



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Students&Companies

REQUIREMENTS ANALYSIS
AND SPECIFICATION DOCUMENT

**Andrea Carrara
and Federica Currò Dossi**

Contents

Contents	iii
1 Introduction	1
1.1 Purpose	1
1.1.1 Goals	2
1.2 Scope	2
1.2.1 World Phenomena	2
1.2.2 Shared Phenomena	3
1.3 Glossary	6
1.4 Reference Documents	7
1.5 Document Structure	7
2 Overall Description	9
2.1 Product Perspective	9
2.1.1 Scenarios	9
2.1.2 Class Diagram	11
2.1.3 State Diagrams	11
2.2 Product Functions	11
2.3 User Characteristics	14
2.4 Assumptions and Constraints	15
3 Specific Requirements	17
3.1 External Interface Requirements	17
3.1.1 User Interfaces	17
3.1.2 Hardware Interfaces	17
3.1.3 Software Interfaces	17
3.1.4 Communication Interfaces	19
3.2 Functional Requirements	19
3.2.1 Requirements	19
3.2.2 Use Cases	25
3.3 Performance Requirements	76
3.4 Design Constraints	76
3.4.1 Standard Compliance	76
3.4.2 Hardware Limitations	77
3.5 Software System Attributes	77
3.5.1 Reliability	77
3.5.2 Availability	77

3.5.3	Security	77
3.5.4	Maintainability	77
3.5.5	Portability	78
4	Formal Analysis	79
4.1	Model	79
4.1.1	Example Worlds	91
4.1.2	Base World	91
5	Workload	95
	Bibliography	97
	List of Figures	99
	List of Tables	101

1 | Introduction

Universities focus on academic subjects and emphasize theoretical knowledge, often leaving students without sufficient practical experience. Internships fill this gap by helping students refine their skills, understand leadership dynamics, gain insights into workplace culture and build professional networks. However, finding internships can be challenging due to limited knowledge of relevant companies, and matching the right student with the right opportunity is even harder for both students and companies, given the difficulty in assessing competencies. In this context, Students&Companies comes to help by connecting university students and companies offering internships.

1.1. Purpose

Students&Companies is designed to connect university students seeking internships with companies offering them. The platform enables students to proactively search through a wide range of internship positions and it also provides personalized recommendations using advanced mechanisms such as statistical analysis. Similarly, companies receive suggestions for students who stand out as strong candidates.

Students can enhance their profiles by adding CVs and specifying job preferences, ensuring their applications are tailored and relevant. Companies, in turn, can post precise job descriptions, outlining project details, required skills and terms of the internship, such as compensation or mentorship opportunities.

Beyond matching, S&C streamlines the entire selection process. Companies can schedule interviews, use structured questionnaires to evaluate candidates and finalize decisions directly through the platform.

The system also supports internships after placement. Both students and companies can track progress and address concerns by exchanging comments, promoting transparency and accountability. Universities can also actively oversee their students' internships, ensuring quality experiences and adequate support.

Lastly, to feed the statistical analysis used for recommendations, S&C collects feedback about the platform through forms from both students and companies, enhancing its ability to refine and personalize the recommendation system.

1.1.1. Goals

By analyzing the previous description of the product and by clustering its features, the following goals emerge:

ID	Description
G1	S&C allows USs to search and apply for POs.
G2	S&C recommends POs to USs and USs to COs.
G3	S&C provides support for the selection process.
G4	S&C allows USs, COs and UNs monitor INs.
G5	S&C collects feedback from USs and COs.

Table 1.1: Goals

1.2. Scope

To clearly understand the scope of Students&Companies, it is helpful to characterize its context using the world and the machine model [2]. The system is divided in two: the machine, representing the software to be developed, and the world, encompassing the portion of reality it influences. This approach allows for a formal definition of the world and shared phenomena, facilitating a better analysis of the product's domain.

1.2.1. World Phenomena

The machine does not observe the following world phenomena:

ID	Description
WP1	A US wants to do an IN.
WP2	A CO wants to host an IN.
WP3	A UN wants to monitor a US.
WP4	A US writes his CV.
WP5	A CO interviews a US.
WP6	A CO evaluates a US.
WP7	A US does an IN at a CO.

Table 1.2: World phenomena

1.2.2. Shared Phenomena

The following are phenomena involving both the world and the machine, categorized by their controller and observer.

Controller: World - Observer: Machine

ID	Description
SP1	A US signs up on S&C.
SP2	A US deletes his profile.
SP3	A US logs in S&C.
SP4	A US logs out.
SP5	A US adds his CV.
SP6	A US updates his CV.
SP7	A US adds preferences on POs.
SP8	A US updates preferences on POs.
SP9	A US searches for a PO.
SP10	A US views a PO.
SP11	A US views a CO.
SP12	A US applies for a PO.
SP13	A US withdraws an application.
SP14	A US accepts a recommended PO.
SP15	A US declines a recommended PO.
SP16	A US fills out a feedback form.
SP17	A US views the status of a selection process.
SP18	A US updates the status of a selection process.
SP19	A US fills out a questionnaire for a selection process.
SP20	A US accepts a scheduled interview for a selection process.
SP21	A US declines a scheduled interview for a selection process.
SP22	A US views the outcome of a selection process.
SP23	A US views the status of an IN.
SP24	A US updates the status of an IN.

SP25	A US comments an IN.
SP26	A US views a comment.
SP27	A CO signs up on S&C.
SP28	A CO deletes its profile.
SP29	A CO logs in S&C.
SP30	A CO logs out.
SP31	A CO updates its profile.
SP32	A CO posts a PO.
SP33	A CO removes a PO.
SP34	A CO adds the description of a PO.
SP35	A CO updates the description of a PO.
SP36	A CO views a US.
SP37	A CO accepts a recommended US.
SP38	A CO declines a recommended US.
SP39	A CO fills out a feedback form.
SP40	A CO views the status of a selection process.
SP41	A CO updates the status of a selection process.
SP42	A CO adds a questionnaire for a selection process.
SP43	A CO views a questionnaire for a selection process.
SP44	A CO schedules an interview for a selection process.
SP45	A CO updates the outcome of a selection process.
SP46	A CO views the status of an IN.
SP47	A CO updates the status of an IN.
SP48	A CO comments an IN.
SP49	A CO views a comment.
SP50	A UN signs up on S&C.
SP51	A UN deletes its profile.
SP52	A UN logs in S&C.
SP53	A UN logs out.

SP54	A UN views a US.
SP55	A UN views an IN.
SP56	A UN views a CO.
SP57	A UN views the status of an IN.
SP58	A UN updates the status of an IN.
SP59	A UN comments an IN.
SP60	A UN views a comment.

Table 1.3: Shared phenomena controlled by the world and observed by the machine

Controller: Machine - Observer: World

ID	Description
SP61	S&C notifies a US of a recommended PO.
SP62	S&C notifies a CO of a recommended US.
SP63	S&C notifies a US of a match with a CO.
SP64	S&C notifies a CO of a match with a US.
SP65	S&C notifies a US of an update on the status of a selection process.
SP66	S&C notifies a CO of an update on the status of a selection process.
SP67	S&C notifies a US if a CO adds a questionnaire for a selection process.
SP68	S&C notifies a CO if a US fills out a questionnaire for a selection process.
SP69	S&C notifies a US if a CO schedules an interview for a selection process.
SP70	S&C notifies a CO if a US accepts a scheduled interview for a selection process.
SP71	S&C notifies a CO if a US declines a scheduled interview for a selection process.
SP72	S&C notifies a US of the outcome of a selection process.
SP73	S&C notifies a US of an update on the status of an IN.

