internships and adherence to procedural rules, are verifiable.

To maintain a balance between complexity and manageability, we incorporated multiple predicates and facts that ensure a consistent representation of system behaviors, such as application processing, advertisement closure, and feedback provision. Temporal constraints, such as ensuring no return to previous states (e.g., WAITING) or eventual resolution of complaints, were introduced to demonstrate the system progression over time without overcomplicating the analysis. This approach enabled us to comprehensively capture the problem intricacies while facilitating the exploration of its temporal dynamics.

### 4.1 Alloy Code

```
----- DEFINITION OF ENUM -----  
  
enum InternshipStatus {WAITING, ONGOING, REVIEW, FINISHED}  
enum Bool {true, false}  
  
----- DEFINITION OF SIGNATURES -----  
  
abstract sig User {}  
  
sig Student extends User {  
    skills: set Skill,  
    var assignedInternship: lone Internship,  
}{  
    #skills > 0  
}
```

```

sig Company extends User {
  advertisement: set Advertisement,
}

sig University extends User {
  students: set Student,
}

sig Advertisement {
  requirements: set Skill,
  description: one Description,
  spots: one Int,
  var applicants: set Student,
  var assigned: one Bool,
  internship: set Internship,
}{
  spots >= 1 and spots <= 3
}

sig Internship {
  offeredBy: one Company,
  var assignedTo: lone Student,
  var feedback: lone Feedbacks,
  var status: one InternshipStatus,
}

sig Feedbacks {}

sig Description{}

sig Skill {}

```

```

----- PREDICATES -----

-- This predicate handles the application of a student to an advertisement
pred Apply[s: Student, a: Advertisement] {

  -- Precondition: all internships related to the advertisement must have
    a status of WAITING
  all i: a.internship |
    i.status = WAITING

  -- No changes allowed
  a.assigned' = a.assigned
  all i: a.internship | {
    i.status' = i.status and
    i.assignedTo' = i.assignedTo
  }
}

```

```

    -- Postcondition: the student is added to the list of applicants
    a.applicants' = a.applicants + s
}

-- This predicate closes an advertisement and assigns internships to
  applicants
pred CloseAdv[a: Advertisement] {

    -- Precondition: the number of applicants must be greater than or equal
      to the available spots, and the advertisement is not yet assigned
    #a.applicants >= a.spots
    a.assigned = false
    all i: a.internship |
        i.status = WAITING

    -- No changes allowed
    a.applicants' = a.applicants

    -- Postcondition: the advertisement is marked as assigned, and
      internships are assigned
    a.assigned' = true
    AssignInternship[a]
}

-- This predicate assigns internships to students in an advertisement
pred AssignInternship[a: Advertisement] {

    -- Precondition: the number of applicants must be greater than or equal
      to the available spots, and the advertisement is not yet assigned
    #a.applicants >= a.spots
    a.assigned = false
    all i: a.internship |
        i.status = WAITING

    -- Postcondition: every internship is assigned to a unique student
    all i: a.internship | {
        some s: a.applicants | {
            i.assignedTo' = s and
            assignedInternship' = ~assignedTo'
        }
    }

    -- Ensure that each internship is assigned to a different student
    all disj i1, i2: a.internship |
        i1.assignedTo' != i2.assignedTo'

    -- Update the status of all internships to ONGOING
    all i: a.internship |
        i.status' = ONGOING

