



**POLITECNICO**  
**MILANO 1863**

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

# Software Engineering II

Document: **Requirement Analysis and Specification**  
**Document - Students & Companies**

Author(s): **APPOURCHAUX Léo 11078414**  
**YE Yumeng 11072406**

Advisor: **Prof. DI NITTO ELISABETTA**

Academic Year: 2024-2025



# Contents

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.1.1	Goals . . . . .	1
1.2	Scope . . . . .	2
1.2.1	World Phenomena . . . . .	2
1.2.2	Shared Phenomena . . . . .	2
1.3	Definitions, Acronyms, Abbreviations . . . . .	3
1.4	Revision History . . . . .	4
1.5	Reference Documents . . . . .	4
1.6	Document Structure . . . . .	5
<b>2</b>	<b>Overall Description</b>	<b>7</b>
2.1	Product Perspective . . . . .	7
2.1.1	Scenarios . . . . .	7
2.1.2	Domain class diagram . . . . .	9
2.1.3	State diagram . . . . .	10
2.2	Product functions . . . . .	10
2.3	User characteristics . . . . .	11
2.4	Assumptions, dependencies and constraints . . . . .	12
2.4.1	Regulatory policies: . . . . .	12
2.4.2	Domain Assumptions: . . . . .	12
<b>3</b>	<b>Specific Requirements</b>	<b>13</b>
3.1	External Interface Requirements . . . . .	13
3.1.1	User Interfaces . . . . .	13
3.1.2	Hardware Interfaces . . . . .	15

3.1.3	Software Interfaces . . . . .	15
3.1.4	Communication Interfaces . . . . .	16
3.2	3.2 Functional Requirements . . . . .	16
3.2.1	Use cases diagram . . . . .	19
3.2.2	Use cases . . . . .	20
3.2.3	Sequence diagrams . . . . .	36
3.2.4	Requirement mapping . . . . .	53
3.3	Performance Requirements . . . . .	57
3.4	Design Constraints . . . . .	57
3.4.1	Standards compliance . . . . .	58
3.4.2	Hardware limitations . . . . .	58
3.4.3	Any other constraint . . . . .	58
3.5	Software System Attributes . . . . .	58
3.5.1	Reliability . . . . .	58
3.5.2	Availability . . . . .	59
3.5.3	Security . . . . .	59
3.5.4	Maintainability . . . . .	59
3.5.5	Portability . . . . .	59
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>61</b>
4.1	Model Overview . . . . .	61
4.2	Code Description . . . . .	61
4.3	Scenario 1: Two Students and One Internship Offer . . . . .	64
4.3.1	Explanation of Scenario . . . . .	64
4.3.2	Code for Scenario 1 . . . . .	64
4.4	Scenario 2: No Application and Recommendation Conflict . . . . .	66
4.4.1	Explanation of Scenario . . . . .	66
4.4.2	Code for Scenario 2 . . . . .	66
<b>5</b>	<b>Effort Spent</b>	<b>69</b>

# 1 | Introduction

## 1.1. Purpose

In today's rapidly evolving job market, it can be difficult for students to transition from university to the working world. Students struggle to find opportunities that fit their skills and career goals, and companies spend a huge amount of time and effort trying to identify the right candidates. It's critical to tackle this problem as a mismatch can influence a student's early career development and a company's ability to gather relevant talents. Students&Companies (S&C) tries to solve this problem. The platform allows students to search for internships, receive personalized recommendations, and refine their profiles to better align with industry expectations. Companies can advertise internships, receive tailored candidate suggestions, and benefit from a simplified hiring process through structured tools and mechanisms. Universities can monitor internships, ensure educational standards are met, and handle complaints or issues that may arise during internships.

### 1.1.1. Goals

- [G1] Students search for internships that align with their skills and career goals
- [G2] Companies advertise their internship offers
- [G3] The platform facilitates connections between companies and students and allows structured recruitment processes through interviews and tests
- [G4] Students and companies receive personalized suggestions for CV or offer improvement
- [G5] The platform sends recommendations to students and companies by matching student profiles and internship offers
- [G6] Universities monitor internship progress, complaints and issues

## 1.2. Scope

In this section, we define the domain of the Students&Companies(S&C) platform, which serves two primary user groups: students and companies. The platform is designed to simplify the process of matching students with internship opportunities based on their skills and aspirations, while allowing companies to find the most suitable candidates for their internship offers. For students, the process begins with creating an account on the platform, where they upload their CVs and other relevant documents. Companies, on the other hand, create detailed internship offers that describe the tasks, required skills and benefits. The platform uses matching algorithms to help both parties find the best fit. Once students apply, companies can review CVs, send assessments, schedule interviews, and select candidates. The platform also sends personalized suggestions:

1. To students: the platform provides suggestions for improving their CVs and profiles to better match industry expectations
2. To companies: the platform suggests ways to make internship offers more appealing

After each selection process, the platform collects feedback from both students and companies. This feedback is analyzed and shared with relevant parties to refine recommendations and further improve the matching process.

### 1.2.1. World Phenomena

These are the real-world events or actions that influence how the platform works:

[WP1] Companies define internship needs

[WP2] Students prepare and update their CVs

[WP3] Students decide which internships to apply for

[WP4] Companies decide which candidates to interview or assess

[WP5] Students and companies provide feedback on the selection process and interactions

### 1.2.2. Shared Phenomena

These phenomena are controlled by both the world (students, companies, universities) and the machine (platform algorithms and systems):

#### World-Controlled Phenomena

- [SP1] Students create accounts and upload relevant documents (e.g., CVs)
- [SP2] Companies post internships on the platform

- [SP3] Students search for and browse internships
- [SP4] Students apply for internships
- [SP5] Students receive recommendations and decide to accept or decline
- [SP6] Companies receive and evaluate student applications
- [SP7] Companies receive recommendations and decide to accept or decline
- [SP8] Companies initiate interviews or assessments for selected candidates
- [SP9] Universities monitor internship and address complaints or issues
- [SP10] Students update their profiles

#### Machine-Controlled Phenomena

- [SP11] The system matches students with internships
- [SP12] The system notifies students about relevant internships
- [SP13] The system alerts companies about suitable candidates
- [SP14] The system establishes contact between students and companies
- [SP15] The system processes assessment results
- [SP16] The system sends assessment results to students and companies
- [SP17] The system tracks and updates the selection status (accepted, ongoing, rejected)
- [SP18] The platform collects feedback from students and companies and shares it with relevant parties
- [SP19] The platform provides suggestions to students on improving their CVs and profiles
- [SP20] The platform provides suggestions to companies on making their internship offers more appealing

### 1.3. Definitions, Acronyms, Abbreviations

- S&C: Students&Companies
- CV: Curriculum Vitae
- UI: User Interface

- API: Application Programming Interface
- DBMS: Database Management System
- SLA: Service Level Agreement
- 2FA: Two-Factor Authentication
- GDPR: General Data Protection Regulation

## 1.4. Revision History

- Version 1 (24/11/2024)

## 1.5. Reference Documents

The document is based on the following materials:

- The specification of the RASD and DD assignment of the Software Engineering II course a.a. 2024/25
- Slides of the course on WeBeep



## 1.6. Document Structure

1. Introduction: it aims to give a brief description of the project. In particular it's focused on the reasons and the goals that are going to be achieved with its development
2. Overall Description: it is an high-level description of how the system works with a detailed explanation of the phenomena that involve the world,the machine or both,there is also the domain description with its assumptions
3. Specific Requirements: in this section there is a detailed analysis of the requirements needed to achieve the goals. Moreover, it contains more information useful for developers (i.e constraints about HW and SW)
4. Formal analysis: it's a formal description of the world phenomena by the means of Alloy
5. Effort spent: it shows the time spent to realize this document organized by section
6. References: it contains the references to any documents and software used to write this paper



## 2 | Overall Description

### 2.1. Product Perspective

#### 2.1.1. Scenarios

**Student uploads/updates a CV:**

Students can upload an already existing CV on the platform and update their CVs by uploading a new one.

**Student browses and applies for the offers:** The student can look at the list of offers and apply different filters (e.g., location, time, field, level of education) and filter by using keywords. They can then apply by creating an application, which sends a notification to the company.

**Student or Company receives a suggestion:** The student and the company periodically receives suggestions. For the student it's about how to improve their CV and can follow the suggestion step by step. for the company it's about how to make the offers / project descriptions more appealing to students.

**Student or Company receives a recommendation:** With all the data at its disposal, the matching algorithm computes best matches and creates recommendation for companies and students. When a match is computed by the matching algorithm, a recommendation is sent to both the company and the user in the match. The company can accept it and the platform will send a notification to the corresponding student.

**User participates to a complaint space:** The user can submit a complaint or take part to the conversation to resolve the issue.

**Student takes a test:** During the recruiting process, the student may take a test for a specific offer.

**Student or company participates in an interview:** During the recruiting process, the student and the company can organise and participate to an interview together.

**Student monitors recruiting process:** The student can monitor the current state of the recruiting process.

**User provides feedback:** The user can provide feedback at any time about the platform and the suggestions or recommendations they received. The feedback is then sent to the database of the matching algorithm as additional data to perform analysis onto.

**Company creates an offer:** The company can create a new offer and specify all the details about the field, time, location, expected degree. This offer is then available for all students to see and advertised to those that may match.

**Company evaluates an application:** When a student applies for a specific offer the company can accept the application if they want to proceed with the recruiting process with this student. Once the user and the company both agree to their counterpart's offer, the company can communicate, send testing materials, and interview the student on the platform. This starts the recruiting process.

**Company starts the recruiting process:** Once the user and the company both agree to their counterpart's offer or the company finds a suitable candidate, the company can send invitations to students for interview or tests.

**Company collects questionnaire answers:** Once the user has taken the questionnaire, the company can collect the answers and include them in their decision process.

**University monitors the recruiting process:** The university has access to tools to monitor the status of the internships and to get info. They can interrupt an internship if an issue arises.

2.1.2. Domain class diagram

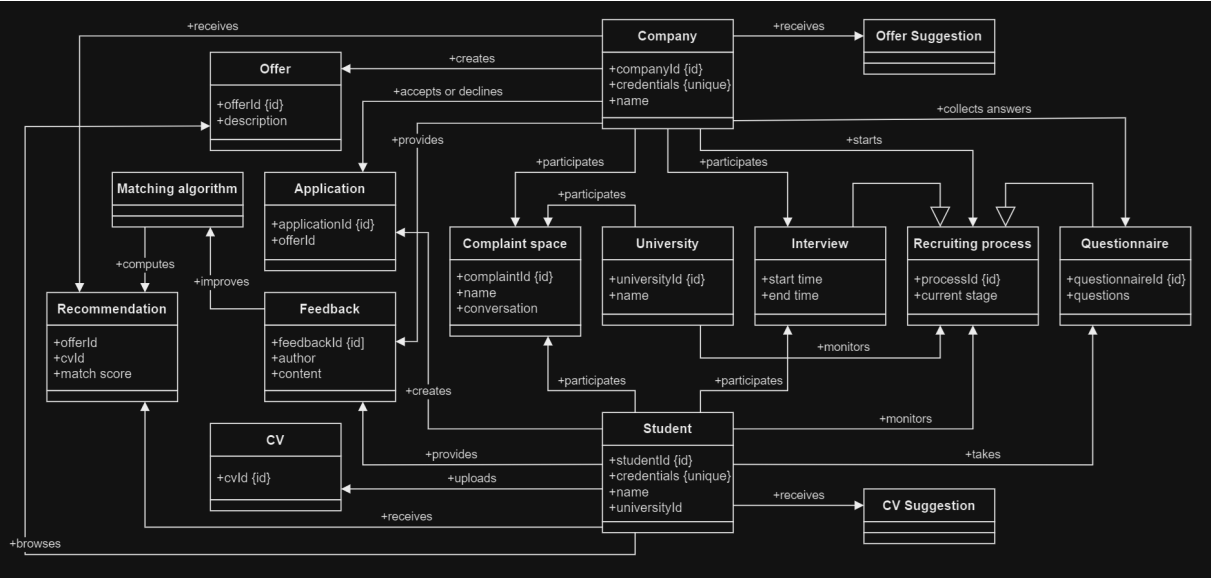


Figure 2.1: Domain class diagram showing the relationships between the main entities in the Students&Companies platform. The diagram illustrates the connections between Students, Companies, Universities, Internship Offers, and other key components of the system.

### 2.1.3. State diagram

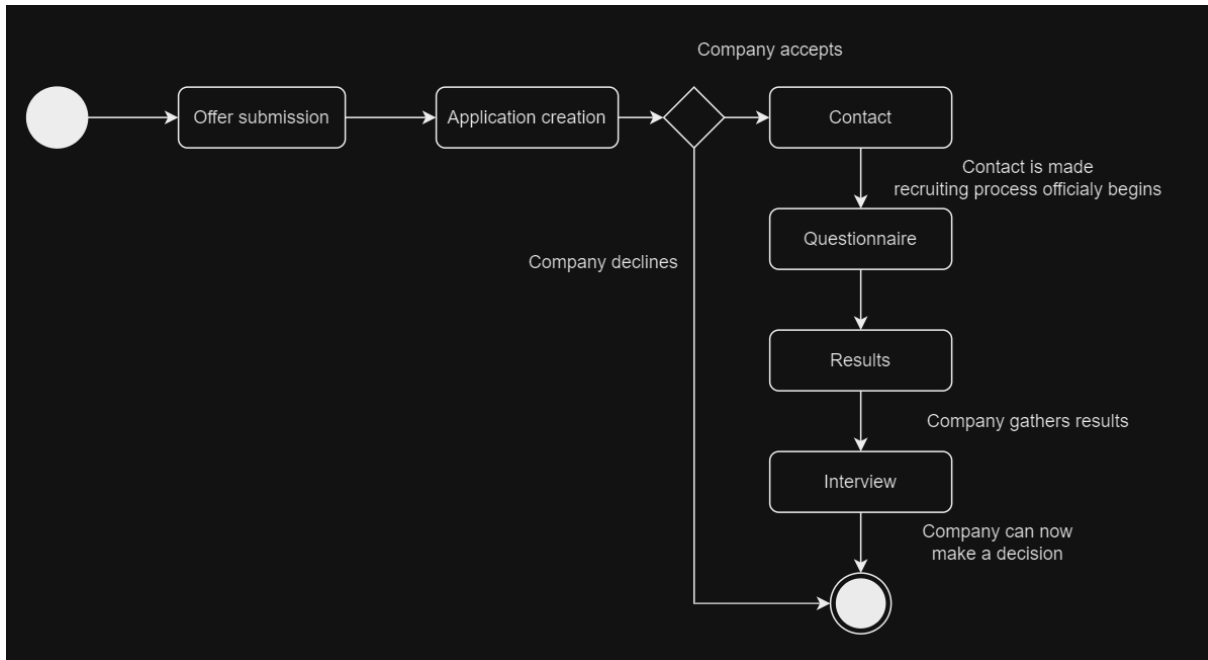


Figure 2.2: State diagram showing the recruiting process between the student and the company

## 2.2. Product functions

**Sign-up and login:** Sign-up is available to all users who want to use the platform. When a new user opens the platform, they can press the "Sign-up" button and will be redirected to a page where they can enter credentials (for example with e-mail or google account) and customise their username, password, and additional account settings. Once an account is created by a user, it can be accessed by pressing the login button and entering the credentials provided during the account creation.

**CV upload:** The students can upload an already existing CV on the platform, provide information about their education, experiences, skills and attitudes.