SP74	S&C notifies a CO of an update on the status of an IN.
SP75	S&C notifies a UN of an update on the status of an IN.
SP76	S&C notifies a US of a comment on an IN.
SP77	S&C notifies a CO of a comment on an IN.
SP78	S&C notifies a UN of a comment on an IN.

Table 1.4: Shared phenomena controlled by the machine and observed by the world

1.3. Glossary

Below are listed definitions, acronyms and abbreviations used throughout the document:

ID	Description
S&C	Students&Companies
US	University student
IN	Internship job
PO	Posted internship position
CO	Company
UN	University
EP	Email provider
CV	Curriculum vitae
Match	A US and a CO declare interest in each other
Contact	Selection process and internship progress
Gn	Goal number n
WPn	World phenomenon number n
SPn	Shared phenomenon number n
Sn	Scenario number n
Fn	Function number n
DAn	Domain assumption number n
Cn	Constraint number n
UCn	Use case number n

Table 1.5: Glossary

1.4. Reference Documents

The document is structured according to the IEEE 29148:2018 standard for requirements engineering [1] and the project assignment by Professor Di Nitto [3].

1.5. Document Structure

The purpose of the document is to analyze the requirements and describe the specification of Students&Companies. It is intended for developers tasked with implementation, but it also serves as a mutual agreement between customers and contractors. Its structure follows a top-down approach, starting with a high-level overview of the problem and its requirements and progressing to more detailed descriptions. The document is divided into five chapters, each with a distinct scope and objective, as outlined below.

Introduction

The first chapter starts by contextualizing the problem, introducing the product and defining its goals. It also provides a detailed list of global and shared phenomena, and concludes with a glossary essential for understanding the subsequent chapters.

Overall Description

The second chapter offers a thorough examination of the problem, exploring the domain in greater detail through various scenarios and the use of class and state diagrams. It also addresses product and user characteristics, along with assumptions and constraints.

Specific Requirements

The third chapter focuses on a detailed analysis of the specific requirements, providing comprehensive insights into external interface requirements, functional requirements and performance requirements.

Formal Analysis

The fourth chapter utilizes Alloy for formal analysis, aiming to validate the accuracy of the model outlined in the previous sections. It emphasizes presenting the results of the checks performed and the relevant assertions.

Workload

Chapter five quantifies the contributions of the two authors in writing this document.

2

Overall Description

This chapter provides a detailed description of the whole problem, offering a deeper exploration of the domain through various scenarios and the application of class and state diagrams. It also covers key aspects such as product and user characteristics, as well as the underlying assumptions and constraints shaping the context.

2.1. Product Perspective

This section employs scenarios to illustrate the platform's role, supported by class and state diagrams that define its structure and dynamic behavior. These tools collectively establish a clear understanding of how the product integrates into its environment and addresses the identified problem.

2.1.1. Scenarios

S1. Signup and Profile Setup

AlgoSphere is a company specializing in algorithm development and artificial intelligence solutions for businesses across industries. To cultivate fresh talent and build a pipeline of future employees, the company is considering offering several internship positions. To save valuable time and resources, AlgoSphere opts to use Students&Companies. It navigates to the S&C platform via a browser, selects the option "Sign Up as a Company" and provides its name, email and field of operation.

Ben Pinter, a first-year Master's student in computer science, is eager to gain real-world experience in AI through an internship. To streamline his search and avoid the inefficiency of navigating multiple websites and job boards, Ben signs up on S&C, hoping to find opportunities aligned with his academic background and career goals. Accessing the platform via a browser, he selects the "Sign Up as a Student" option and completes his profile by providing essential details, including his name, surname and email. He also has the option to add information such as his CV and personal preferences. These preferences may include availability, expected salary and benefits.

TechVille University, where Ben is currently enrolled, recognizes the importance of staying involved in students' professional development to ensure they gain valuable experience and meet academic standards. To facilitate this, TechVille University signs up on S&C by selecting "Sign Up as a University" and providing its name and email. By linking Ben's profile to the university through his institutional email, the platform allows TechVille Uni-

versity to automatically monitor Ben's internships, ensuring alignment with educational goals and providing support throughout his professional journey.

S2. Internship Posting and Candidate Matching

AlgoSphere wants to post an internship position in AI. It clicks the corresponding button and describes the opportunity by outlining its tasks, application domain, project scope and offered terms such as salary and benefits. After publishing, S&C recommends Ben's profile on AlgoSphere's homepage, as his skills and experience closely match the internship requirements. In turn, the platform notifies Ben about the opportunity.

Meanwhile, John Kent, another student in computer science, is registered on the platform but has not uploaded his CV or specified preferences. Interested in artificial intelligence systems for cybersecurity, John uses the search bar to look for relevant opportunities by entering "AI." Among the results, he finds AlgoSphere's internship and applies, which prompts a notification to AlgoSphere. To strengthen his application and improve his chances of being selected, John uploads his CV and specifies preferences.

S3. Selection Process

After being notified, both Ben and AlgoSphere accept S&C's recommendation, formally initiating contact and starting the selection process. Similarly, upon reviewing John's application, the company determines that his profile merits consideration. The platform thus notifies John of the start of the selection process.

AlgoSphere uploads questionnaires tailored to the internship position, prompting notifications to both Ben and John to complete them. When one submits them, the platform notifies AlgoSphere of the update and allows the company to view the answers.

After evaluating the questionnaires, AlgoSphere selects Ben as the stronger candidate due to his more aligned skills. John is notified of his rejection, while Ben receives a notification detailing the next steps. AlgoSphere then schedules a remote interview over an in-person one for better flexibility, which Ben accepts. After a successful interview, the company finalizes the outcome of the selection process and notifies Ben, all through S&C.

S4. Internship Monitoring

Throughout Ben's internship, AlgoSphere will use S&C to provide detailed progress reports or other comments, with notifications keeping Ben and his university informed of these updates in real time. In turn, if Ben will encounter an issue during his time at AlgoSphere, such as unclear expectations from the company, he will comment a complaint visible to both AlgoSphere and TechVille University.

S5. Feedback Forms

At key stages of the matchmaking, selection and internship processes, Ben and AlgoSphere will be asked to provide feedback on the quality of the service offered by S&C. Ben will share comments with respect to the quality of the recommendations he received and the

efficiency of the selection process he completed. In the same way, AlgoSphere will evaluate the platform's ability to provide suitable candidate matches, the usability of the system during selection and the effectiveness of its communication tools.

2.1.2. Class Diagram

The high-level class diagram below represents the entire system, modeling the elements from the previous scenarios along with their interactions and cardinalities.

2.1.3. State Diagrams

The two state diagrams below aim to clarify potential misunderstandings about the recommendations and selection process by illustrating their dynamic behavior.