```

```

}

-- This predicate closes an internship, marking it as finished
pred CloseInternship[i: Internship] {

    -- Precondition: the internship must have a status of ONGOING
    i.status = ONGOING

    -- No changes allowed to the assigned student
    i.assignedTo' = i.assignedTo

    -- Postcondition: the internship status is updated to FINISHED, and
    feedback is provided
    i.status' = FINISHED
    some i.feedback'
}

-- This predicate opens a complaint for an internship
pred OpenComplaint[i: Internship] {

    -- Precondition: the internship must have a status of ONGOING
    i.status = ONGOING

    -- No changes allowed
    i.assignedTo' = i.assignedTo
    i.assignedTo.assignedInternship' = i.assignedTo.assignedInternship

    -- Postcondition: the internship status is updated to REVIEW
    i.status' = REVIEW
}

-- This predicate resolves a complaint for an internship
pred ResolveComplaint[i: Internship] {

    -- Precondition: the internship must have a status of REVIEW
    i.status = REVIEW

    -- No changes allowed
    i.assignedTo' = i.assignedTo
    i.assignedTo.assignedInternship' = i.assignedTo.assignedInternship

    -- Postcondition: the internship status is updated to either ONGOING or
    FINISHED
    i.status' = ONGOING or i.status' = FINISHED
}

----- INITIAL DYNAMIC CONDITIONS -----

-- Ensures that initially, no student is assigned to any internship

```

```

fact InitialConditionStudent {
    all s: Student |
        no s.assignedInternship
}

-- Ensures that initially, no advertisement has any applicants, and none
-- are assigned
fact InitialConditionAdv {
    all a: Advertisement | {
        #a.applicants = 0 and
        a.assigned = false
    }
}

-- Ensures that initially, internships have no assigned students, no
-- feedback, and are in a WAITING state
fact InitialConditionInternship {
    all i: Internship | {
        no i.assignedTo and
        no i.feedback and
        i.status = WAITING
    }
}

----- STATIC FACTS -----

-- Ensures that the number of spots in an advertisement equals the number
-- of internships associated with it
fact SpotForInternship {
    all a: Advertisement |
        a.spots = #a.internship
}

-- Ensures that each internship is associated with exactly one
-- advertisement
fact UniqueAdvertisementForEachInternship {
    all i: Internship |
        one a: Advertisement | i in a.internship
}

-- Ensures that each student belongs to one and only one university
fact UniqueUniversityPerStudent {
    all s: Student | {
        one u: University | s in u.students and
        all disj u1, u2: University | {
            s in u1.students => s not in u2.students
        }
    }
}

```

```

-- Ensures that no university is empty; every university must have at least
  one student
fact NoEmptyUniversity {
  all u: University | some u.students
}

-- Ensures that each advertisement belongs to exactly one company
fact UniqueCompanyPerAdvertisement {
  all a: Advertisement | {
    one c: Company | a in c.advertisement and
    all disj c1, c2: Company | {
      a in c1.advertisement => a not in c2.advertisement
    }
  }
}

-- Ensures that no company is empty; every company must have at least one
  advertisement
fact NoEmptyCompany {
  all c: Company | some c.advertisement
}

-- Ensures that internships are offered by the company that published their
  associated advertisement
fact ConsistentCompanyAdvertisementInternship {
  all c: Company | all a: c.advertisement | {
    all i: a.internship | i.offeredBy = c
  }
}

-- Ensures that all advertisements have unique descriptions
fact UniqueDescriptions {
  all disj a1, a2: Advertisement | a1.description != a2.description
}

-- Ensures that the number of the Feedbacks is equal to the number of
  Internship
fact NumOfInternshipMatchNumOfFeedback {
  #Feedbacks = #Internship
}

----- ALWAYS TRUE FACTS -----

-- Ensures that once an internship is marked as FINISHED, it will always
  not change
fact InternshipStayFinished {
  all i: Internship |
    always (i.status = FINISHED => some i.feedback and