**Internship offer creation:** The companies can create internship offers by specifying the projects (application domain, tasks to be performed, relevant adopted technologies-if any, etc.), terms (paid or unpaid, tangible and intangible benefits, such as training, mentorships, etc.), profiles preferred for the offer (degree, field, etc.), as well as timing and location.

**CV/offer suggestions:** The students have access to tutorials and hints on CV creation.

The companies also have access to help for making project descriptions.

**Students can be proactive:** Students can manually go through the available internships to look for offers that may suit them. They can search with keywords about location, level of education, field, etc.

**Recommendations:** The platform uses statistical analysis and CV-offer similarity to find probable matches. Those probable matches are then periodically proposed to both the students and the companies. The recommendations can be accepted or dismissed by both parties. The platform collects various kinds of information during this process based for example on acceptances and refusals to feed and improve the statistical analysis. Users can also provide feedback which will also be studied by the platform to further improve their recommendation system.

**Support for selection process:** The platform supports the selection by helping manage interviews, questionnaires, etc. The platform also provides the companies with tools to compare how students performed in those tests to finalize the selections.

**Monitoring:** The platform provides spaces for communication, complaints, internship status monitoring, and matchmaking process monitoring to students, universities and companies.

### 2.3. User characteristics

**Student:** The student is able to register and login to the platform, to upload or create a CV and to browse the internship offers to find one that matches their expectations. Offers can also be recommended to the user. Once they are accepted by a company they can participate in the selection process and monitor it on the platform.

**Companies:** The companies can post offers which will be advertised to students. Students will contact them and they will also receive recommendations. Should they accept a recommendation, the company can contact them and go through the recruiting process on the platform.

**Universities:** They can and are advised to monitor the recruiting process and the internships themselves. They have tools to communicate and see complaints.

## 2.4. Assumptions, dependencies and constraints

### 2.4.1. Regulatory policies:

User personal information is only used for login. Mail boxes or google accounts won't be used for commercial purposes.

### 2.4.2. Domain Assumptions:

- [D1] Users must have a reliable internet connection
- [D2] User personal information must be correct
- [D3] Student CVs must correspond to their skills and experiences
- [D4] Company posts must reflect what they truly offer in their projects
- [D5] Universities must monitor the situation of internships
- [D6] User feedback must be truthful



## 3 | Specific Requirements

### 3.1. External Interface Requirements

#### 3.1.1. User Interfaces

The platform should provide intuitive and accessible user interfaces tailored to the three main user groups : Students, Companies and Universities

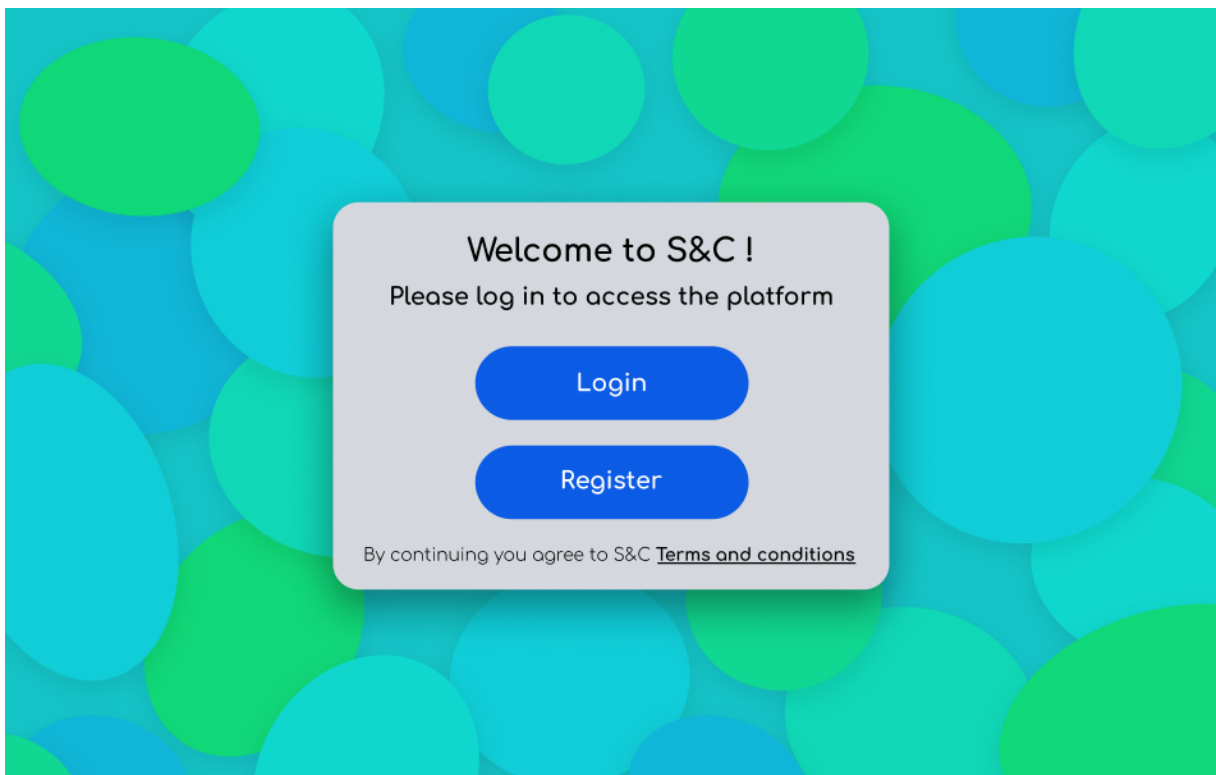


Figure 3.1: Login Interface

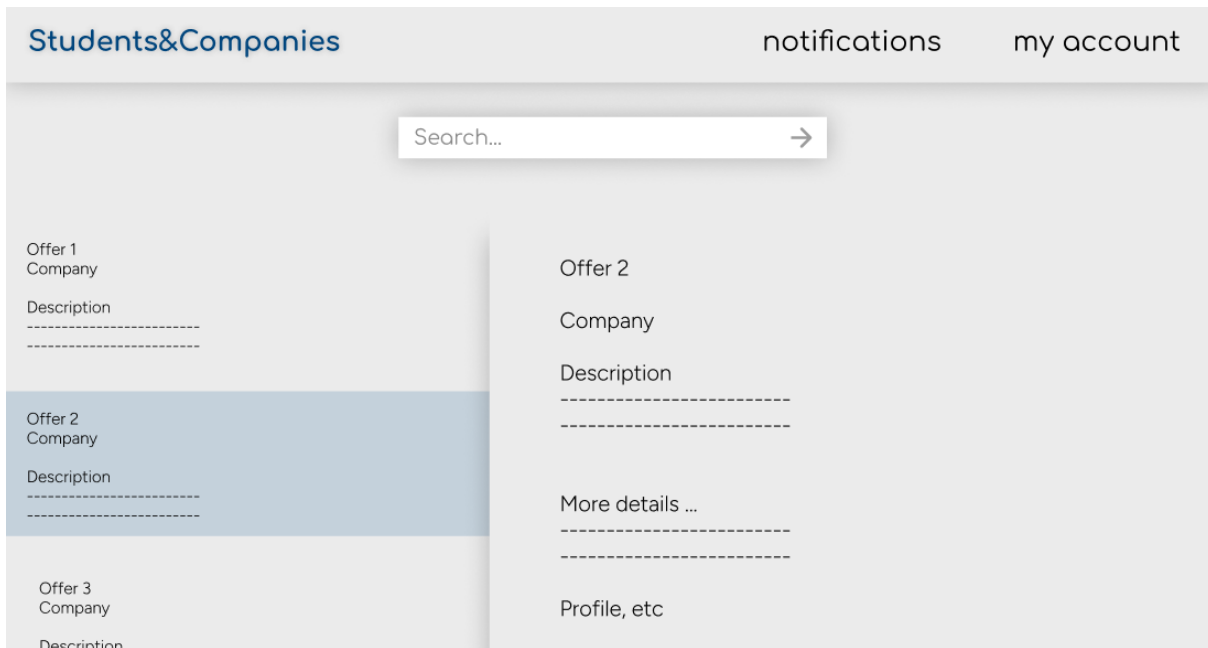


Figure 3.2: Browsing Interface

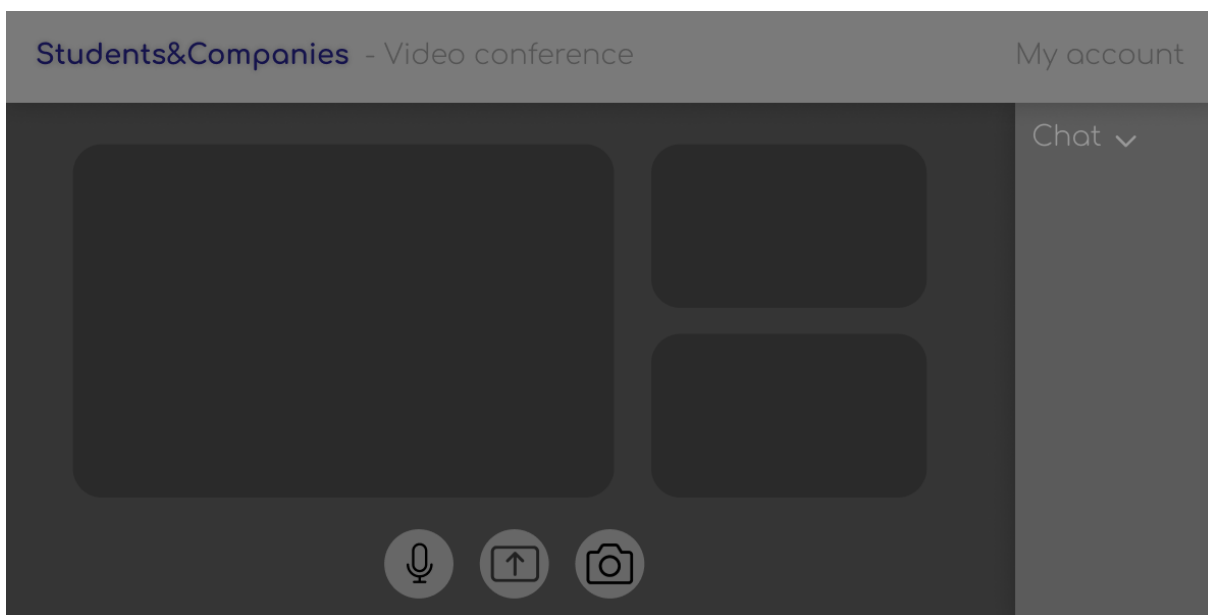


Figure 3.3: Interview Interface

**Student Interface:**

- Allows students to create accounts, upload CVs, search for offers by keywords and browse internship offers
- Displays personalized suggestions to improve CVs and provides recommendations

for matching internships

- Offers a dashboard to track applications, manage interviews and communicate with Companies
- Includes a complaints section for reporting issues

#### **Company Interface:**

- Enables companies to create and publish detailed internship offers
- Offers tools for creating assessment tests and managing candidate evaluations, interviews and recruitment processes
- Displays suggestions to improve internship descriptions and recommendations for suitable candidates
- Includes a complaints section for reporting issues

#### **University Interface:**

- Provides tools to monitor ongoing internships and address complaints from both students and companies
- Displays internship statuses, including student progress and company compliance

### **3.1.2. Hardware Interfaces**

Our platform is a web app, as a consequence, it does not require any specific hardware interface except for computer and any other device with web browser.

### **3.1.3. Software Interfaces**

#### **The system will integrate with:**

- Email services: For sending notifications, such as interview schedules and application updates
- Third-party APIs (University Internal Online Service): For university background checks
- External database systems:
  - Relational database (e.g. cloud-hosted MySQL) to store user information, internship offers and feedback

- NoSQL database (e.g. MongoDB Atlas) for storing unstructured data, such as CSV files or applications logs

### 3.1.4. Communication Interfaces

The system will use the following communication protocols:

#### 1. HTTPS

- For secure communication between the platform and its users
- HTTPS ensures data security by encrypting sensitive information, such as login credentials and feedback submissions
- This guarantees data privacy and integrity during communication

#### 2. RESTful APIs

- For interaction between frontend (user interface) and backend services (server logic)
- Enables structured data exchange and facilitates integration with third-party services like email systems or university APIs
- This ensures scalability and flexibility for future enhancements

#### 3. WebSocket

- For real-time communication between the platform and its users, such as real-time updates on application statuses, interviews schedules, feedback and notifications

## 3.2. 3.2 Functional Requirements

### Sign up and log in

- [R1] The system allows users (students, companies and universities) to register by providing their personal information (full name, organization etc), a valid email address and a password
- [R2] The system allows registered users to log in and access the platform's features based on their roles

### Internship Offers Management

- [R3] The system allows companies to create, modify and delete internship offers

- **[R4]** The system allows companies to manage the status of internship offers (e.g., open, closed, completed)
- **[R5]** The system allows students to browse and filter internship offers based on criteria like location, field, required skills and degree level
- **[R6]** The system notifies students about new internship offers matching their profile and preference

### **CV and Application Management**

- **[R7]** The system allows students to upload CV on the platform
- **[R8]** The system allows students to update their CVs and keep their profiles up to date
- **[R9]** The system allows students to apply to internship offers directly from the platform by attaching their CVs and adding optional documents (e.g. cover letters)
- **[R10]** The system allows companies to view, evaluate and shortlist applications received for their internship offers

### **Recruitment Process Management**

- **[R11]** The system allows companies to initiate recruitment process that may include tests, interviews or both
- **[R12]** The system allows companies to manage recruitment stages, including scheduling interviews, assigning tests and evaluating results
- **[R13]** The system provides tools for companies to evaluate, score and rank candidates based on test results, interview feedback and overall suitability
- **[R14]** The system notifies students of updates related to their recruitment status (e.g., rejected, hired, shortlisted)

### **Suggestions**

- **[R15]** The system analyzes student CVs and internship offers to generate actionable suggestions for both student and company
- **[R16]** The system sends personalized suggestions to students to improve their CVs, profiles, cover letters or applications
- **[R17]** The system sends personalized suggestions to companies to improve their internship offers or recruitment strategies

### Matching and Recommendations

- [R18] The system allows students to provide feedback on their application and recruitment experiences
- [R19] The system allows companies to provide feedback on candidates they evaluated and recruitment experiences
- [R20] The system uses a matching algorithm to analyze student CVs, internship offers and user feedback to generate recommendations
- [R21] The system sends recommendations to both students and companies when a match found

### Complaints and Monitoring

- [R22] The system allows students and companies to submit complaints about any issues related to the internship or recruitment process
- [R23] The system allows universities to monitor and resolve complaints submitted by students or companies
- [R24] The system notifies universities of unresolved issues or complaints requiring attention

### Notifications and Real-time Updates

- [R25] The system notifies students of application deadlines, interview schedules, test results and application status updates in real time
- [R26] The system notifies companies of updates on candidate applications and recruitment stages