Recommendations

Recommendations are initiated when a company posts an internship position or a university student uploads their CV or personal preferences. S&C then checks for potential matches between available internship positions and student profiles. Upon finding a match, the platform notifies the student about the position and the company about the student. Each party can choose to accept or reject the recommendation. If either party rejects it, the system resumes the search for another match. However, if both accept, the system finalizes the match and notifies both the student and the company accordingly.

Selection Process

The selection process begins when a match is made between a university student and a company. This happens either through mutual acceptance of a recommendation or when a student applies for an internship and the company accepts his application. Once this connection is made, the system notifies both parties about the match.

From there, the company can choose to either add a questionnaire for the student or schedule an interview. If an interview is scheduled, the student can accept or decline it. After reviewing the questionnaire or evaluating the interview, the company decides whether to continue by sending another questionnaire or scheduling further interviews, or to conclude the process, notifying the student of the final outcome.

2.2. Product Functions

The following is a summary of the main functions of Students&Companies, providing a concise abstraction and generalization of its goals and scenarios. This serves as an intermediate step bridging goals and scenarios with requirements and use cases.

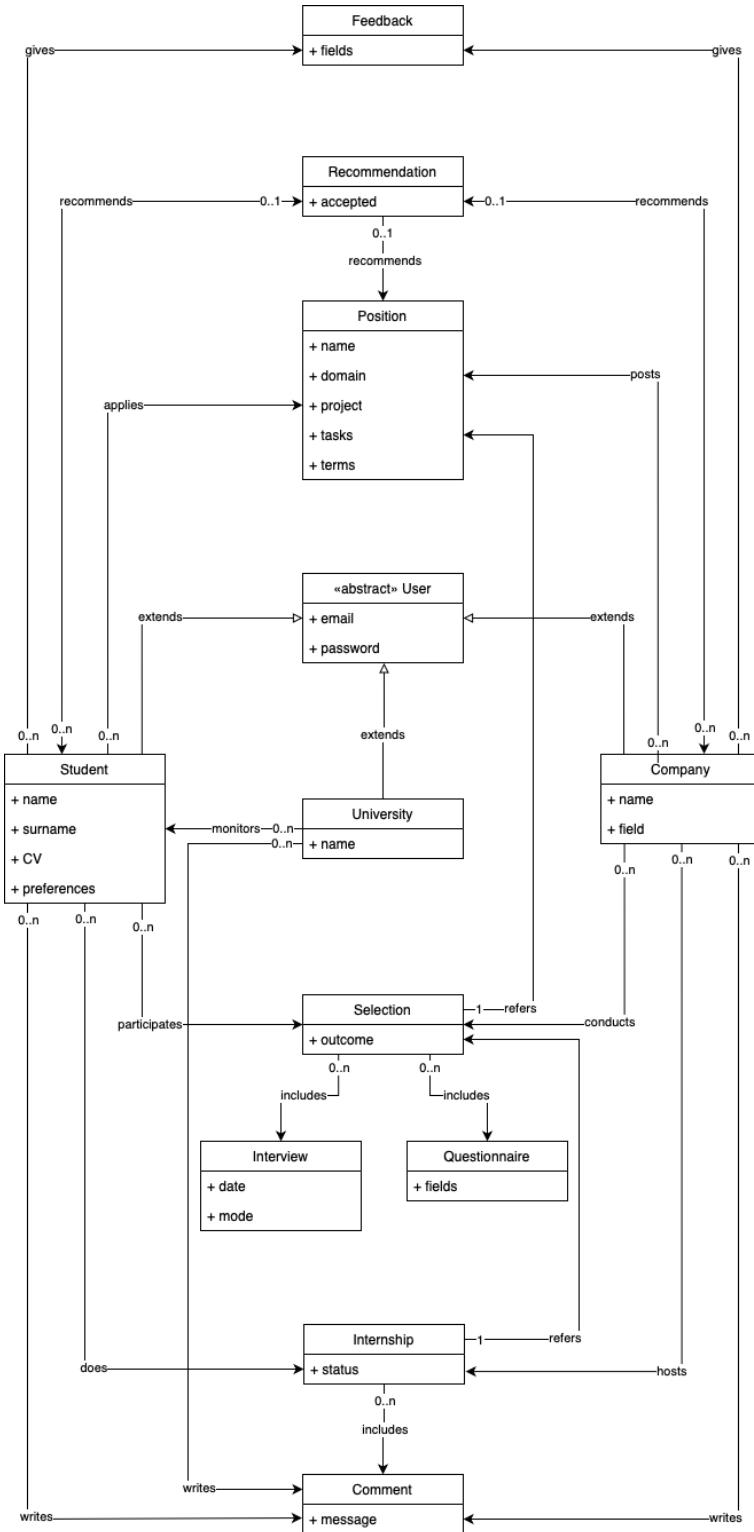


Figure 2.1: Class diagram

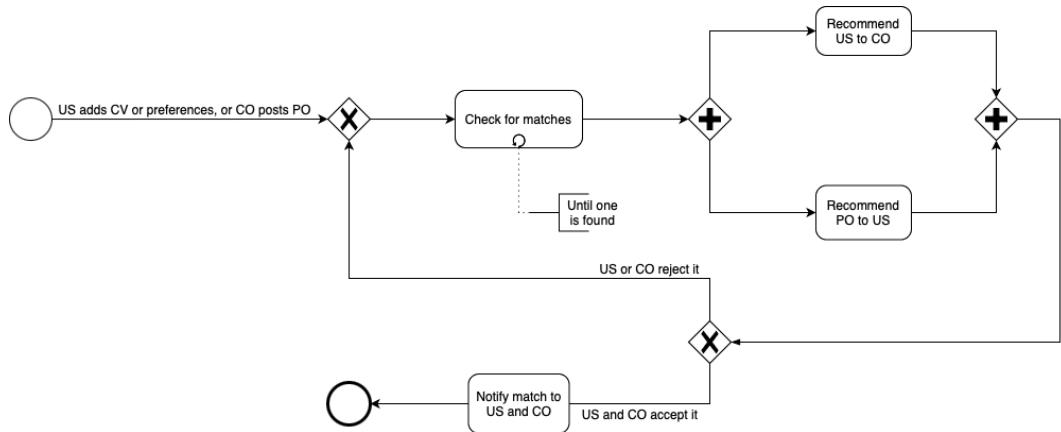


Figure 2.2: Recommendations state diagram

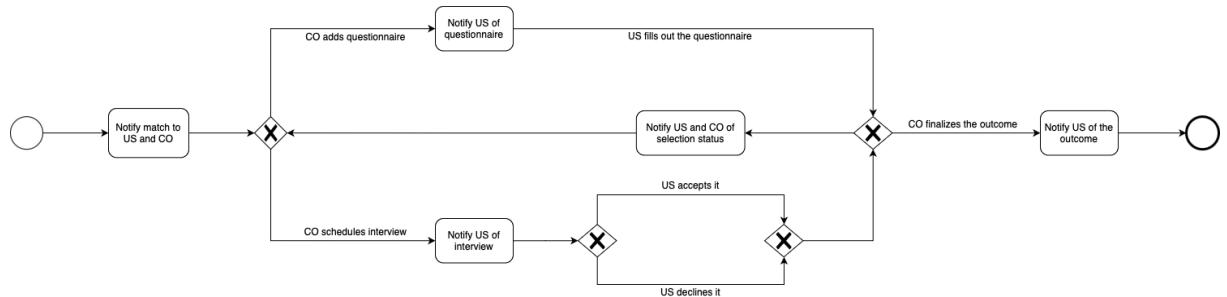


Figure 2.3: Selection process state diagram

F1. Signup and Profile Setup

Students sign up with their name, surname and institutional email. Additionally, they can upload a CV and set preferences. Similarly, a company signs up by providing its name, email and field of operation, while universities do so by submitting their name and institutional email. Students with the same institutional email domain are automatically linked to the university, allowing seamless monitoring. After logging in, users can log out from their session and must log in again to reaccess the platform.