```

```

        always (i.status = FINISHED and i.feedback' = i.feedback ))
    }

    -- Ensures that once an advertisement is assigned, it will always remain
    -- assigned and its list of applicants will not change
    fact AdvertisementStayClose {
        all a: Advertisement |
            always (a.assigned = true => always (a.assigned = true and a.
                applicants' = a.applicants))
    }

    -- Ensures that internships in certain statuses (ONGOING, REVIEW, FINISHED)
    -- always have an assigned student, and those in WAITING have no assigned
    -- student
    fact AssignedStatusConsistency {
        all i: Internship |
            always (i.status in (ONGOING + REVIEW + FINISHED) => some i.
                assignedTo and i.assignedTo' = i.assignedTo) and
            always (i.status = WAITING => no i.assignedTo)
    }

    -- Ensures that internships in ONGOING or REVIEW statuses always have their
    -- assigned student linked to that internship
    fact InternshipAssignedStudentConsistency {
        all i: Internship |
            always (i.status in (ONGOING + REVIEW) => i.assignedTo.
                assignedInternship = i)
    }

    -- Ensures that a student cannot have an assigned internship before it
    -- starts or after it is finished
    fact ValidStudentAssignment {
        all s: Student |
            always (some s.assignedInternship => s.assignedInternship.status in
                (ONGOING + REVIEW))
    }

    -- Ensures consistent historical state transitions for internships
    fact CheckStateHistory {
        all i: Internship | {
            always (i.status = ONGOING => once i.status = WAITING)
            always (i.status = REVIEW => once i.status = ONGOING)
            always (i.status = FINISHED => once i.status = ONGOING)
        }
    }

    -- Ensures that if an advertisement is not assigned, all its internships
    -- must be in the WAITING state
    fact AdvOpen {
        all a: Advertisement |

```



```

        always (a.assigned = false => (all i: a.internship | i.status =
            WAITING))
    }

    -- Ensures that once an internship leaves the WAITING state, it cannot
    -- return to it
    fact NoReturnToWaiting {
        all i: Internship | always (once (i.status != WAITING) => always (i.
            status != WAITING))
    }

    -- Ensures that any student assigned to an internship must have applied to
    -- its advertisement
    fact AssignedStudentApplied {
        always (all i: Internship | {
            some i.assignedTo => some a: Advertisement | {
                i in a.internship and i.assignedTo in a.applicants
            }
        })
    }

    -- Ensures that feedback can only be given after an internship is finished
    fact NoFeedbackBeforeFinished {
        always (all i: Internship | {
            i.status != FINISHED => no i.feedback
        })
    }

    -- Ensures that feedback is unique for each internship
    fact UniqueFeedbackPerInternship {
        always (all disj i1, i2: Internship |
            i1.status = FINISHED and i2.status = FINISHED => i1.feedback != i2.
            feedback)
    }

    ----- DYNAMIC FACTS -----

    -- Ensures that if an advertisement is not assigned, eventually a student
    -- will apply to it
    fact EventuallyApply {
        all a: Advertisement | {
            a.assigned = false => eventually (some s: Student |
                no s.assignedInternship and s not in a.applicants and Apply[s,
                a])
        }
    }

    -- Ensures that if an advertisement is not assigned, it will eventually be
    -- closed

```

```

fact EventuallyCloseAdv {
  all a: Advertisement |
    a.assigned = false => eventually (CloseAdv[a])
}

-- Ensures that all internships will eventually be closed
fact EventuallyCloseInternship {
  all i: Internship |
    eventually (CloseInternship[i])
}

-- Ensures that eventually, at least one internship will have a complaint
  opened
fact EventuallyOpenComplaint {
  some i: Internship |
    eventually OpenComplaint[i]
}

-- Ensures that internships in the REVIEW state will have their complaints
  resolved
fact AlwaysResolveComplaint {
  all i: Internship | {
    always (i.status = REVIEW => ResolveComplaint[i])
  }
}

----- ASSERTIONS -----

-- Ensures that an internship can only be assigned to a student if the
  student has applied to the associated advertisement
assert InternshipAssignmentRequiresApplication {
  all i: Internship |
    some i.assignedTo => some a: Advertisement |
      i in a.internship and i.assignedTo in a.applicants
}

-- Ensures that internships follow the exact state sequence: WAITING ->
  ONGOING -> REVIEW (optional) -> FINISHED, without skipping or repeating
  states
assert ExactStateSequence {
  all i: Internship | {
    always (i.status = WAITING => not once (i.status in (ONGOING +
      REVIEW + FINISHED)))
    always (i.status = ONGOING => once (i.status = WAITING) and not
      once (i.status = FINISHED))
    always (i.status = REVIEW => once (i.status in (ONGOING + WAITING)
      and not once (i.status = FINISHED)))
    always (i.status = FINISHED => once (i.status = REVIEW or i.status
      = ONGOING))
  }
}