### 3.2.1. Use cases diagram

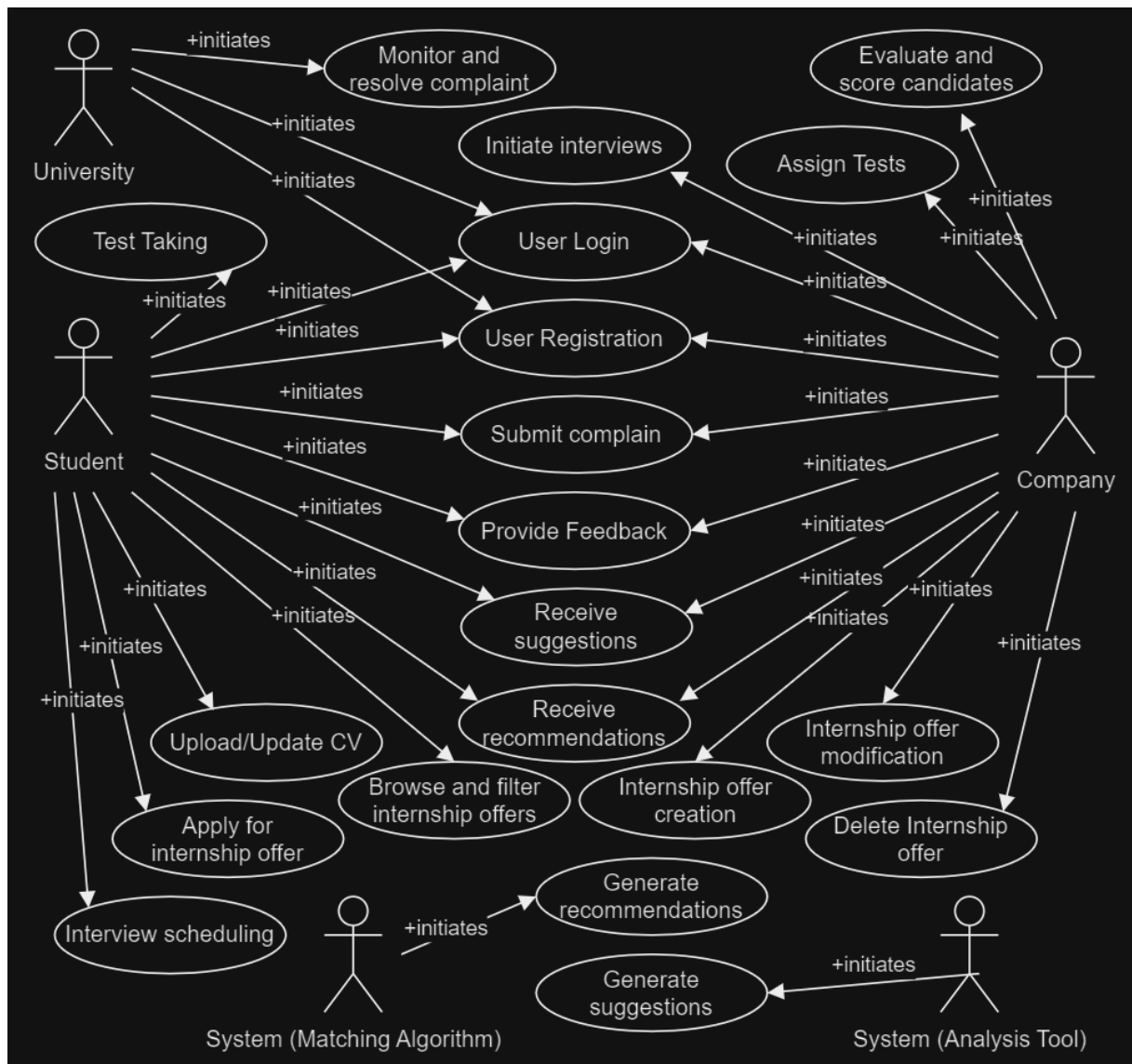


Figure 3.4: Use cases diagram

### 3.2.2. Use cases

#### [UC1] User Registration

Name	User Registration
Actors	User (Student, Company, University)
Entry Condition	The user has accessed the registration page of the platform
Event Flow	<ul style="list-style-type: none"><li>• The user clicks the “Register” button</li><li>• The system displays a registration form</li><li>• The user fills in the required fields (e.g., full name, email, password)</li><li>• The user submits the form</li><li>• The system validates the data and creates an account</li><li>• The system displays the login view</li></ul>
Exit Condition	The user account is successfully registered
Exception	If the email already exists, the system notifies the user and suggests logging in instead



**[UC2] User Login**

Name	User Login
Actors	User (Student, Company, University)
Entry Condition	The user has accessed the registration page of the platform
Event Flow	<ul style="list-style-type: none"><li>• The user enters their email and password</li><li>• The user clicks the “Login” button</li><li>• The system verifies the credentials</li><li>• The user is redirected to their dashboard</li></ul>
Exit Condition	The user is successfully logged in
Exception	If credentials are incorrect, the system displays an error message

**[UC3] Create Internship Offer**

Name	Internship offer creation
Actors	Company
Entry Condition	The company is logged in and has accessed the "Create Offer" section
Event Flow	<ul style="list-style-type: none"><li>• The company clicks “Add New Offer”</li><li>• The system displays a form to fill in offer details (e.g., title, description, required skills)</li><li>• The company completes the form and submits it</li><li>• The system validates the input and creates the offer</li><li>• The offer is displayed to students</li></ul>
Exit Condition	The offer is successfully published
Exception	If required fields are missing, the system prompts the company to complete the form

**[UC4] Modify Internship offer**

Name	Internship offer modification
Actors	Company
Entry Condition	The company is logged in and has accessed an existing internship offer they created
Event Flow	<ul style="list-style-type: none"><li>• The company selects an internship offer to modify</li><li>• The system displays the existing details of the offer</li><li>• The company updates the fields (e.g., title, description, required skills, duration)</li><li>• The company submits the form</li><li>• The system validates the input and saves the updates</li><li>• The modified offer is displayed to students</li></ul>
Exit Condition	The internship offer is successfully updated and published with the new details
Exception	If required fields are missing, the system prompts the company to complete the updates

**[UC5] Delete Internship offer**

Name	Delete Internship offer
Actors	Company
Entry Condition	The company is logged in and has accessed an existing internship offer they created
Event Flow	<ul style="list-style-type: none"><li>• The company selects an internship offer to delete</li><li>• The system prompts for confirmation of the deletion</li><li>• The company confirms the deletion</li><li>• The system deletes the offer from the platform and notifies any students who had applied to it</li></ul>
Exit Condition	The internship offer is successfully removed from the platform
Exception	If the deletion fails due to system errors, the platform notifies the company and retries

**[UC6] Browse and Filter Internship Offers**

Name	Browse and Filter Internship Offers
Actors	Student
Entry Condition	The student is logged in and has accessed the internship offers section
Event Flow	<ul style="list-style-type: none"><li>• The system displays a list of internship offers available on the platform</li><li>• The student uses filters (e.g., location, required skills, degree level) or keyword searching to refine the list</li><li>• The system updates the displayed offers based on the applied filters</li><li>• The student selects an offer to view its details</li><li>• The system displays the selected offer's full description</li></ul>
Exit Condition	The student has found one or more relevant internship offers
Exception	If no offers match the student's filters, the system notifies the student and suggests removing some filters

**[UC7] Upload/Update CV**

Name	Upload/Update CV
Actors	Student
Entry Condition	The student is logged in and has accessed their profile section
Event Flow	<ul style="list-style-type: none"><li>• The student clicks the “Upload CV” button</li><li>• The system prompts the student to select a file from their device</li><li>• The student uploads the file</li><li>• The system validates the file format and size</li><li>• The system saves the CV to the student’s profile</li></ul>
Exit Condition	The CV is successfully uploaded and visible on the student’s profile
Exception	If the file format is unsupported or the file exceeds size limits, the system rejects the upload and notifies the student

**[UC8] Apply for Internship Offer**

Name	Apply for Internship Offer
Actors	Student
Entry Condition	The student is logged in, has uploaded a CV and is viewing an internship offer
Event Flow	<ul style="list-style-type: none"><li>• The student selects an internship offer to apply for</li><li>• The system displays a prompt to attach a CV or optional documents (e.g., cover letters)</li><li>• The student attaches the required documents and submits the application</li><li>• The system validates the documents and saves the application</li><li>• system notifies the company about the new application</li><li>• The system sends a mail of confirmation of succesful application to student's email</li></ul>
Exit Condition	The application is successfully submitted to the company
Exception	If no CV is attached or the file format is invalid, the system notifies the student to correct the issue

**[UC9] Initiate Interviews**

Name	Initiate Interviews
Actors	Company
Entry Condition	The company has received applications for offers
Event Flow	<ul style="list-style-type: none"><li>• The company logs into the platform</li><li>• The company selects an internship offer</li><li>• The system displays a list of received applications for that offer</li><li>• The company selects a candidate from the list and click on "Add Interview"</li><li>• The system prompts the company to configure interview slots by selecting multiple possible dates, times, and formats (e.g., online or in-person)</li><li>• The company confirms the configuration</li><li>• The system notifies the selected candidate about the interview slots</li></ul>
Exit Condition	The interview schedule is successfully configured and the candidate is notified

**[UC10] Assign Tests**

Name	Assign Tests
Actors	Company
Entry Condition	The company has received applications for offers
Event Flow	<ul style="list-style-type: none"><li>• The company logs into the platform</li><li>• The company selects an internship offer</li><li>• The system displays a list of received applications for that offer</li><li>• The company selects a candidate from the list and click on "Add Test"</li><li>• The system prompts the company to assign a test by selecting it from a list of available tests or creating a new test</li><li>• The company confirms the configuration</li><li>• The system notifies the candidate about the assigned test, providing a link for access</li></ul>
Exit Condition	The test is successfully assigned to the candidates



**[UC11] Evaluate and Score Candidates**

Name	Evaluate and Score Candidates
Actors	Company
Entry Condition	Company is logged in and tests have been completed or interviews have been conducted
Event Flow	<ul style="list-style-type: none"> <li>• The system provides the company with access to test results and interview feedback</li> <li>• The company evaluates candidates based on the provided data</li> <li>• The company assigns scores to each candidate</li> <li>• The system updates each candidate's application status and rankings based on the scores</li> </ul>
Exit Condition	The evaluation is saved and candidates are ranked accordingly
Exception	If scores are not entered, the system flags the evaluation as incomplete

**[UC12] Provide Feedback**

Name	Provide Feedback
Actors	User (Company,Student)
Entry Condition	The recruitment process has been completed
Event Flow	<ul style="list-style-type: none"> <li>• The system prompts students and companies to provide feedback</li> <li>• The user fills in a feedback form with ratings and comments</li> <li>• The system validates and saves the feedback</li> </ul>
Exit Condition	The feedback is successfully recorded in the system

**[UC13] Generate Suggestions**

Name	Generate Suggestions
Actors	System (Analysis tool), User (Company, Student)
Entry Condition	Students have uploaded their CVs and companies have published their offers
Event Flow	<ul style="list-style-type: none"><li>• The system analyzes student CVs and internship offers</li><li>• The system identifies areas for improvement (e.g., CV content, job descriptions)</li><li>• The system generates personalized suggestions for students and companies</li></ul>
Exit Condition	Suggestions are generated and stored

**[UC14] Receive Suggestions**

Name	Receive Suggestions
Actors	User (Company, Student)
Entry Condition	Suggestions have been generated by the system
Event Flow	<ul style="list-style-type: none"><li>• The system notifies the user about the availability of suggestions</li><li>• The user views the suggestions on their dashboard</li><li>• The user accepts or ignores the suggestions</li></ul>
Exit Condition	The user has received and reviewed the suggestions

**[UC15] Generate Recommendations**

Name	Generate Recommendations
Actors	System (Matching algorithm), User (Company, Student)
Entry Condition	The system has access to CVs, offers and user feedback
Event Flow	<ul style="list-style-type: none"> <li>• The system analyzes CVs, offers, preferences and previous recruitment feedback</li> <li>• The system generates personalized recommendations for students and companies using the matching algorithm</li> <li>• The recommendations are saved in the system</li> </ul>
Exit Condition	Recommendations are successfully generated

**[UC16] Receive Recommendations**

Name	Receive Recommendations
Actors	User (Company, Student)
Entry Condition	Recommendations have been generated and sent to the respective users
Event Flow	<ul style="list-style-type: none"> <li>• The system notifies the user about the availability of recommendations</li> <li>• The user views the recommendations on their dashboard</li> <li>• The user accepts or declines the recommendations: 1) If accepted, the system updates the recommendation status to "accepted" for that user 2) If rejected, the system updates the status to "declined"</li> <li>• When both parties (student and company) accept a recommendation, the system establishes a "Contact" between them and triggers the recruitment process</li> </ul>
Exit Condition	The user has received and reviewed the recommendations

**[UC17] Submit Complaint**

Name	Submit Complaint
Actors	User (Company, Student)
Entry Condition	The user has an issue related to the internship or recruitment process and the user is logged in
Event Flow	<ul style="list-style-type: none"><li>• The user navigates to the complaints section</li><li>• The user fills in a complaint form, providing details of the issue</li><li>• The system saves the complaint and notifies the university</li></ul>
Exit Condition	The complaint is successfully submitted

**[UC18] Monitor and Resolve Complaint**

Name	Monitor and Resolve Complaint
Actors	University
Entry Condition	A complaint has been submitted by a student or company, and the university is logged in
Event Flow	<ul style="list-style-type: none"><li>• The university accesses the complaint management section</li><li>• The system displays the list of pending complaints</li><li>• The university selects a specific complaint</li><li>• The system displays all relevant details about the complaint</li><li>• The university reviews the complaint details and severity</li><li>• The university creates appropriate communication channels with involved parties</li><li>• The university documents the actions taken and resolution</li><li>• The system updates the complaint status</li><li>• The system notifies involved parties about the resolution</li></ul>
Exit Condition	A resolution is proposed
Exception	If the complaint needs further investigation, the university can keep the status as "in progress" and continue the resolution process

**[UC19] Interview scheduling**

Name	Interview scheduling
Actors	Student
Entry Condition	The student has received a notification about the interview invitation
Event Flow	<ul style="list-style-type: none"> <li>• The system notifies the student about the interview process and available slots</li> <li>• The student logs into the system and navigates to the interview scheduling section</li> <li>• The system displays the available interview slots</li> <li>• The student selects a preferred interview slot from the list</li> <li>• The system confirms the slot and updates the interview schedule for both the student and the company</li> <li>• If no preferred slot is available, the student clicks on the "Indicate Availability" button</li> <li>• The system prompts the student to specify their available dates and times</li> <li>• The system sends this information to the company for further action</li> </ul>
Exit Condition	The slot is scheduled or the student's availability has been communicated to the company for further scheduling
Exception	If the student does not respond to the notification or skips the steps, they are marked as "not participating"

**[UC20] Test taking**

Name	Test taking
Actors	Student
Entry Condition	The student has been notified about the recruitment process and is eligible to take a recruitment test
Event Flow	<ul style="list-style-type: none"><li>• The system notifies the student about the recruitment test and provides a link to access it</li><li>• The student logs into the system and selects the recruitment test link</li><li>• The system presents the test interface (e.g., coding problems, multiple-choice questions)</li><li>• The student completes the test within the provided time frame</li><li>• The student submits the test</li><li>• The system validates the submission, records the results and updates the recruitment progress</li></ul>
Exit Condition	The student has successfully submitted the test