F2. Internship Posting and Searching

Companies post internships by specifying their tasks, application domain, project scope and terms. In turn, students write keywords in the search bar to find positions.

F3. Internship Application

Students apply for internships and notifications are sent to companies, which can in turn review candidates' profiles and start the selection process.

F4. Recommendations

Students are recommended internship positions from suitable companies, while companies are recommended students based on skills and preferences. Both students and companies must accept a recommendation to start the selection process.

F5. Selection Process

Companies assess students through questionnaires and scheduled interviews, either virtual or in-person. They then finalize the outcome, with notifications keeping both parties informed at every stage.

F6. Internship Monitoring

Students, companies and universities are notified of the status of the ongoing internship and can write comments to share progress or raise concerns.

F7. Feedback Forms

Students and companies provide feedback on matchmaking to refine the statistical analysis on which recommendations are based.

2.3. User Characteristics

Students&Companies has three types of users: university students, companies and universities. Below is a brief explanation of their identities and main characteristics.

University Students

University students are the users seeking internships. A student can sign up by providing his name, surname and valid institutional email address in his personal data. They can also upload a CV and specify personal preferences, such as availability and desired benefits.

Companies

Companies are the users hosting internships and they are typically represented by the human resources department acting on behalf of the organization. Signing up requires the company's name, a valid email address and its field of expertise.

Universities

Universities are the users monitoring their students and they are typically represented by the career services department acting on behalf of the institution. Signing up requires the university's name and a valid institutional email address.

2.4. Assumptions and Constraints

The table below outlines the domain assumptions of Students&Companies:

ID	Description
DA1	USs have a valid institutional email.
DA2	COs have a valid email.
DA3	UNs have a valid institutional email.
DA4	USs have a working device with a reliable internet connection.
DA5	COs have a working device with a reliable internet connection.
DA6	UNs have a working device with a reliable internet connection.

Table 2.1: Domain assumptions

The following table lists the constraints of S&C:

ID	Description
C1	S&C cannot prevent a CO from posting a PO.
C2	S&C cannot prevent a US from applying for a PO.
C3	S&C cannot prevent a UN from monitoring its US.

C4	S&C cannot declare a match if neither the CO evaluated positively the application of a US nor a recommendation was accepted by both.
C5	S&C cannot start a selection process before a match.
C6	S&C cannot end a selection process before an outcome.
C7	S&C cannot start an IN before the end of a selection process.

Table 2.2: Constraints

3

Specific Requirements

This chapter focuses on a detailed analysis of the specific requirements, offering comprehensive insights into external interface requirements, functional requirements and performance requirements, together with design constraints.

3.1. External Interface Requirements

The external interface requirements define how the Students&Companies platform interacts with its users and external systems to deliver its core functionalities. The following interfaces are designed to prioritize accessibility, accommodating a diverse range of user needs while maintaining robust system performance.

3.1.1. User Interfaces

S&C will feature a modern, accessible and responsive web interface designed to meet the needs of students, companies and universities. A mobile-friendly version will ensure accessibility across devices, boosting user engagement.

Wireframe

To illustrate the platform's design, an exemplary wireframe follows to offer a simplified visual representation of the home page from the point of view of a university student.

3.1.2. Hardware Interfaces

The platform is designed to be hardware-agnostic, supporting all standard internet-enabled devices, including desktops, laptops, smartphones and tablets. While S&C ensures compatibility with smaller devices, such as smartphones, features requiring detailed input, for example uploading documents or adding questionnaires, are better suited to a desktop or laptop environment for optimal user experience.

3.1.3. Software Interfaces

S&C requires an email provider interface to send confirmation emails to users during the registration process. Additionally, the platform will send notifications to users through the email provider, ensuring timely updates as needed.