```

```

    }
}

-- Ensures that if an internship enters the REVIEW state, it will
   transition to the FINISHED state
assert ReviewEventuallyLeadsToFinished {
  all i: Internship | {
    once (i.status = REVIEW) => always (i.status = FINISHED)
  }
}

-- Ensures that an advertisement can only be closed if all its internships
   are assigned
assert NoClosedAdvertisementWithoutAssignedInternships {
  all a: Advertisement |
    a.assigned = true => all i: a.internship | some i.assignedTo
}

-- Ensures that each student is assigned to only one internship at a time
assert UniqueAssignmentPerStudent {
  always (all disj i1, i2: Internship |
    (i1.status != WAITING and i2.status != WAITING) => i1.assignedTo !=
    i2.assignedTo)
}

----- RUN -----

check InternshipAssignmentRequiresApplication for 5
check ExactStateSequence for 5
check ReviewEventuallyLeadsToFinished for 5
check NoClosedAdvertisementWithoutAssignedInternships for 5
check UniqueAssignmentPerStudent for 5

pred show {
  #Advertisement = 2
  #internship = 5
  #University = 2
  #Company = 2
  #Student = 6
  #Skill = 2
  eventually (one i: Internship | i.status = REVIEW)
}

run show for 10

```

## 4.2 Assertion Results

### Executing "Check InternshipAssignmentRequiresApplication for 5"

Solver=sat4j Steps=1..10 Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
1..10 steps. 412546 vars. 10900 primary vars. 864193 clauses. 1400ms.  
No counterexample found. Assertion may be valid. 6ms.

An Internship can only be assigned to a student if it has applied to the associated advertisement

### Executing "Check ExactStateSequence for 5"

Solver=sat4j Steps=1..10 Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
1..10 steps. 590524 vars. 11320 primary vars. 1313792 clauses. 3826ms.  
No counterexample found. Assertion may be valid. 508ms.

Internships follow the exact temporal sequence

### Executing "Check ReviewEventuallyLeadsToFinished for 5"

Solver=sat4j Steps=1..10 Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
1..10 steps. 413500 vars. 10910 primary vars. 865750 clauses. 1416ms.  
No counterexample found. Assertion may be valid. 8ms.

if an internship enters the REVIEW state, it will transition to the FINISHED state

### Executing "Check NoClosedAdvertisementWithoutAssignedInternships for 5"

Solver=sat4j Steps=1..10 Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
1..10 steps. 412486 vars. 10950 primary vars. 864043 clauses. 1836ms.  
No counterexample found. Assertion may be valid. 28ms.

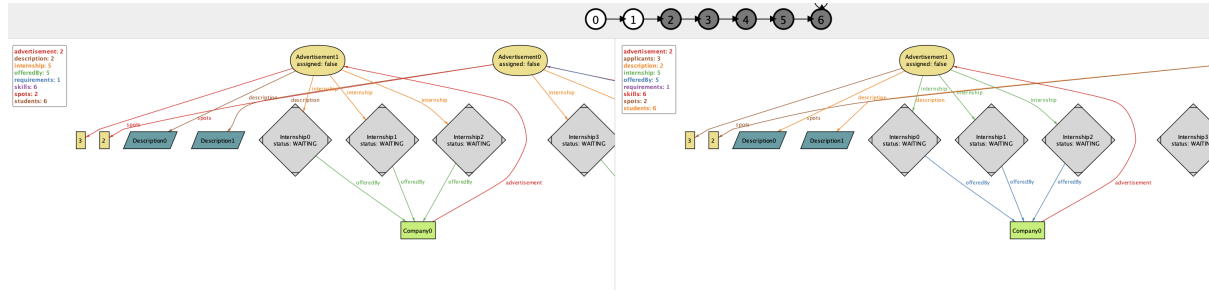
An advertisement can only be closed if all its internships are assigned