### 3.2.3. Sequence diagrams

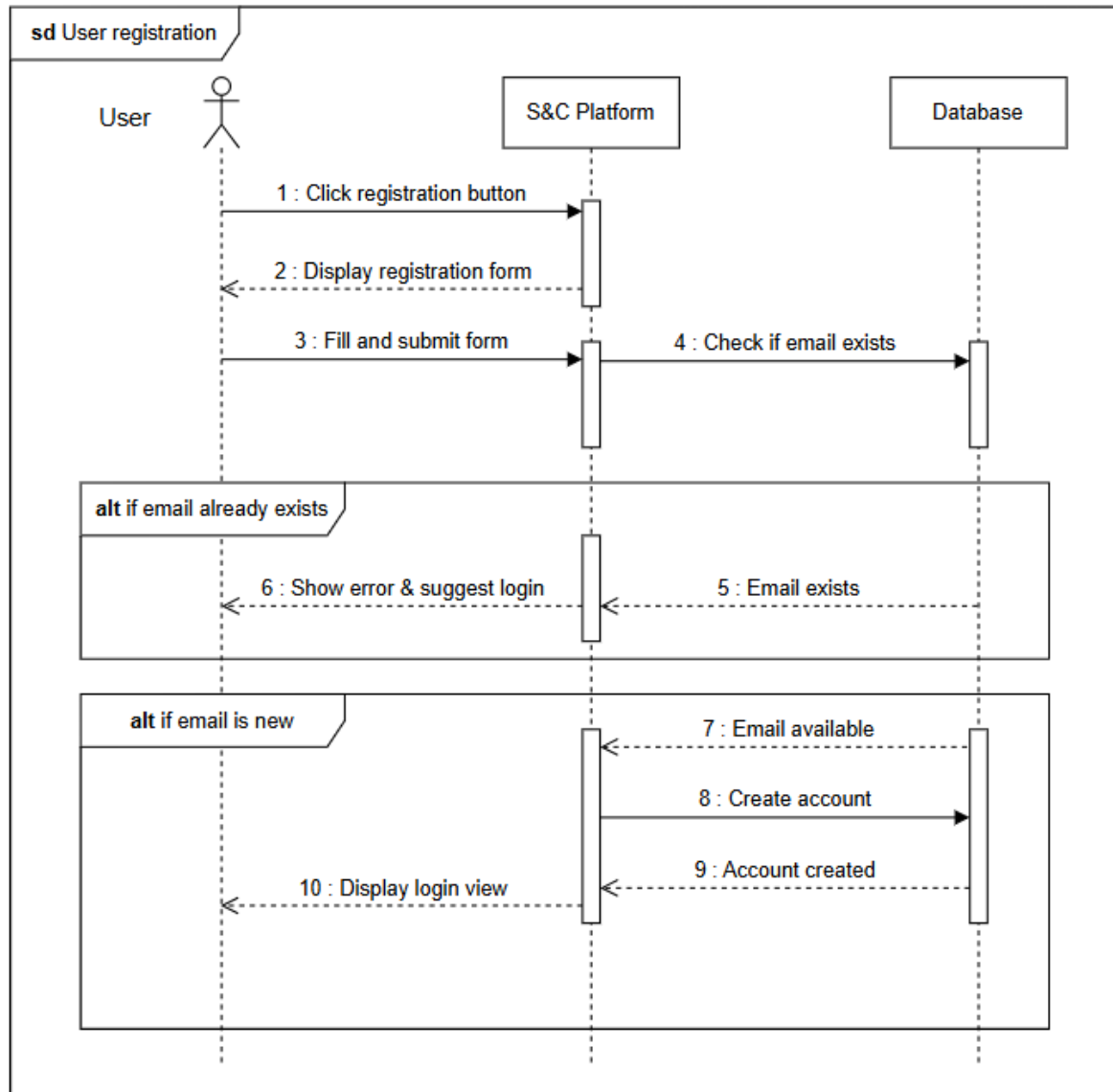


Figure 3.5: [UC1] User Registration



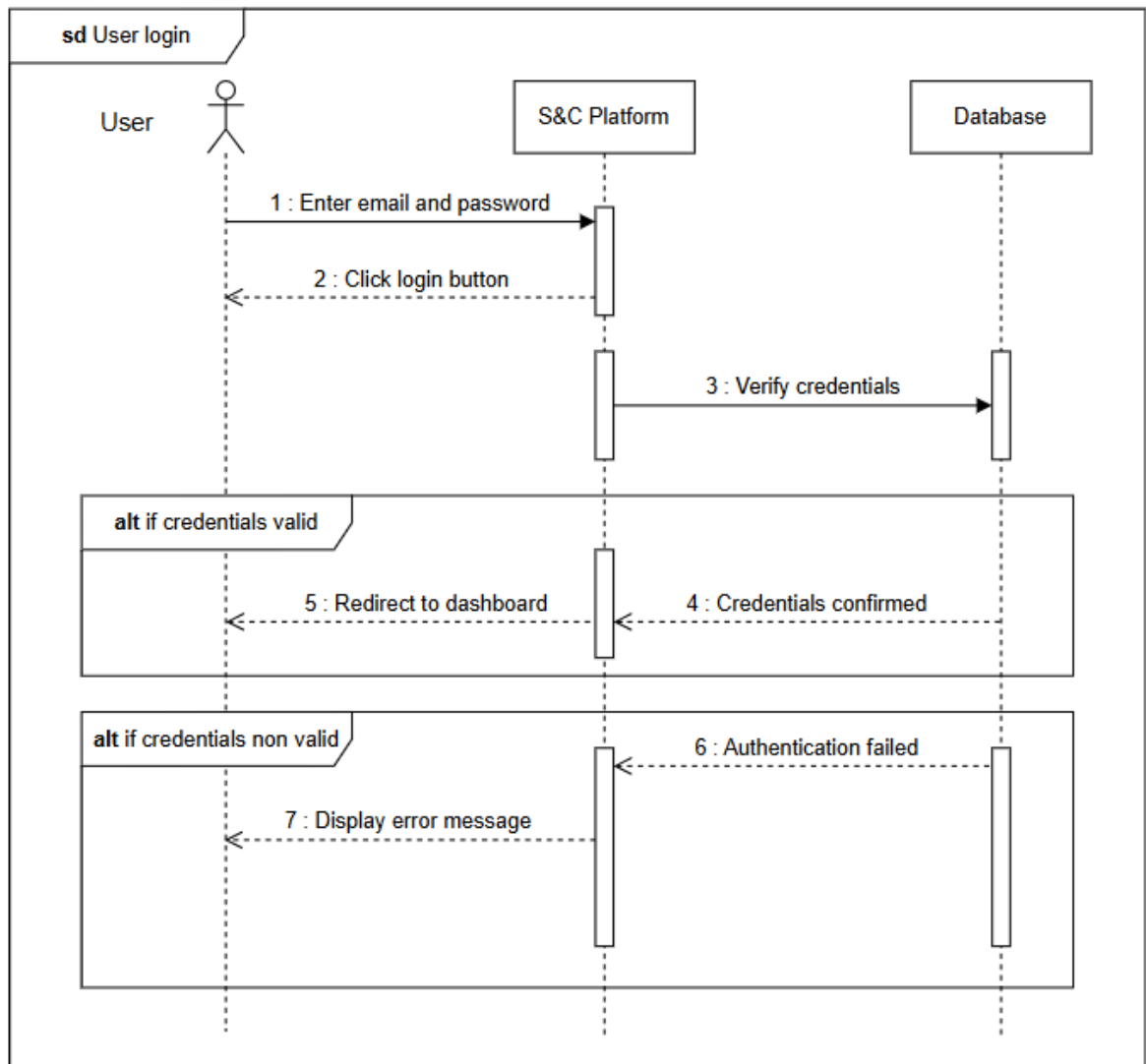


Figure 3.6: [UC2] User Login

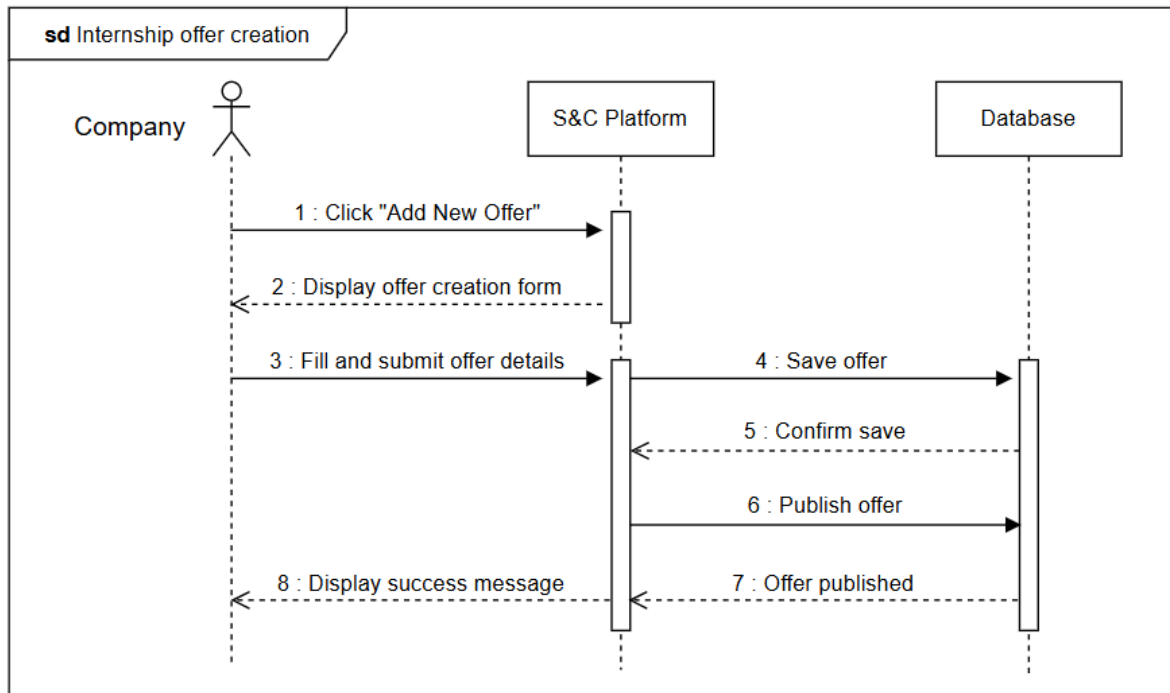


Figure 3.7: [UC3] Internship offer creation

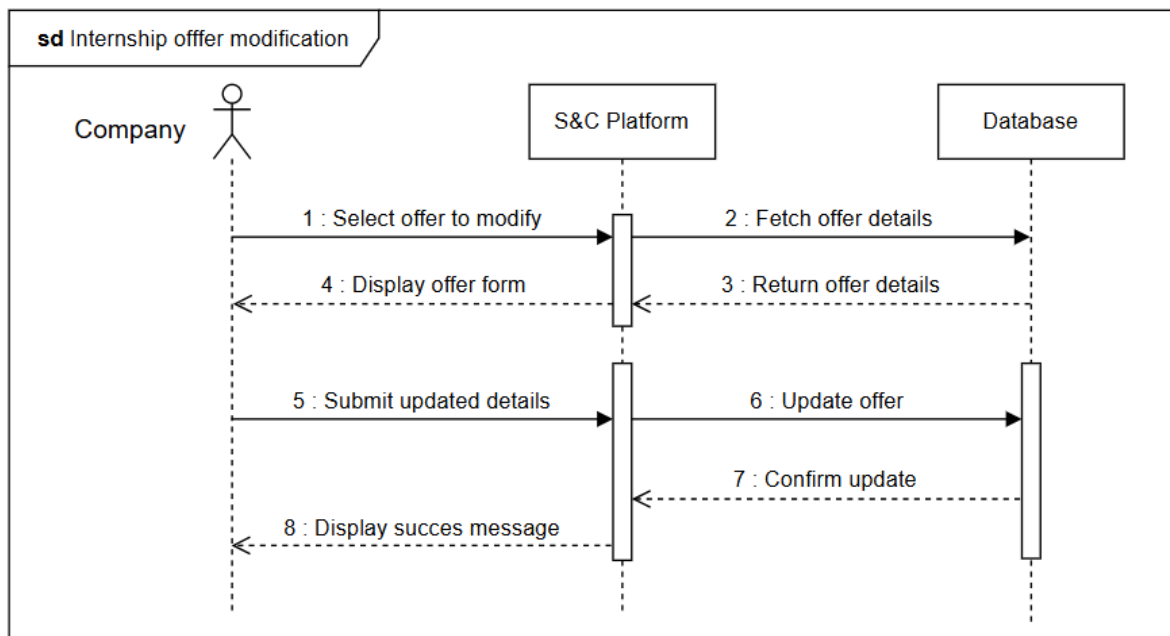


Figure 3.8: [UC4] Internship offer modification

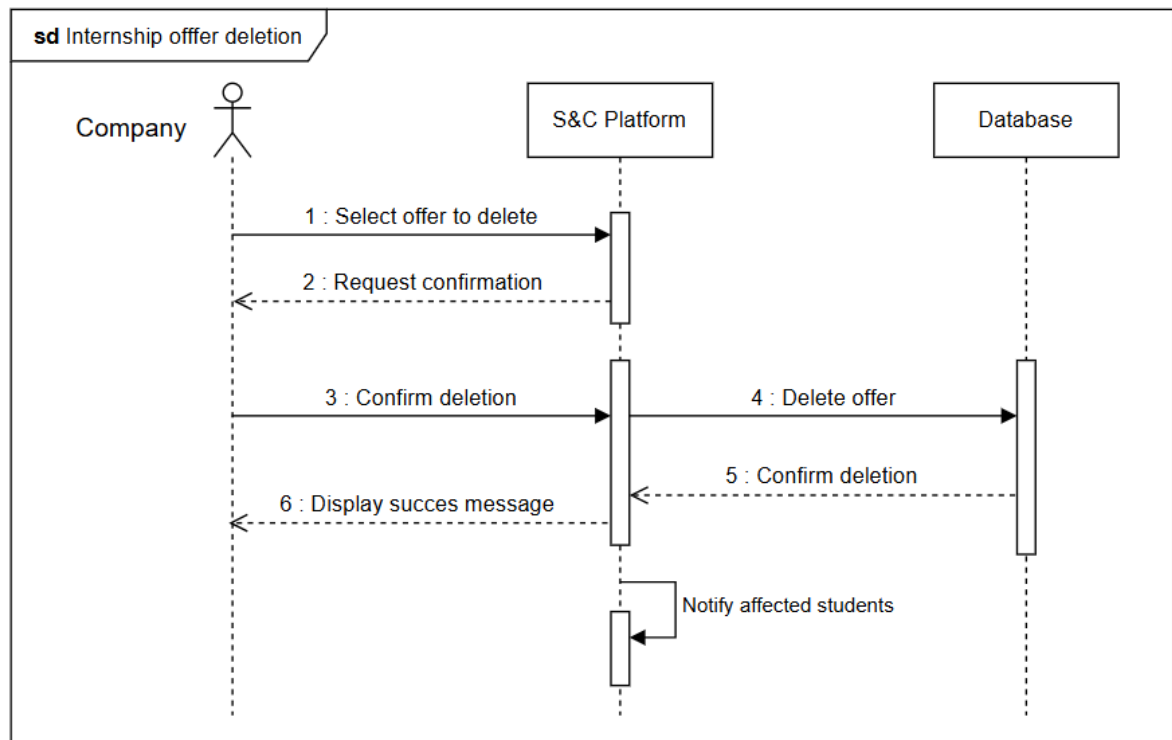


Figure 3.9: [UC5] Internship offer deletion

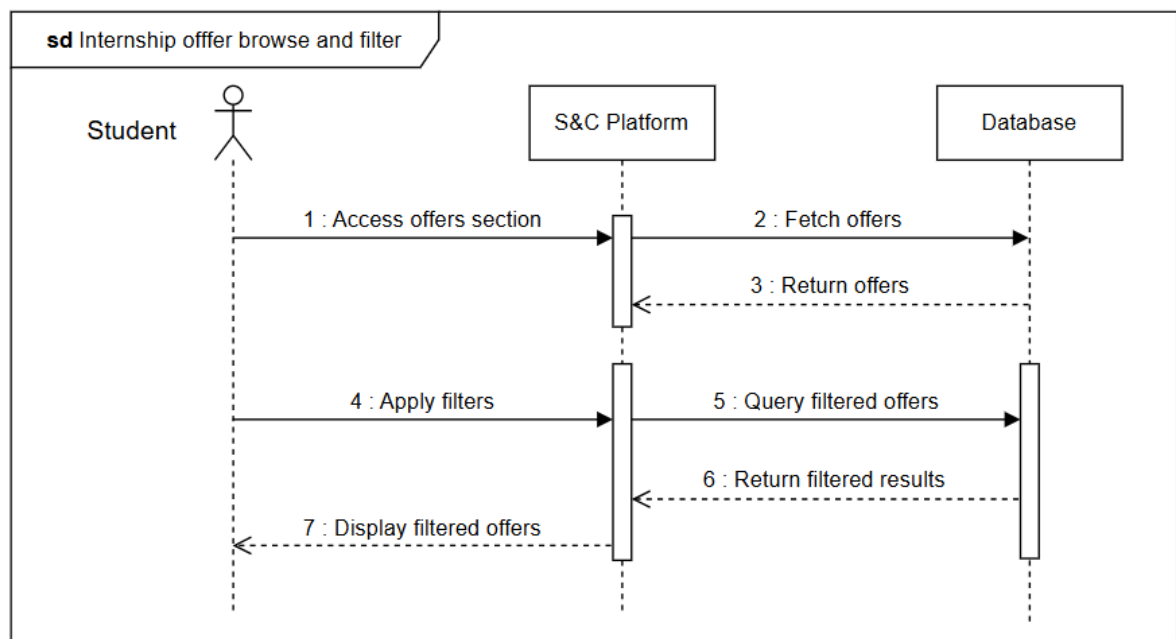


Figure 3.10: [UC6] Internship offer browse and filter

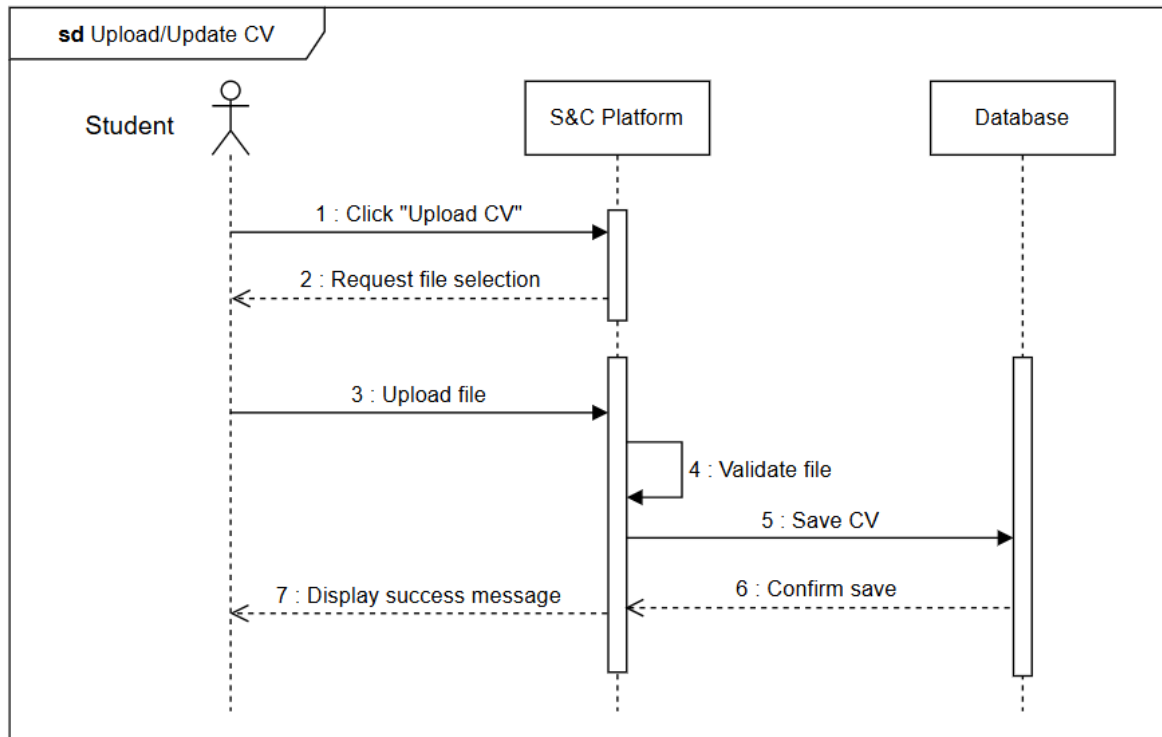


Figure 3.11: [UC7] Upload/Update CV

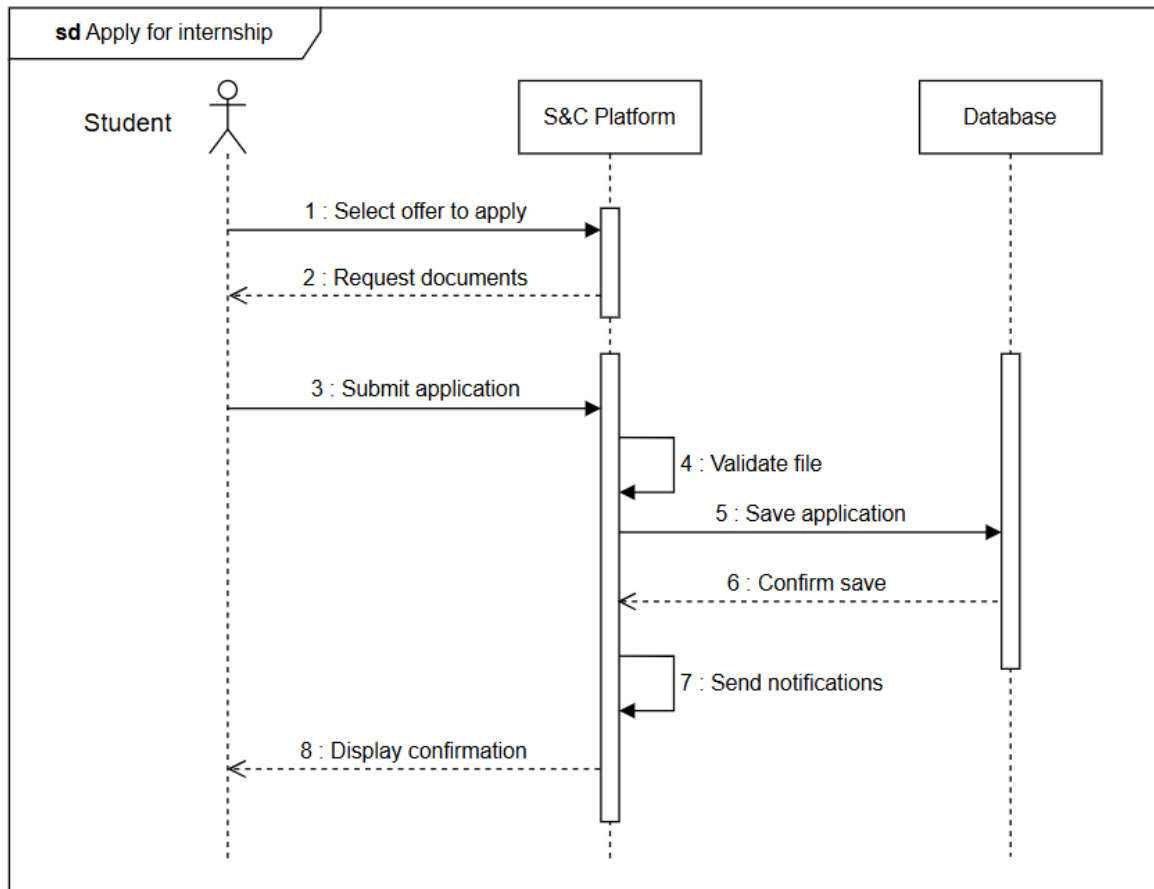


Figure 3.12: [UC8] Apply for intership