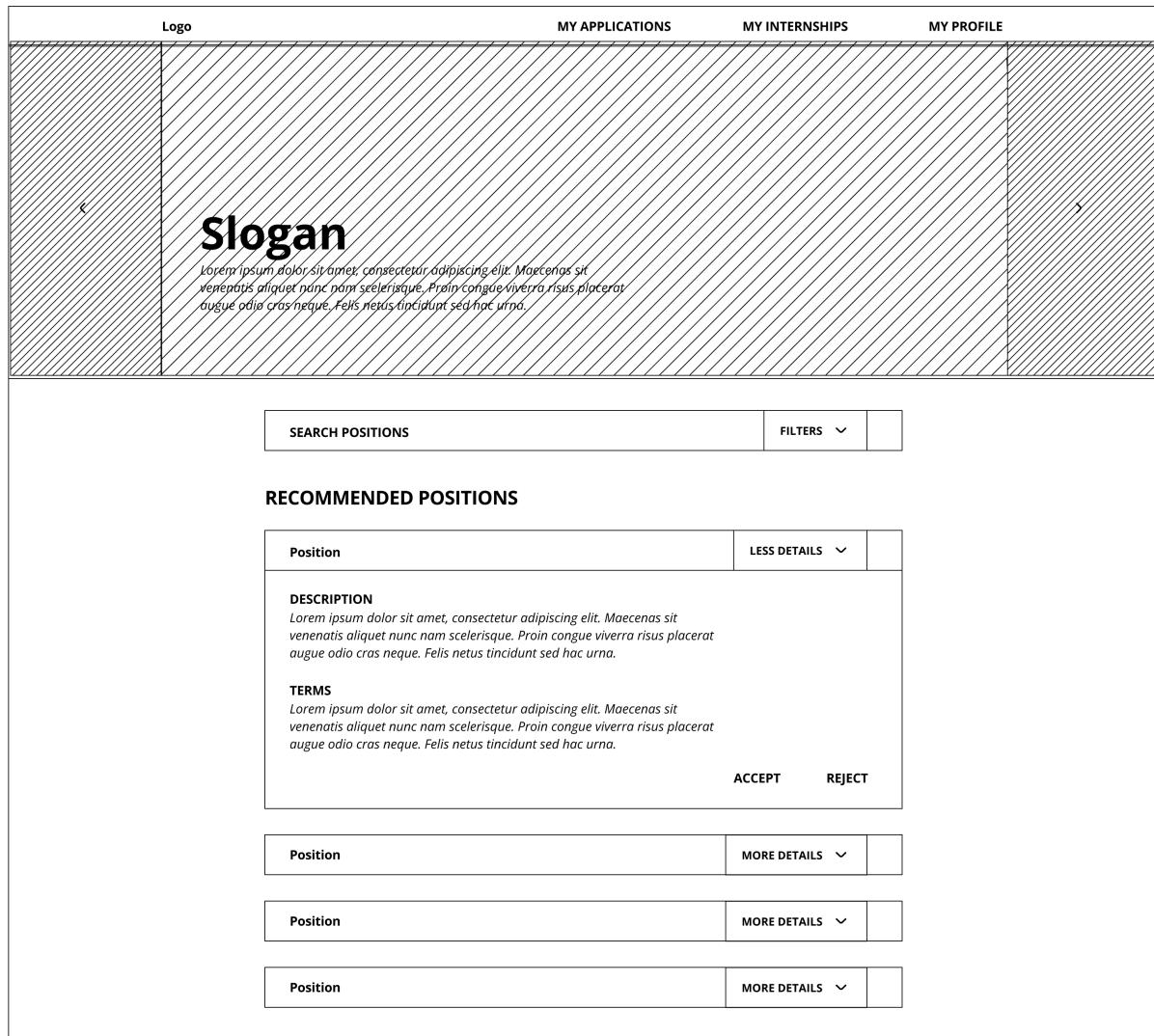


Figure 3.1: Wireframe

3.1.4. Communication Interfaces

The platform will rely on the HTTPS protocol to ensure secure data transmission, protecting user information during interactions. SMTP will handle email notifications, ensuring timely delivery of updates. Additionally, the system will integrate with WebSocket to enable live notifications.

3.2. Functional Requirements

This section lists the detailed functional requirements of Students&Companies, mapped directly to the overarching goals for clarity and traceability. Additionally, use cases are provided to illustrate specific scenarios, ensuring the requirements are grounded in real-world user needs.

3.2.1. Requirements

The table below presents detailed functional requirements.

ID	Description
R1	S&C allows USs to sign up.
R2	S&C allows USs to delete their profile.
R3	S&C allows USs to log in.
R4	S&C allows USs to log out.
R5	S&C allows USs to add their CV.
R6	S&C allows USs to update their CV.
R7	S&C allows USs to add preferences on POs.
R8	S&C allows USs to update preferences on POs.
R9	S&C allows USs to search for a PO.
R10	S&C allows USs to view a PO.
R11	S&C allows USs to view a CO.
R12	S&C allows USs to apply for a PO.
R13	S&C allows USs to withdraw an application.
R14	S&C allows USs to accept a recommended PO.
R15	S&C allows USs to decline a recommended PO.
R16	S&C allows USs to fill out a feedback form.
R17	S&C allows USs to view the status of a selection process.

R18	S&C allows USs to update the status of a selection process.
R19	S&C allows USs to fill out a questionnaire for a selection process.
R20	S&C allows USs to accept a scheduled interview for a selection process.
R21	S&C allows USs to decline a scheduled interview for a selection process.
R22	S&C allows USs to view the outcome of a selection process.
R23	S&C allows USs to view the status of an IN.
R24	S&C allows USs to update the status of an IN.
R25	S&C allows USs to comment an IN.
R26	S&C allows USs to view a comment.
R27	S&C allows COs to sign up.
R28	S&C allows COs to delete their profile.
R29	S&C allows COs to log in.
R30	S&C allows COs to log out.
R31	S&C allows COs to update their profile.
R32	S&C allows COs to post a PO.
R33	S&C allows COs to remove a PO.
R34	S&C allows COs to add the description of a PO.
R35	S&C allows COs to update the description of a PO.
R36	S&C allows COs to view a US.
R37	S&C allows COs to accept a recommended US.
R38	S&C allows COs to decline a recommended US.
R39	S&C allows COs to fill out a feedback form.
R40	S&C allows COs to view the status of a selection process.
R41	S&C allows COs to update the status of a selection process.
R42	S&C allows COs to add a questionnaire for a selection process.
R43	S&C allows COs to view a questionnaire for a selection process.

R44	S&C allows COs to schedule an interview for a selection process.
R45	S&C allows COs to update the outcome of a selection process.
R46	S&C allows COs to view the status of an IN.
R47	S&C allows COs to update the status of an IN.
R48	S&C allows COs to comment an IN.
R49	S&C allows COs to view a comment.
R50	S&C allows UNs to sign up.
R51	S&C allows UNs to delete their profile.
R52	S&C allows UNs to log in.
R53	S&C allows UNs to log out.
R54	S&C allows UNs to view a US.
R55	S&C allows UNs to view an IN.
R56	S&C allows UNs to view a CO.
R57	S&C allows UNs to view the status of an IN.
R58	S&C allows UNs to update the status of an IN.
R59	S&C allows UNs to comment an IN.
R60	S&C allows UNs to view a comment.
R61	S&C notifies a US of a recommended PO.
R62	S&C notifies a CO of a recommended US.
R63	S&C notifies a US of a match with a CO.
R64	S&C notifies a CO of a match with a US.
R65	S&C notifies a US of an update on the status of a selection process.
R66	S&C notifies a CO of an update on the status of a selection process.
R67	S&C notifies a US if a CO adds a questionnaire for a selection process.
R68	S&C notifies a CO if a US fills out a questionnaire for a selection process.

R69	S&C notifies a US if a CO schedules an interview for a selection process.
R70	S&C notifies a CO if a US accepts a scheduled interview for a selection process.
R71	S&C notifies a CO if a US declines a scheduled interview for a selection process.
R72	S&C notifies a US of the outcome of a selection process.
R73	S&C notifies a US of an update on the status of an IN.
R74	S&C notifies a CO of an update on the status of an IN.
R75	S&C notifies a UN of an update on the status of an IN.
R76	S&C notifies a US of a comment on an IN.
R77	S&C notifies a CO of a comment on an IN.
R78	S&C notifies a UN of a comment on an IN.

Table 3.1: Requirements

Mapping on Goals

The following table presents the traceability matrix, demonstrating the alignment between specific requirements and corresponding goals to ensure coverage and accountability.

Requirement	G1	G2	G3	G4	G5
R1	✓	✓	✓	✓	✓
R2	✓	✓	✓	✓	✓
R3	✓	✓	✓	✓	✓
R4	✓	✓	✓	✓	✓
R5	✓	✓	✗	✗	✗
R6	✓	✓	✗	✗	✗
R7	✗	✓	✗	✗	✗
R8	✗	✓	✗	✗	✗
R9	✓	✗	✗	✗	✗
R10	✓	✓	✗	✗	✗
R11	✓	✓	✗	✗	✗

R12	✓	✗	✗	✗	✗
R13	✓	✗	✗	✗	✗
R14	✗	✓	✗	✗	✗
R15	✗	✓	✗	✗	✗
R16	✗	✗	✗	✗	✓
R17	✗	✗	✓	✗	✗
R18	✗	✗	✓	✗	✗
R19	✗	✗	✓	✗	✗
R20	✗	✗	✓	✗	✗
R21	✗	✗	✓	✗	✗
R22	✗	✗	✓	✗	✗
R23	✗	✗	✗	✓	✗
R24	✗	✗	✗	✓	✗
R25	✗	✗	✗	✓	✗
R26	✗	✗	✗	✓	✗
R27	✓	✓	✓	✓	✓
R28	✓	✓	✓	✓	✓
R29	✓	✓	✓	✓	✓
R30	✓	✓	✓	✓	✓
R31	✓	✓	✓	✓	✓
R32	✓	✓	✗	✗	✗
R33	✓	✓	✗	✗	✗
R34	✓	✓	✗	✗	✗
R35	✓	✓	✗	✗	✗
R36	✗	✓	✗	✗	✗
R37	✗	✓	✗	✗	✗
R38	✗	✓	✗	✗	✗
R39	✗	✗	✗	✗	✓
R40	✗	✗	✓	✗	✗