### Executing "Check UniqueAssignmentPerStudent for 5"

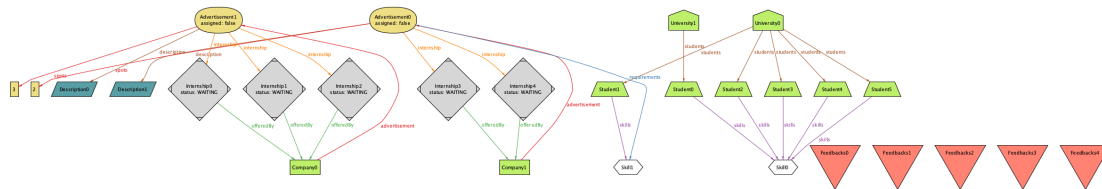
Solver=sat4j Steps=1..10 Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20 Mode=batch  
1..10 steps. 416394 vars. 11060 primary vars. 874886 clauses. 2088ms.  
No counterexample found. Assertion may be valid. 337ms.

Each student is assigned to only one internship at a time

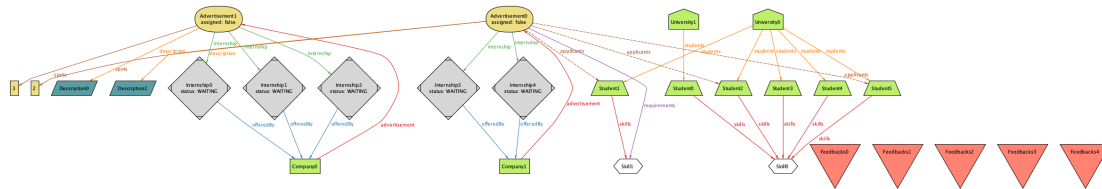
### 4.3 World Generated



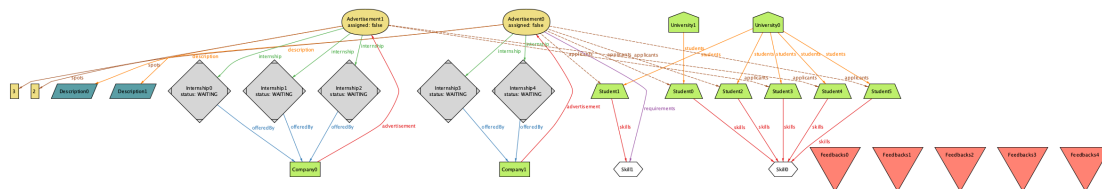
Temporal steps



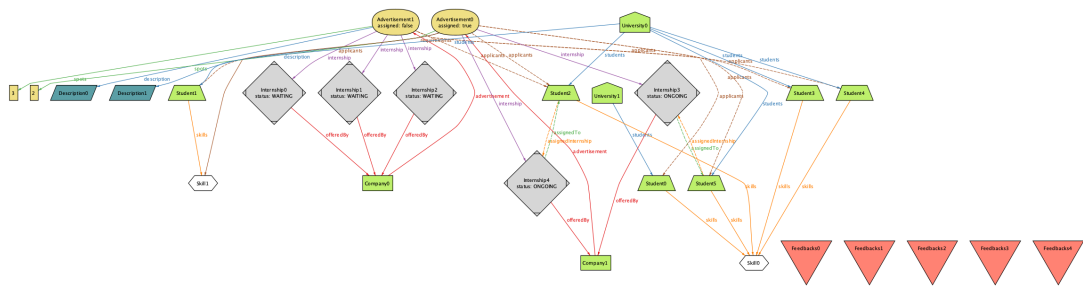
### I - Temporal step



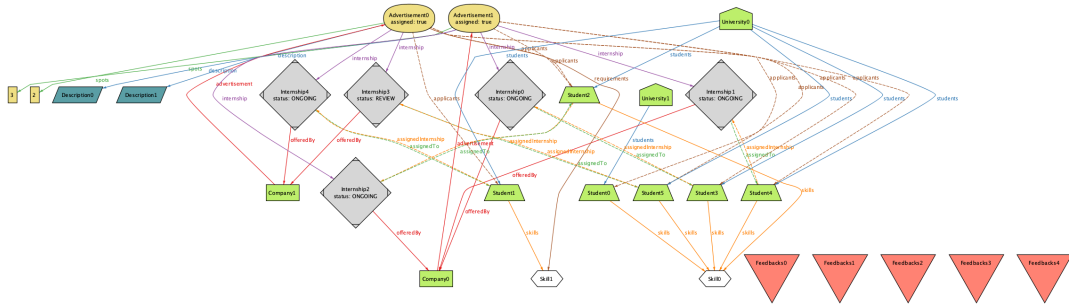
## II - Temporal step



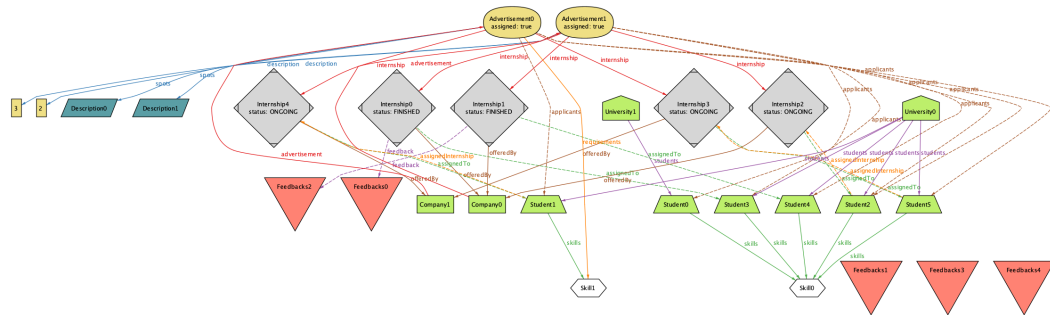
### III - Temporal step



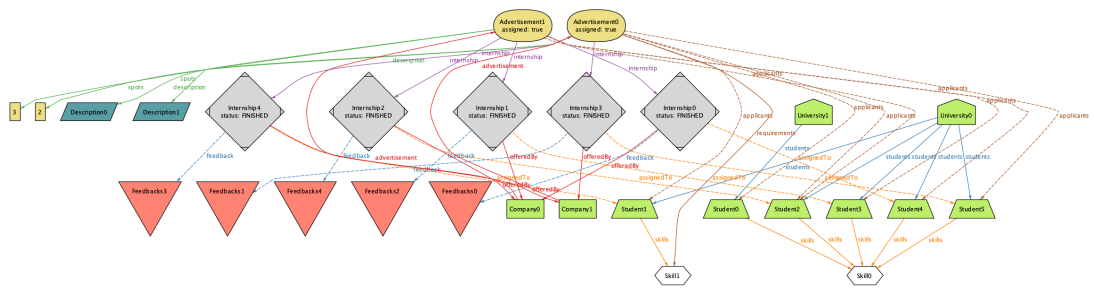
IV - Temporal step



V - Temporal step



VI - Temporal step



## VII - Temporal step

## 5 Effort spent

Unit	Member	Hours
Setup	Ostidich	4
Introduction	Ostidich, Rivitti	2
Overall description	Ostidich, Salari	6
Class diagram	Ostidich, Rivitti	3
State charts	Ostidich	2
Specific requirements	Ostidich, Rivitti	7
User interfaces	Rivitti	6
Use cases diagrams	Ostidich	3
Use cases	Ostidich, Rivitti	5
Sequence diagrams	Ostidich, Rivitti	12
Mapping	Rivitti	3
Alloy	Salari	24



## 6 References

1. The UI mockups have been created using [figma.com](https://figma.com)
2. The document has been written using [latex-project.org](https://latex-project.org)
3. The sequence diagrams have been created using [sequencediagram.org](https://sequencediagram.org)
4. The other diagrams have been created using [draw.io](https://draw.io)
5. The Alloy tool used can be found on [alloytools.org](https://alloytools.org)
6. The project repository has been uploaded on [github.com](https://github.com)