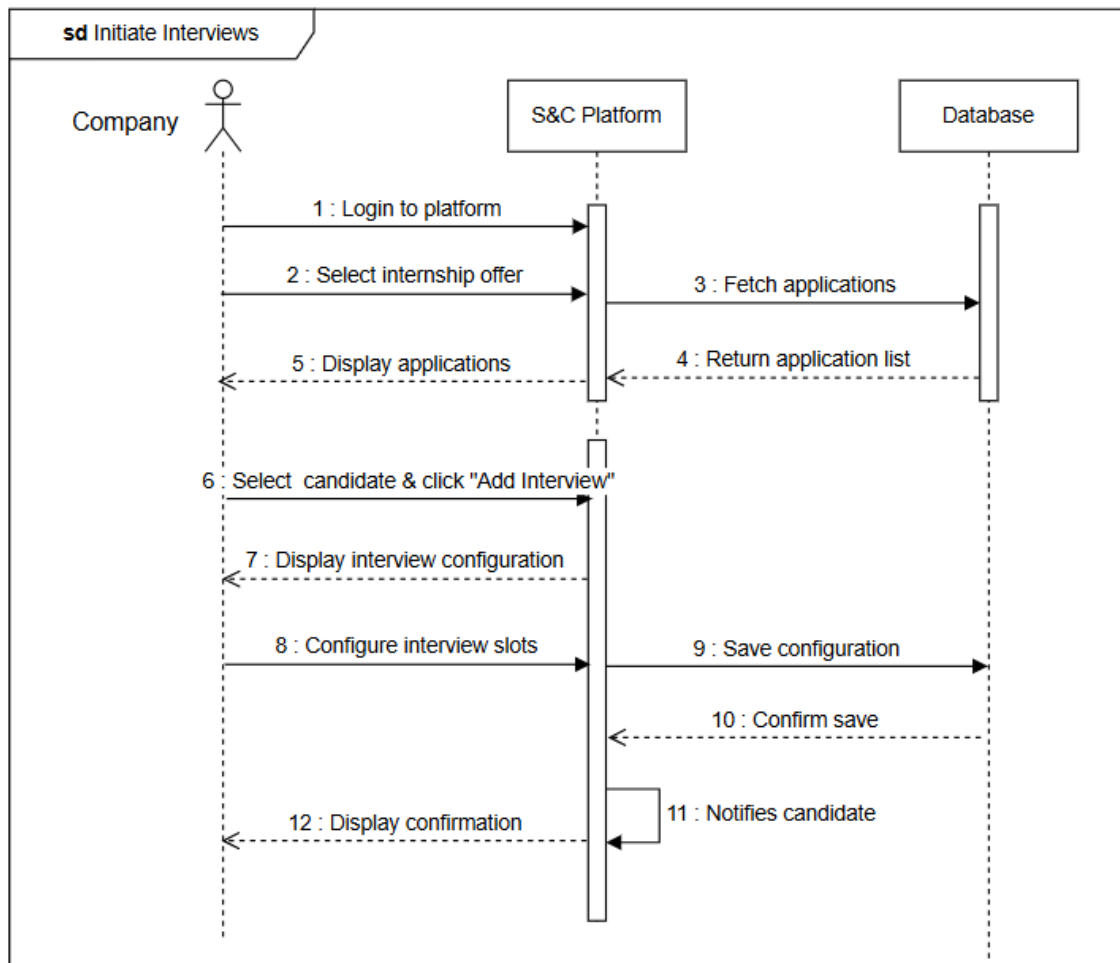


Figure 3.13: [UC9] Initiate interviews

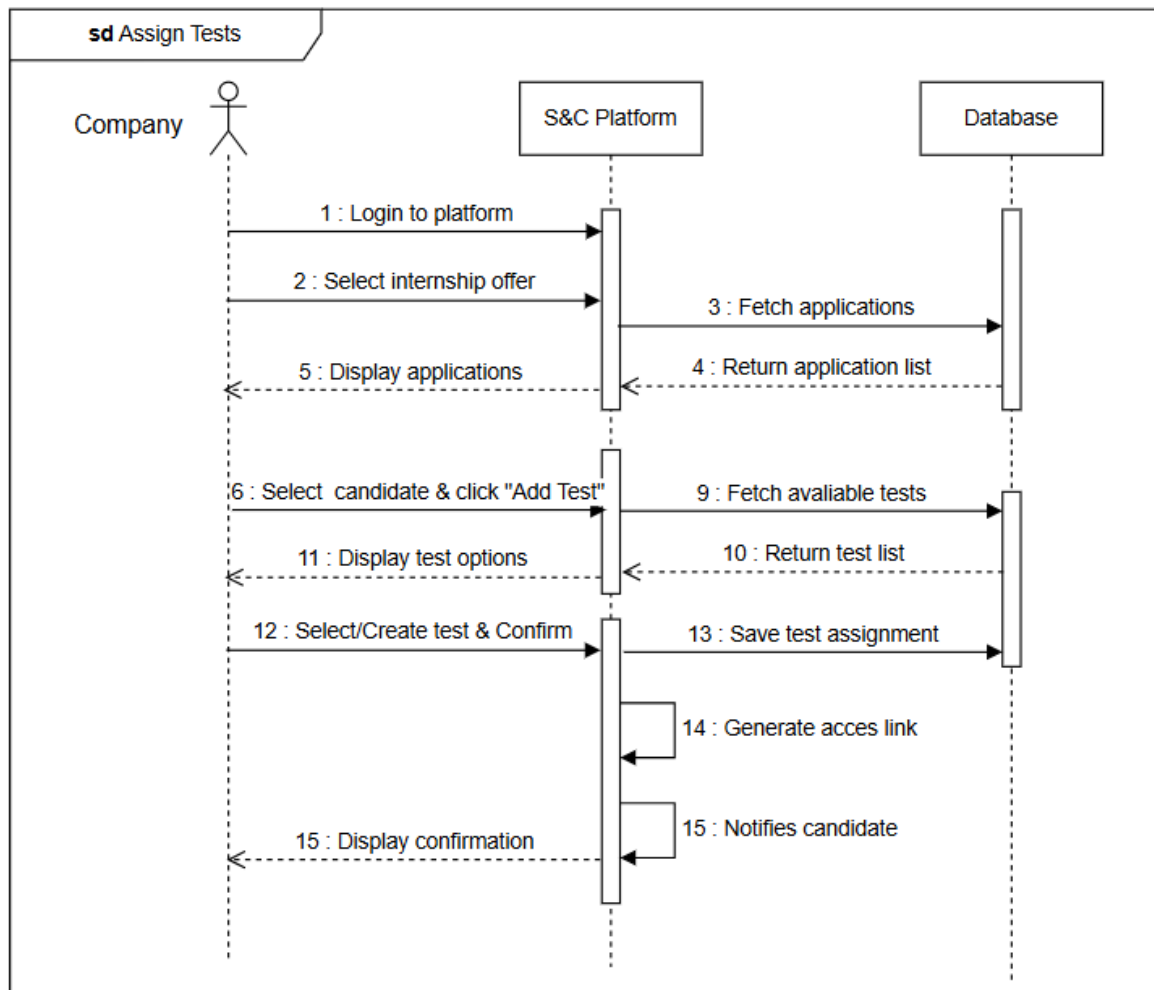


Figure 3.14: [UC10] Assign Tests

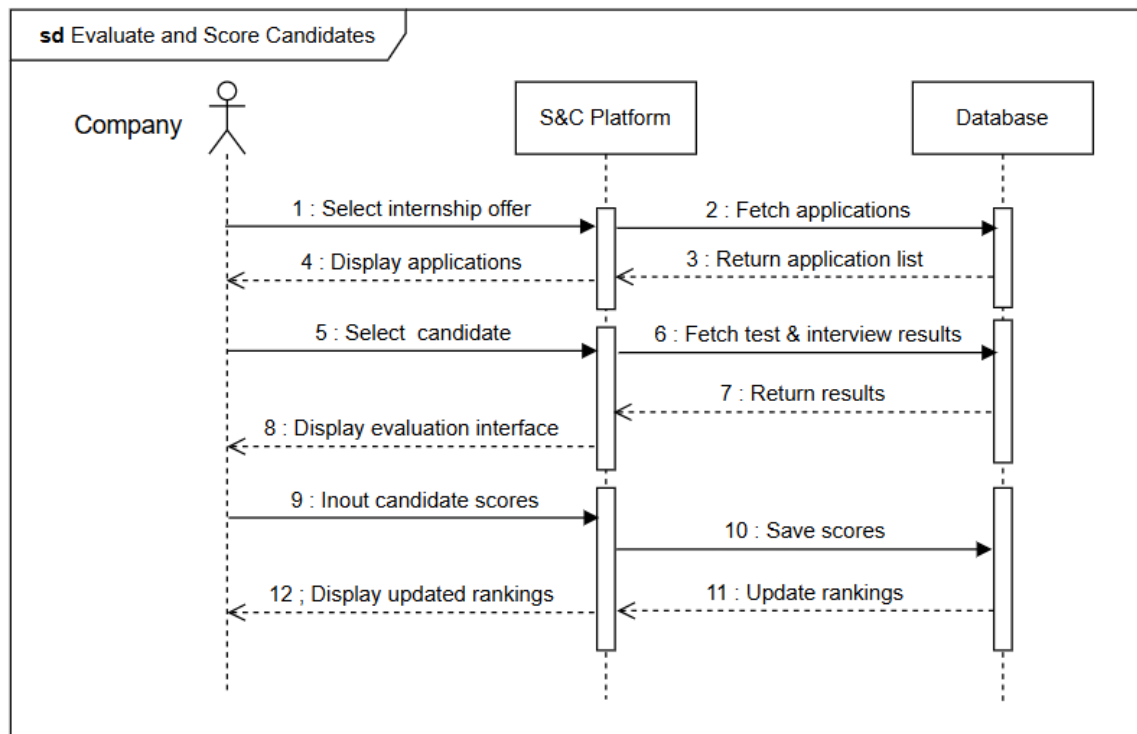


Figure 3.15: [UC11] Evaluate and Score Candidates



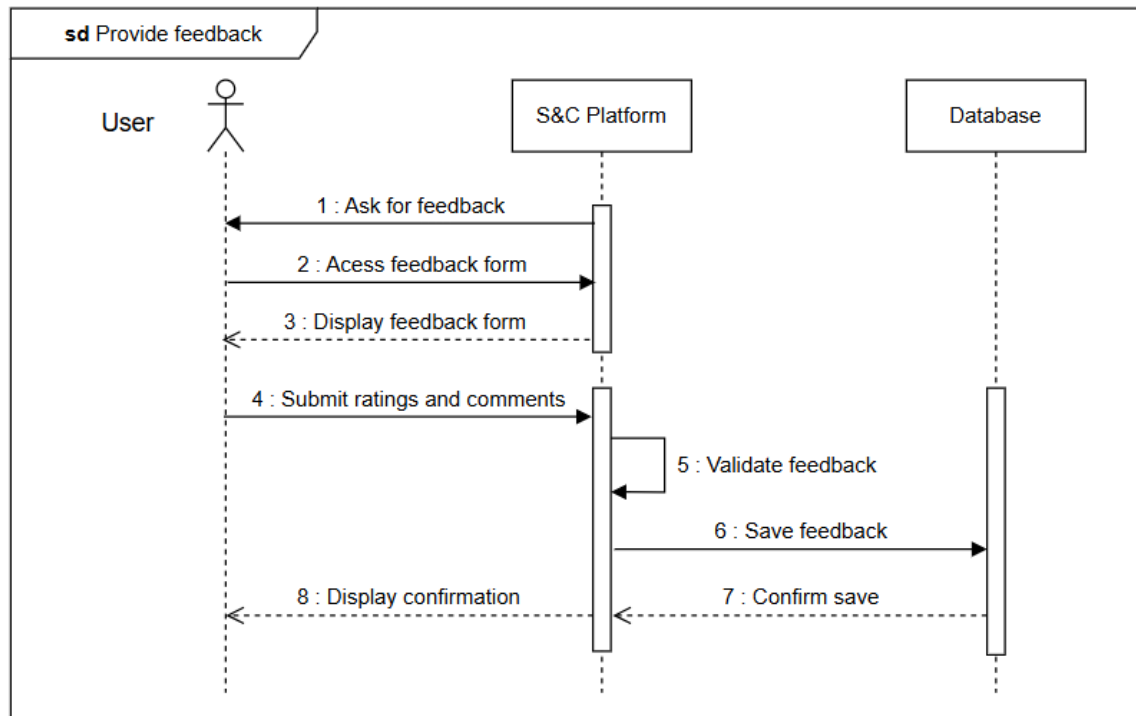


Figure 3.16: [UC12] Provide Feedback

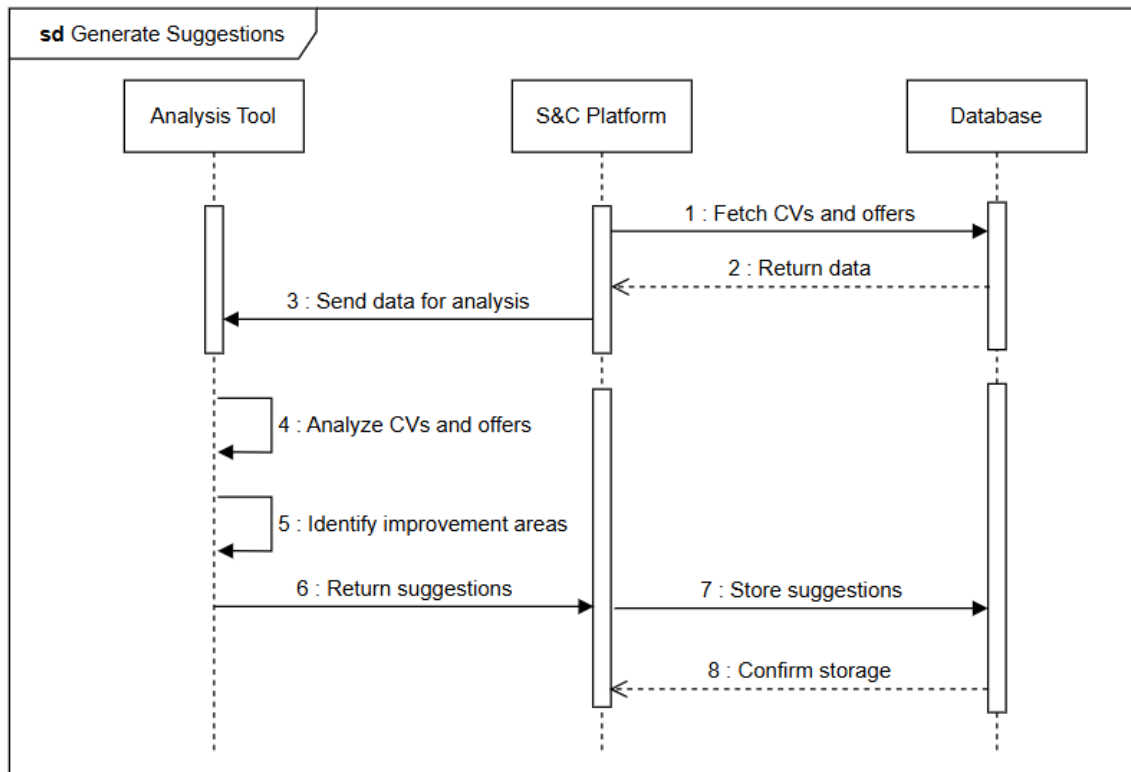


Figure 3.17: [UC13] Generate Suggestions

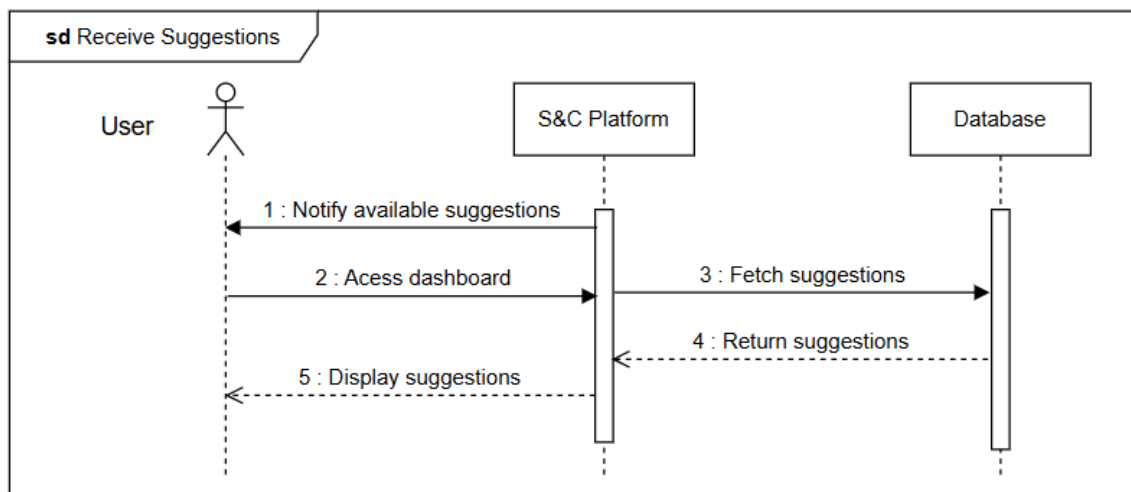


Figure 3.18: [UC14] Receive Suggestions

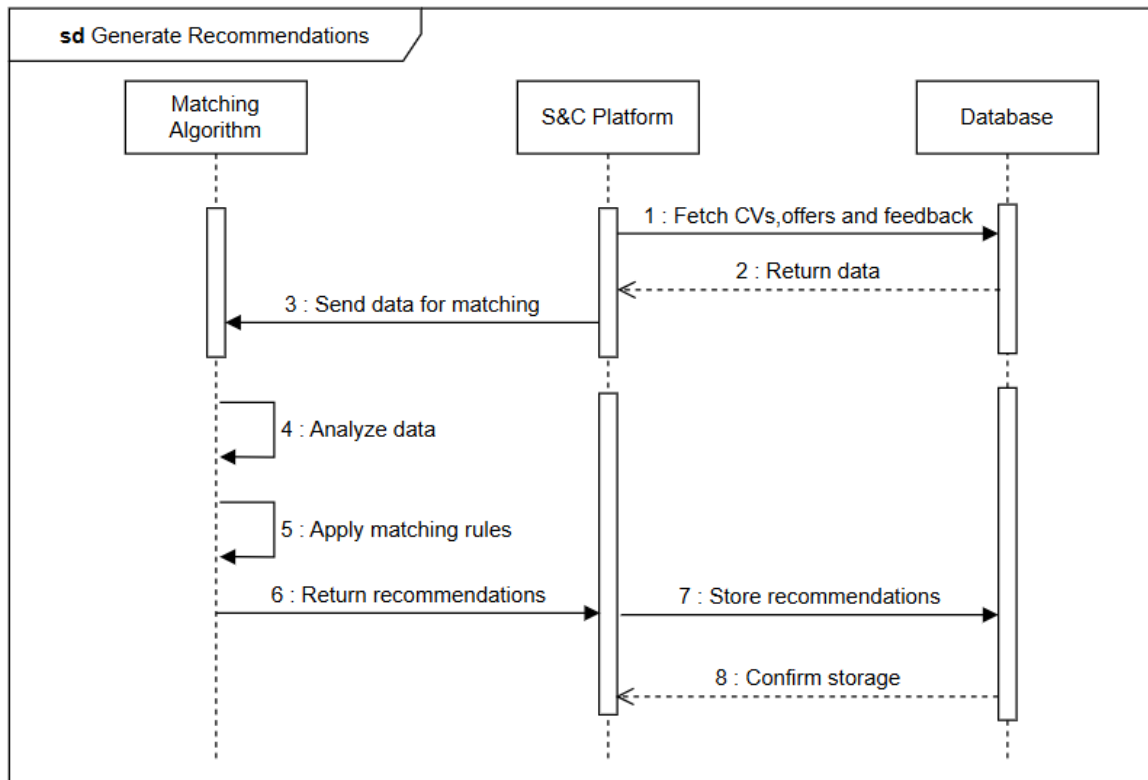


Figure 3.19: [UC15] Generate Recommendations

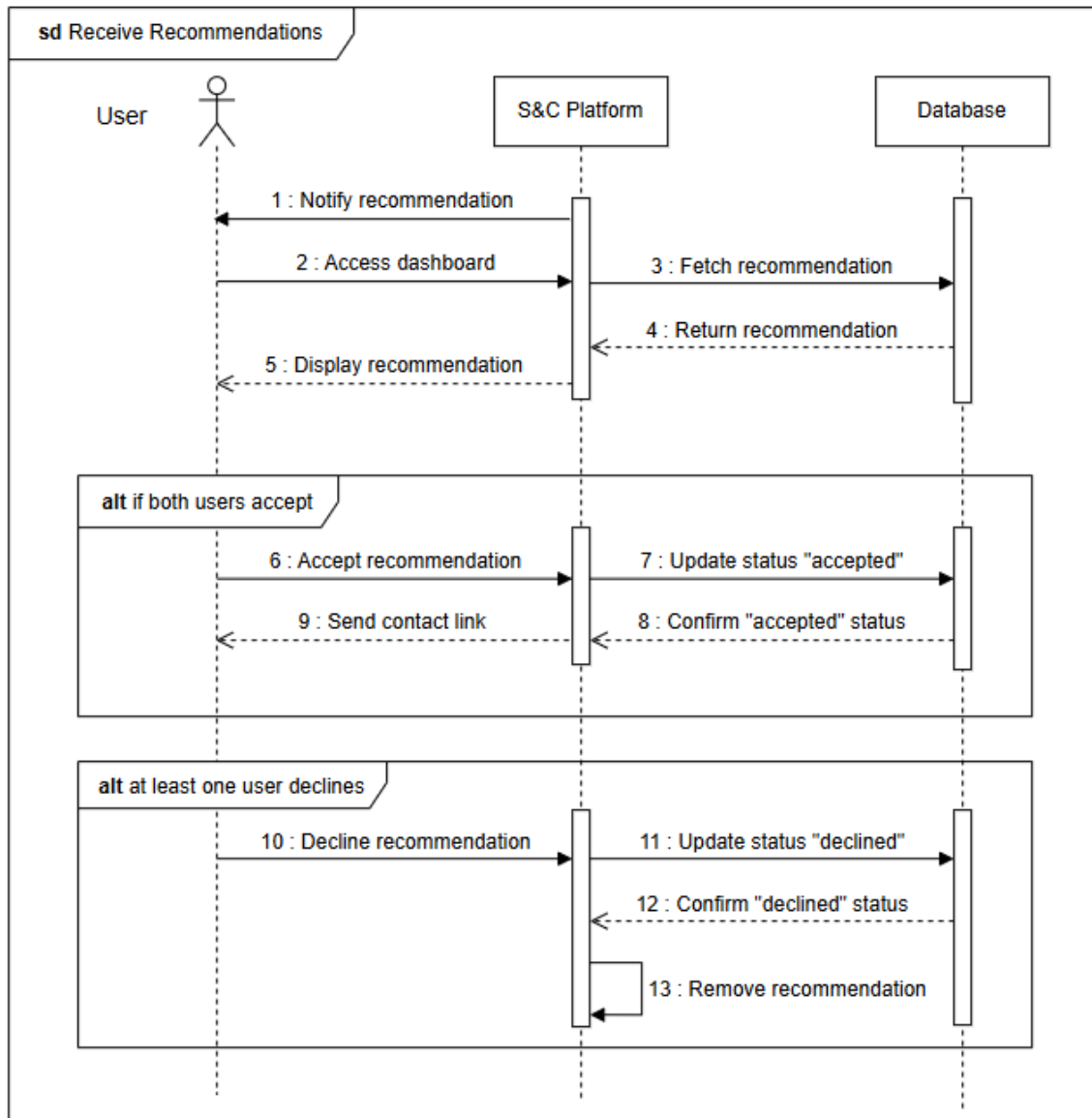


Figure 3.20: [UC16] Receive Recommendations

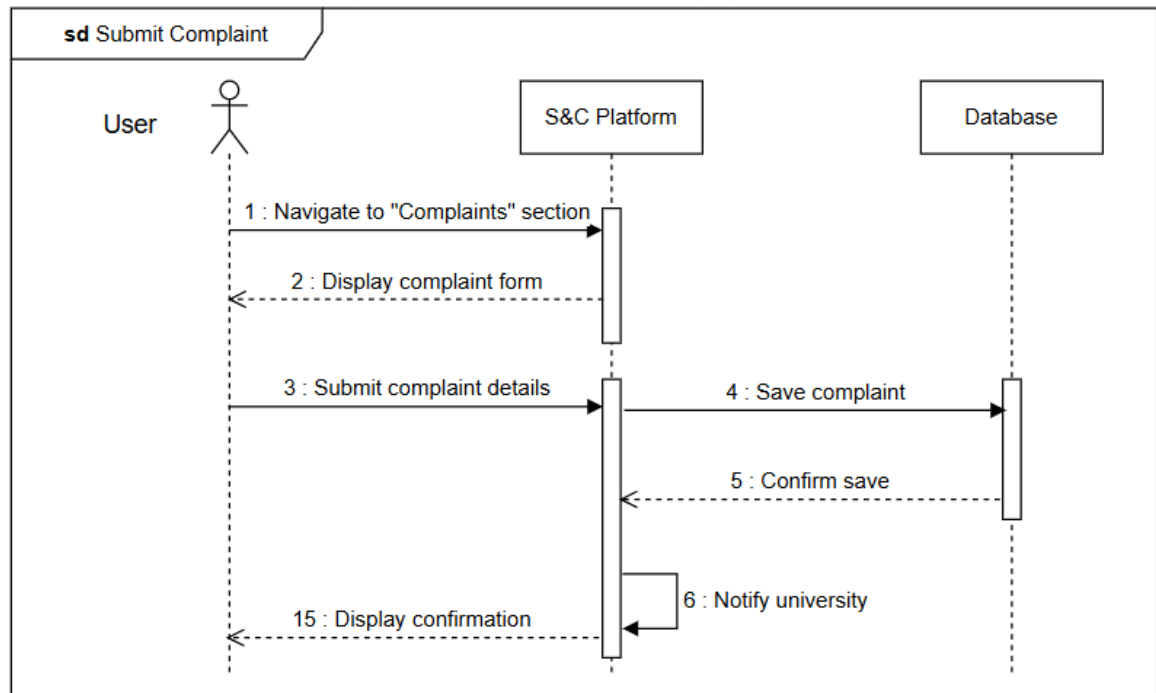


Figure 3.21: [UC17] Submit Complaints