R41	X	X	✓	X	X
R42	X	X	✓	X	X
R43	X	X	✓	X	X
R44	X	X	✓	X	X
R45	X	X	✓	X	X
R46	X	X	X	✓	X
R47	X	X	X	✓	X
R48	X	X	X	✓	X
R49	X	X	X	✓	X
R50	X	X	X	✓	X
R51	X	X	X	✓	X
R52	X	X	X	✓	X
R53	X	X	X	✓	X
R54	X	X	X	✓	X
R55	X	X	X	✓	X
R56	X	X	X	✓	X
R57	X	X	X	✓	X
R58	X	X	X	✓	X
R59	X	X	X	✓	X
R60	X	X	X	✓	X
R61	X	✓	X	X	X
R62	X	✓	X	X	X
R63	X	X	✓	X	X
R64	X	X	✓	X	X
R65	X	X	✓	X	X
R66	X	X	✓	X	X
R67	X	X	✓	X	X
R68	X	X	✓	X	X
R69	X	X	✓	X	X

R70	X	X	✓	X	X
R71	X	X	✓	X	X
R72	X	X	✓	X	X
R73	X	X	X	✓	X
R74	X	X	X	✓	X
R75	X	X	X	✓	X
R76	X	X	X	✓	X
R77	X	X	X	✓	X
R78	X	X	X	✓	X

Table 3.2: Traceability matrix

3.2.2. Use Cases

This section explains the main identified use cases, presenting each one through a table and a sequence diagram. The tables outline entry conditions, event flow, exit conditions and exceptions, while the sequence diagrams illustrate the messages exchanged between entities and the functions called.

User Use Cases

The following use case diagram illustrates the key actions a generic user can perform.

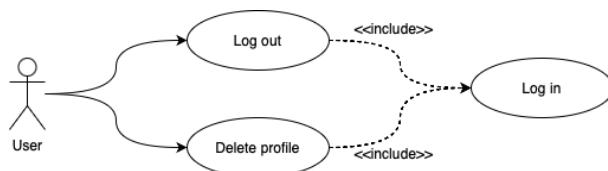


Figure 3.2: User use cases diagram

UC1. User Logs In	
Actors	User
Entry Condition	The user is signed up on S&C.
Event Flow	<ol style="list-style-type: none"> 1. The user navigates to the landing page. 2. S&C displays the landing page. 3. The user clicks the "Log In" button. 4. S&C displays the login page. 5. The user enters his email address and password. 6. The user clicks the "Log In" button. 7. S&C validates the credentials. 8. S&C displays the home page.
Exit Condition	The user is logged in and S&C displays the home page.
Exceptions	<ul style="list-style-type: none"> • The email address is invalid. • The password is wrong. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.3: Use case 1

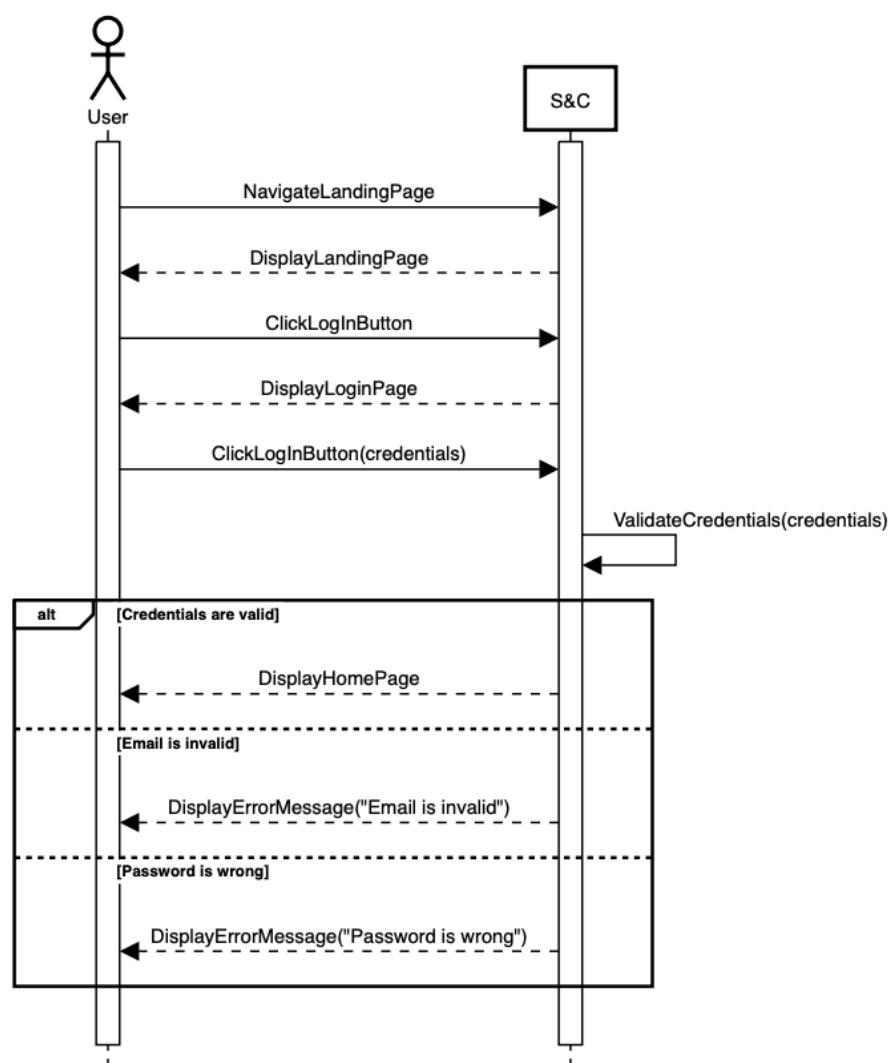


Figure 3.3: UC1 sequence diagram

UC2. User Logs Out	
Actors	User
Entry Condition	The user is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The user clicks the "My Profile" button. 2. S&C displays the profile page. 3. The user clicks the "Log Out" button. 4. S&C displays the landing page.
Exit Condition	The user is logged out and S&C displays the landing page.
Exceptions	None.

Table 3.4: Use case 2

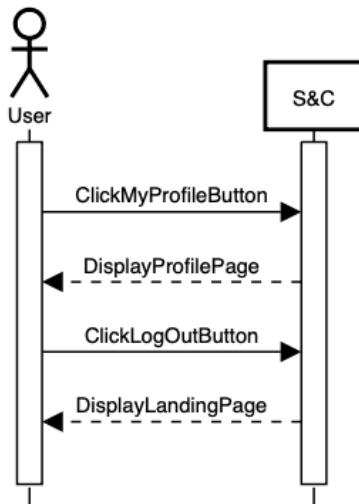


Figure 3.4: UC2 sequence diagram

UC3. User Deletes Profile	
Actors	User, EP
Entry Condition	The user is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The user clicks the "My Profile" button. 2. S&C displays the profile page. 3. The user clicks the "Delete Profile" button. 4. S&C sends a confirmation email to the user via the EP. 5. The user clicks the confirmation link in the email. 6. S&C displays the landing page.
Exit Condition	The profile is deleted and S&C displays the landing page.
Exceptions	None.

Table 3.5: Use case 3

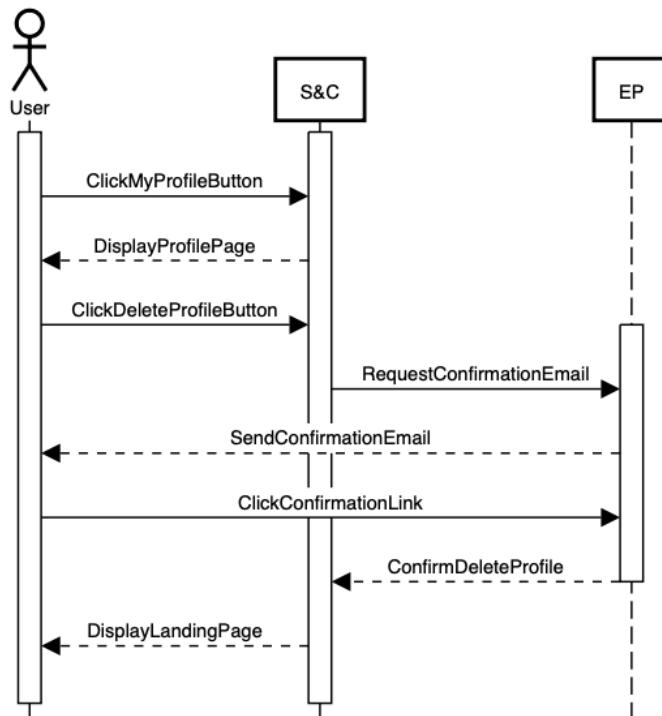


Figure 3.5: UC3 sequence diagram

Student Use Cases

The following use case diagram outlines the main interactions a student can have.

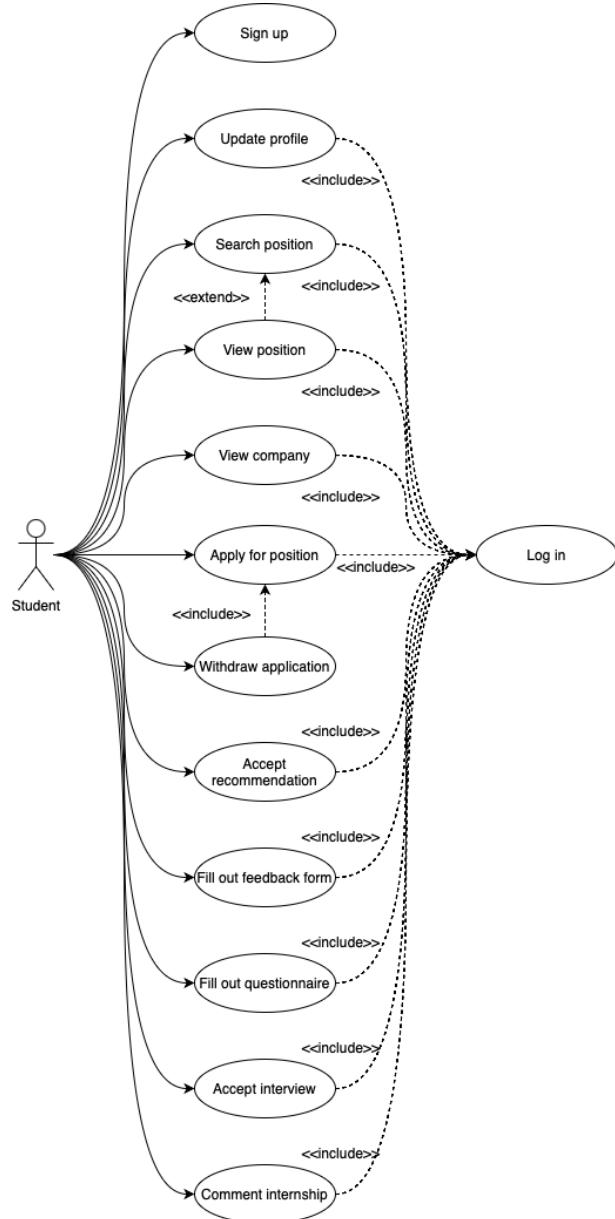


Figure 3.6: Student use cases diagram

UC4. Student Signs Up	
Actors	US, EP
Entry Condition	The US is not signed up on S&C.
Event Flow	<ol style="list-style-type: none"> 1. The US navigates to the landing page. 2. S&C displays the landing page. 3. The US clicks the "Sign Up as a Student" button. 4. S&C displays the signup page. 5. The US enters their name, surname, institutional email address, password, and confirms the password. 6. The US can upload a CV and enter preferences. 7. The US can tick the "Keep Me Updated" field. 8. The US clicks the "Sign Up" button. 9. S&C validates the fields. 10. S&C sends a confirmation email to the US via the EP. 11. The US clicks the confirmation link in the email. 12. S&C displays the login page.
Exit Condition	The US is signed up and S&C displays the login page.
Exceptions	<ul style="list-style-type: none"> • The email is not a valid institutional email address. • The email is already linked to another profile. • The password is shorter than 8 characters. • The passwords do not match. • Another field is invalid. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.6: Use case 4