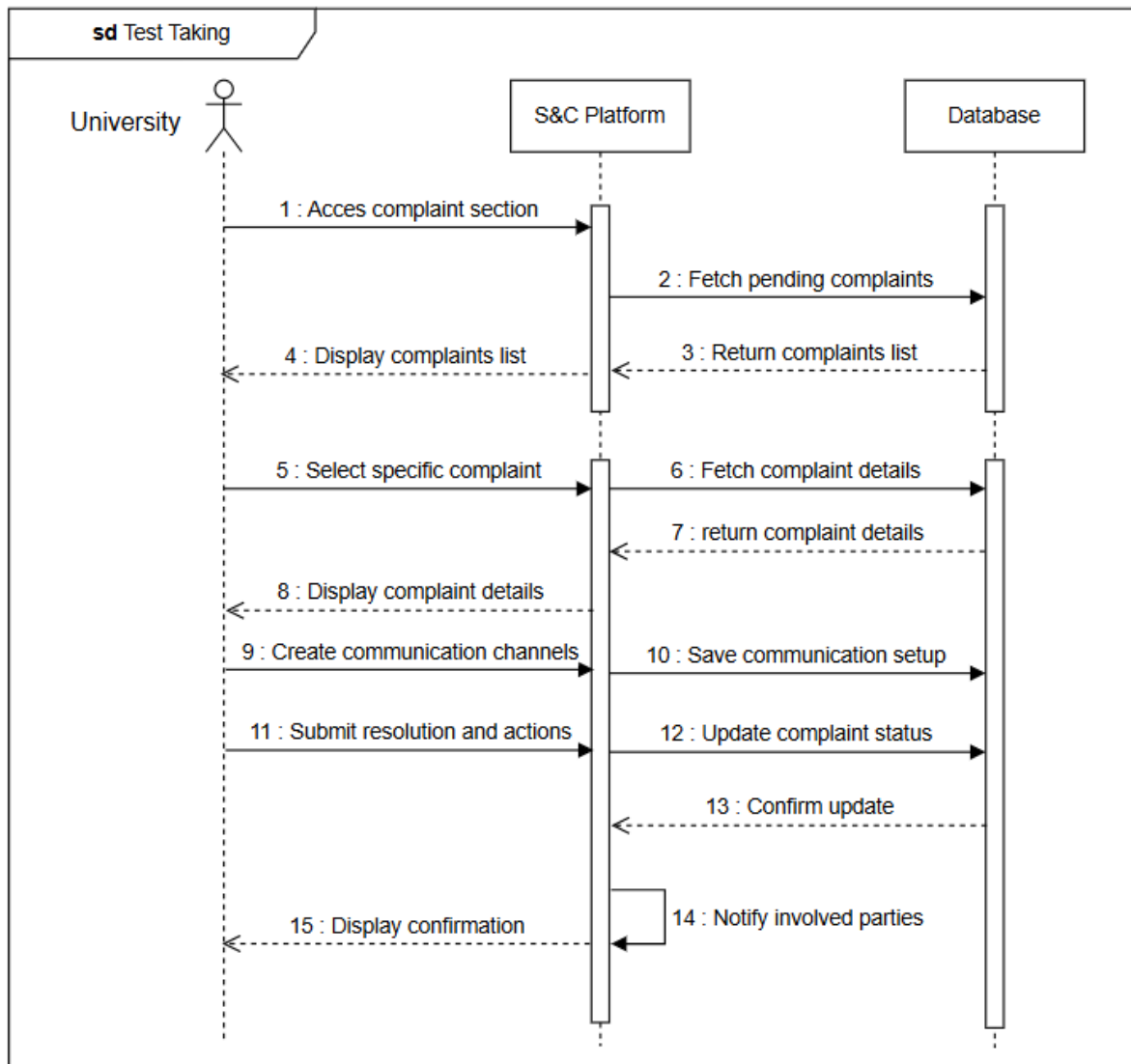


Figure 3.22: [UC18] Test Taking

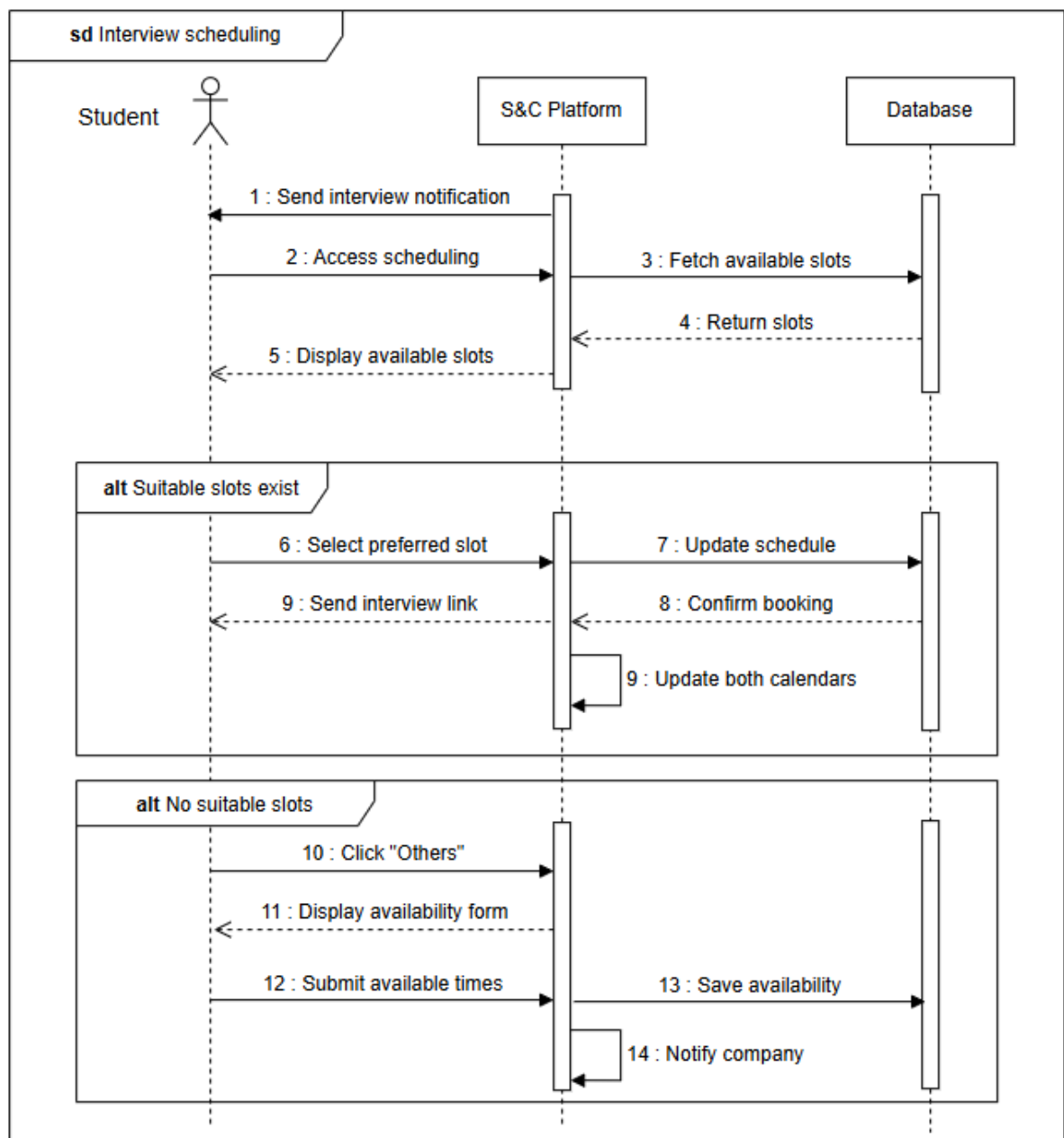


Figure 3.23: [UC19] Interview Scheduling

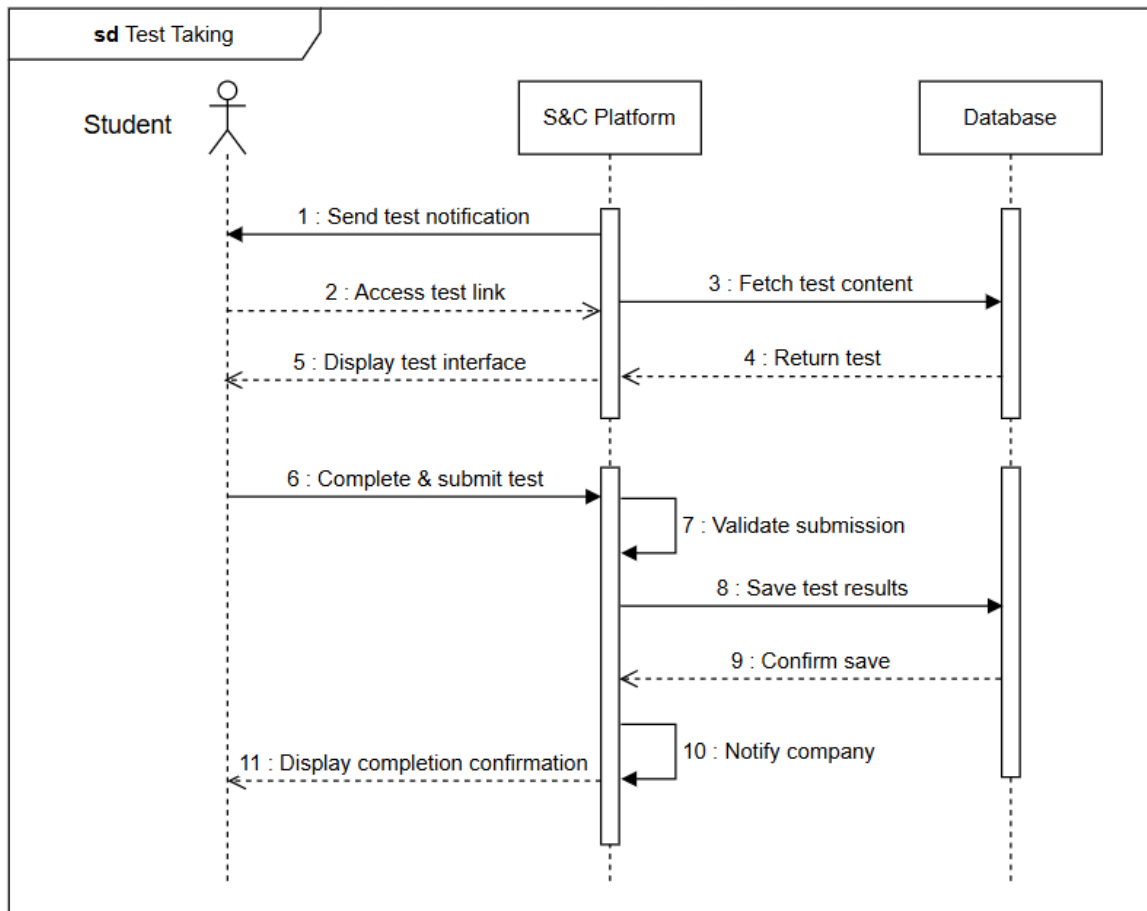


Figure 3.24: [UC20] Test Taking



### 3.2.4. Requirement mapping

<b>G[1]: Students discover and apply for internships</b>	
R[1]: The system allows users (students, companies, and universities) to register by providing their personal information, a valid email address, and a password	D[1]: Users must have a reliable internet connection
R[2]: The system allows registered users to log in and access the platform's features based on their roles	D[2]: User personal information must be correct
R[5]: The system allows students to browse and filter internship offers based on criteria like location, field, required skills, and degree level	D[3]: Student CVs must correspond to their skills
R[6]: The system notifies students about new internship offers matching their profile and preferences	
R[7]: The system allows students to upload their CVs on the platform	
R[8]: The system allows students to update their CVs and keep their profiles up to date	
R[9]: The system allows students to apply to internship offers directly from the platform by attaching their CVs and adding optional documents	
R[25]: The system notifies students of application deadlines, interview schedules, test results, and application status updates in real-time	

Table 3.1: Goal Analysis: Student Discovery and Application Process

<b>G[2]: Companies advertise their internship offers</b>	
R[1]: The system allows users (students, companies, and universities) to register by providing their personal information, a valid email address, and a password	D[1]: Users must have a reliable internet connection
R[2]: The system allows registered users to log in and access the platform's features based on their roles	D[2]: User personal information must be correct
R[3]: The system allows companies to create, modify, and delete internship offers	D[4]: Company posts must reflect what they truly offer
R[4]: The system allows companies to manage the status of internship offers (e.g., open, closed, completed)	
R[26]: The system notifies companies of updates on candidate applications and recruitment stages	

Table 3.2: Goal Analysis: Company Internship Advertising

<b>G[3]: Facilitates connections and structured recruitment processes</b>	
R[11]: The system allows companies to initiate recruitment processes that may include tests, interviews, or both	D[1]: Users must have a reliable internet connection
R[12]: The system allows companies to manage recruitment stages, including scheduling interviews, assigning tests, and evaluating results	D[2]: User personal information must be correct
R[13]: The system provides tools for companies to evaluate, score, and rank candidates based on test results, interview feedback, and overall suitability	D[3]: Student CVs must correspond to their skills
R[14]: The system notifies students of updates related to their recruitment status (e.g., rejected, hired, shortlisted)	D[4]: Company posts must reflect what they truly offer
R[18]: The system allows students to provide feedback on their application and recruitment experiences	
R[19]: The system allows companies to provide feedback on candidates they evaluated and their recruitment experiences	
R[25]: The system notifies students of interview schedules, test results, and recruitment updates in real time	
R[26]: The system notifies companies of updates on candidate applications and recruitment stages	

Table 3.3: Goal Analysis: Facilitating Connections and Recruitment Processes

<b>G[4]: Students and companies receive personalized suggestions</b>	
R[15]: The system analyzes student CVs and internship offers to generate actionable suggestions for both students and companies	D[1]: Users must have a reliable internet connection
R[16]: The system sends personalized suggestions to students to improve their CVs, profiles, cover letters, or applications	D[2]: User personal information must be correct
R[17]: The system sends personalized suggestions to companies to improve their internship offers or recruitment strategies	D[3]: Student CVs must correspond to their skills
	D[6]: User feedback must be truthful

Table 3.4: Goal Analysis: Personalized Suggestions for Students and Companies

<b>G[5]: Sends recommendations by matching profiles</b>	
R[20]: The system uses a matching algorithm to analyze student CVs, internship offers, and user feedback to generate recommendations	D[1]: Users must have a reliable internet connection
R[21]: The system sends recommendations to both students and companies when a match is found	D[2]: User personal information must be correct
R[18]: The system allows students to provide feedback on their application and recruitment experiences	
R[19]: The system allows companies to provide feedback on candidates they evaluated and their recruitment experiences	

Table 3.5: Goal Analysis: Sends Recommendations by Matching Profiles

G[6]: Universities monitor internship progress and issues	
R[1]: The system allows users (students, companies, and universities) to register by providing their personal information, a valid email address, and a password	D[1]: Users must have a reliable internet connection
R[2]: The system allows registered users to log in and access the platform’s features based on their roles	D[2]: User personal information must be correct
R[22]: The system allows students and companies to submit complaints about any issues related to the internship or recruitment process	D[5]: Universities must monitor the situation of internships
R[23]: The system allows universities to monitor and resolve complaints submitted by students or companies	
R[24]: The system notifies universities of unresolved issues or complaints requiring attention	

Table 3.6: Goal Analysis: Universities Monitor Internship Progress and Issues

3.3. Performance Requirements

The platform must deliver good performance to accommodate a substantial user base comprising both educators and students. To enhance user satisfaction and meet performance targets, the system must maintain swift response times not exceeding one second. It’s important to note that user experience may be impacted by slow internet connectivity, potentially resulting in increased response times.

3.4. Design Constraints

### 3.4.1. Standards compliance

The platform prioritizes user privacy by strictly adhering to the General Data Protection Regulation (GDPR), which governs data protection and privacy rights for individuals within the European Union (EU) and European Economic Area (EEA). Additionally, the system implements international date and time formats to maintain compliance with current standards.

### 3.4.2. Hardware limitations

The following hardware specifications outline the minimum requirements for optimal platform usage:

- **Network Connectivity:** Users must have access to reliable internet service. Compatible network standards include 3G, 4G, 5G, IEEE 802.11, and IEEE 802.3. Connection must be established through appropriate network infrastructure such as modems, access points, or similar devices
- **Device Specifications:** Users require devices equipped with standard processing capabilities (equivalent to Intel i3 or i5), HD-quality display resolution or better, and minimum 4GB RAM capacity.

### 3.4.3. Any other constraint

Given the platform's extensive feature set and expected prolonged user engagement, the interface must emphasize user-friendliness and intuitive navigation.

## 3.5. Software System Attributes

The following section details essential software attributes required by the system.

### 3.5.1. Reliability

Long-term continuous operation necessitates robust system reliability. This is achieved through implementation of data replication and consistency mechanisms to prevent system failures. Additionally, regular offline backup procedures are essential for data recovery purposes.

### **3.5.2. Availability**

System availability is paramount, with a target uptime of 99%. To achieve this, the platform implements comprehensive replication strategies and eliminates single points of failure. The system is specially engineered to handle peak loads during submission deadlines.

### **3.5.3. Security**

Given the sensitive nature of stored personal data, security is critically important. The system implements encrypted password storage in the central database, comprehensive data protection measures against internal and external threats, and ensures data integrity, consistency, and confidentiality through robust cyber-security policies. Special attention is given to secure execution of student code during dynamic analysis to prevent malicious code execution.

### **3.5.4. Maintainability**

The system emphasizes maintainability through comprehensive code documentation. Testing protocols must achieve minimum 75% code coverage, excluding UI components.

### **3.5.5. Portability**

As a web-based application, the platform ensures cross-browser compatibility (Firefox, Google Chrome, etc.) and device responsiveness (smartphones, computers, and other devices).





## 4 | Formal Analysis Using Alloy

In the part, we focus on the matching algorithm used in the platform, specifically the recommendation generation and application process. The matching system works by generating recommendations based on the student's CV, the internship offer's required skills, and the student's job preferences. We did not go deeper into the interview process as that is not the primary focus of this implementation. We will describe the model in detail, explaining each part of the code, including the relevant facts and run scenarios.

### 4.1. Model Overview

The Alloy model includes several key signatures and facts that simulate the matching process. The main components are:

- **Student:** Students have a CV, which includes skills and preferences, and belong to a university.
- **InternshipOffer:** Internship offers have titles, required skills, and are published by companies.
- **Company:** Companies publish internship offers.
- **Recommendation:** Recommendations are created based on the matching of student skills and internship offer requirements. Each recommendation is sent to both the student and the company.
- **Application:** Students can apply for internship offers, and each application has a status (e.g., Pending, Accepted, Rejected).
- **Status:** The status of applications or recommendations, defined by states like Pending, Accepted, or Rejected.

### 4.2. Code Description

Below is the full Alloy code, with explanations inline:

```
module internshipPlatformMatching

abstract sig User {}

sig Student extends User {
  cv: one CV,
  from: one University
}

sig CV {
  skills: set String,
  preferences: set String
}

sig Company extends User {}

sig University extends User {}

sig InternshipOffer {
  publishedBy: one Company,
  title: one String,
  requiredSkills: set String
}

// Status as abstract sig with extensions instead of enum
abstract sig Status {}
one sig Pending extends Status {}
one sig Accepted extends Status {}
one sig Rejected extends Status {}

sig Application {
  sentBy: one Student,
  relatedTo: one InternshipOffer,
  status: one Status
}

// Signature for Recommendation
```

```

sig Recommendation {
    offer: one InternshipOffer, // The offer being recommended
    sentToStudent: one Student, // The student receiving the recommendation
    sentToCompany: one Company // The company receiving the recommendation
}

// Fact to ensure each CV is linked to exactly one student
fact cvLinkedToOneStudent {
    all c: CV | one s: Student | c = s.cv
}

// Fact to ensure unique applications per student and offer
fact uniqueApplications {
    all s: Student, o: InternshipOffer |
        lone a: Application |
            a.sentBy = s and
            a.relatedTo = o
}

// Fact to create recommendations
fact validRecommendations {
    all r: Recommendation | {
        let s = r.sentToStudent, o = r.offer, c = r.sentToCompany |
        // A recommendation must have valid links
        some (s.cv.skills & o.requiredSkills) and
        some (s.cv.preferences & o.title) and
        c = o.publishedBy
    }
}

// Fact to ensure a student who applied cannot receive a recommendation
fact recommendationApplicationMutualExclusion {
    all s: Student, o: InternshipOffer | {
        (some a: Application | a.sentBy = s and a.relatedTo = o)
        implies
        (no r: Recommendation | r.sentToStudent = s and r.offer = o)
    }
}

```

```

all s: Student, o: InternshipOffer | {
    (some r: Recommendation | r.sentToStudent = s and r.offer = o)
    implies
    (no a: Application | a.sentBy = s and a.relatedTo = o)
}
}

```

### 4.3. Scenario 1: Two Students and One Internship Offer

The first scenario we run focuses on generating a recommendation for a student. This scenario ensures the basic workflow of matching a student's skills with an internship offer and generating a recommendation. Specifically:

- Two students are defined, each with distinct skills and preferences.
- One internship offer is created, requiring specific skills (Python and Java).
- The first student receives a recommendation based on their matching skills with the internship offer.

#### 4.3.1. Explanation of Scenario

In this scenario, we define two students, 's1' and 's2', each with their own skills and preferences. The internship offer requires skills like "Python" and "Java". If the first student's skills match these requirements, a recommendation is generated and sent to both the student and the company. We check that the student has not applied for the internship offer before the recommendation is created, ensuring mutual exclusion between applications and recommendations.

#### 4.3.2. Code for Scenario 1

```

run twoStudentsScenario {
    // Make sure we have exactly 2 distinct students
    some disj s1, s2: Student | {
        // First student has programming skills
        s1.cv.skills = {"Python" + "Java"}
        s1.cv.preferences = {"Developer"}
    }
}

```

```

// Second student has different skills
s2.cv.skills = {"SQL" + "Database"}
s2.cv.preferences = {"Developer"}

// Set up the offer
some o: InternshipOffer | {
    o.title = "Developer"
    o.requiredSkills = {"Python" + "Java"}
    o.publishedBy in Company // Make sure offer is linked to a company

// Ensure recommendation exists for first student
some r: Recommendation | {
    r.sentToStudent = s1
    r.offer = o
    r.sentToCompany = o.publishedBy // Link recommendation to the comp
}
}

} for 5 but
    exactly 2 Student,
    exactly 2 CV,
    exactly 1 Company,
    exactly 1 InternshipOffer,
    exactly 1 University,
    exactly 1 Recommendation

```



In the second scenario, we check that if a student applies for an internship offer, no recommendation can be generated for the same student and offer. Similarly, if a recommendation exists, no application can be submitted. This ensures that there is no overlap in the processes of application and recommendation for a given student and internship offer.

The goal of this scenario is to test the enforcement of mutual exclusion between applications and recommendations. Specifically:

- #### 4.4.2. Code for Scenario 2

```
run noApplicationAndRecommendationScenario {
  // Ensure no recommendation exists if a student has applied
  some s: Student, o: InternshipOffer | {
```

```

s.cv.skills = {"Python" + "Java"}
o.title = "Developer"
o.requiredSkills = {"Python" + "Java"}
s.cv.preferences = {"Developer"}

// Student applies for the offer
some a: Application | {
  a.sentBy = s
  a.relatedTo = o
  a.status = Pending
}
}
} for 5 but
  exactly 1 Student,
  exactly 1 CV,
  exactly 1 InternshipOffer,
  exactly 1 Company

```

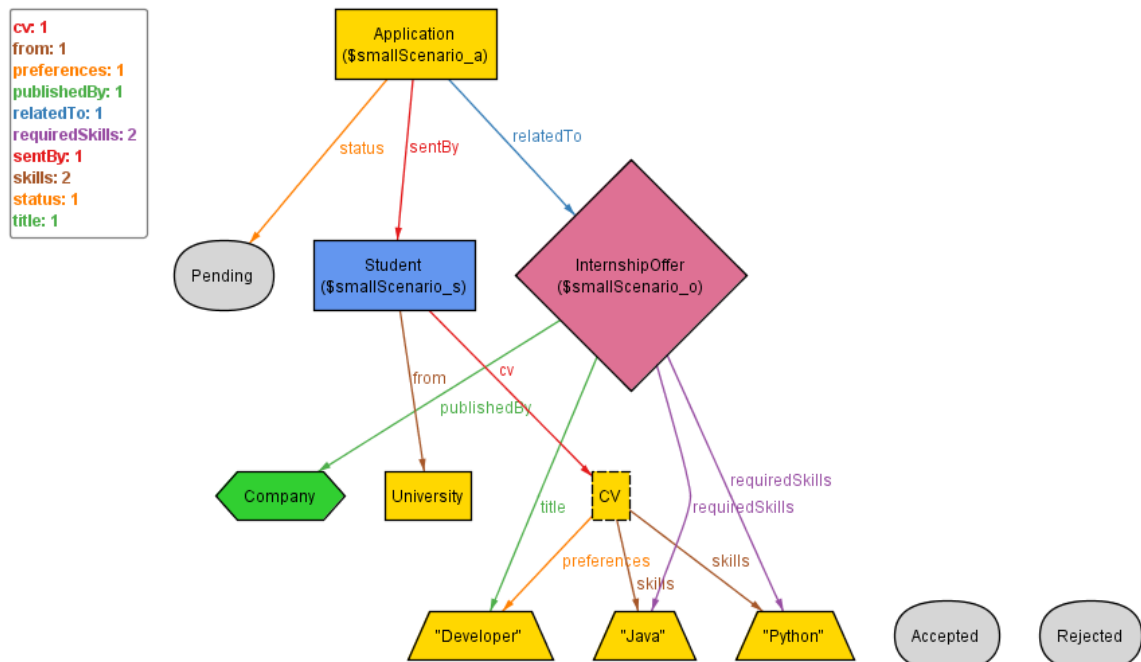


Figure 4.2: Scenario 2: A student applying for an offer





# 5 | Effort Spent

- YE Yumeng

Chapter	Effort (in hours)
1	5
2	3
3	12
4	10

- Léo APPOURCHAUX

Chapter	Effort (in hours)
1	5
2	12
3	8
4	2