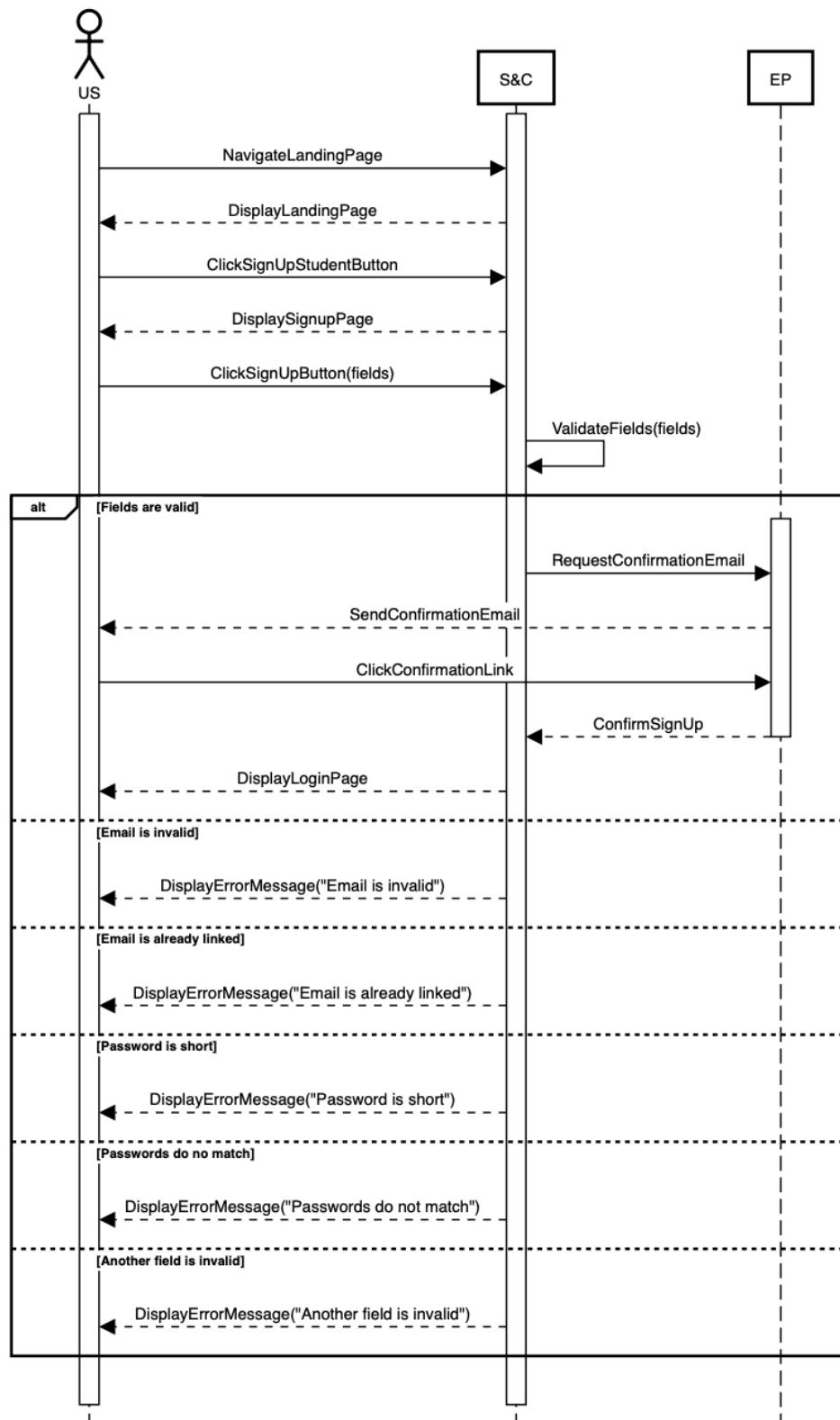


Figure 3.7: UC4 sequence diagram

UC5. Student Updates Profile	
Actors	US
Entry Condition	The US is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the "My Profile" button. 2. S&C displays the profile page. 3. The US clicks the "Update Profile" button. 4. S&C displays the profile editor. 5. The US updates the desired fields. 6. The US clicks the "Save Profile" button. 7. S&C validates the fields. 8. S&C displays the profile page.
Exit Condition	The profile is updated and S&C displays the profile page.
Exceptions	<ul style="list-style-type: none"> • The email is not a valid institutional email address. • The email is already linked to another profile. • The password is shorter than 8 characters. • The passwords do not match. • Another field is invalid. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.7: Use case 5

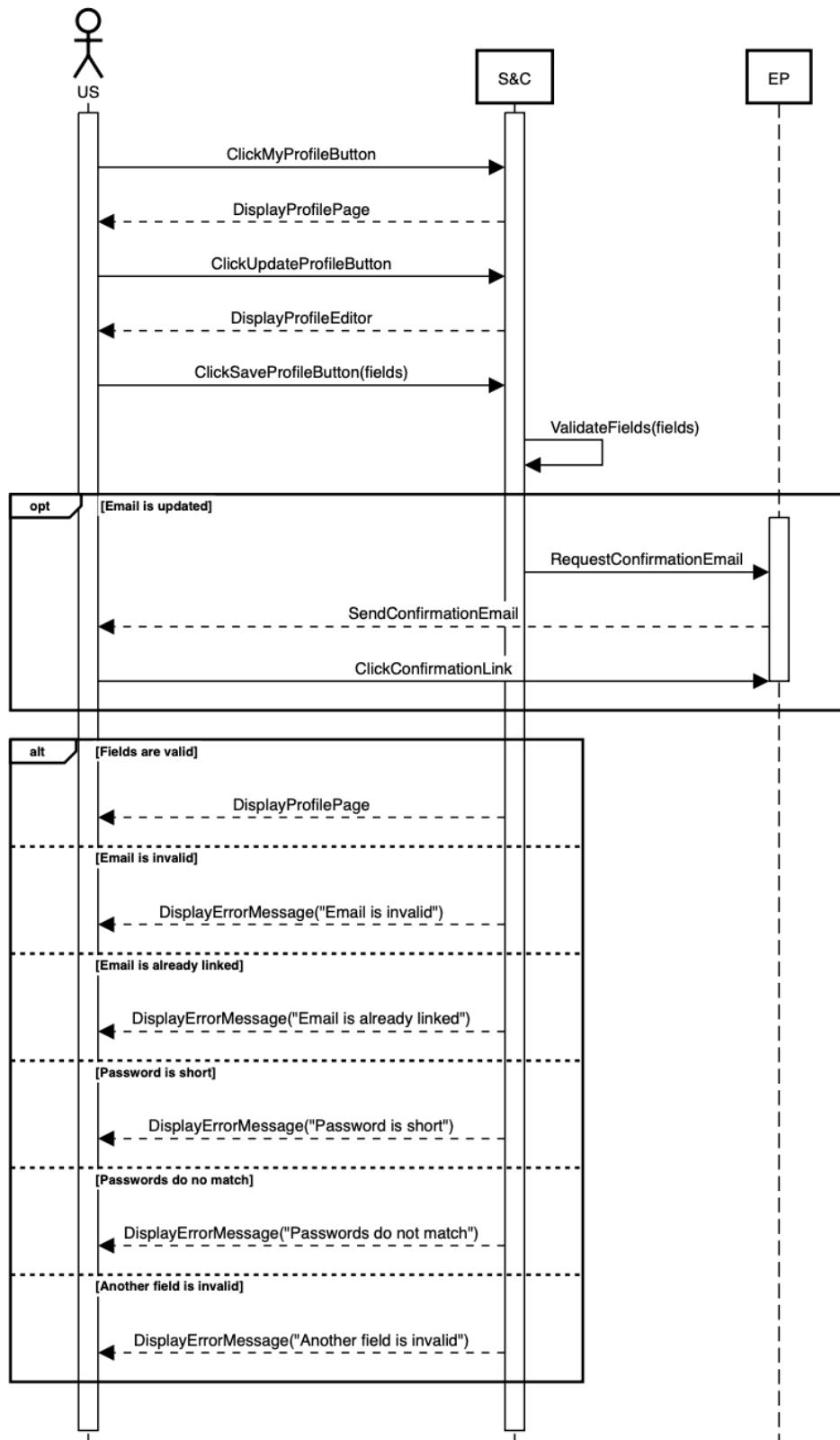


Figure 3.8: UC5 sequence diagram

UC6. Student Searches Position	
Actors	US
Entry Condition	The US is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The US enters a keyword in the search bar. 2. The US clicks the "Search" button. 3. S&C computes the results. 4. S&C displays the results page.
Exit Condition	S&C displays the results page.
Exceptions	<ul style="list-style-type: none"> • No results are computed. In this case, S&C displays a descriptive error message.

Table 3.8: Use case 6

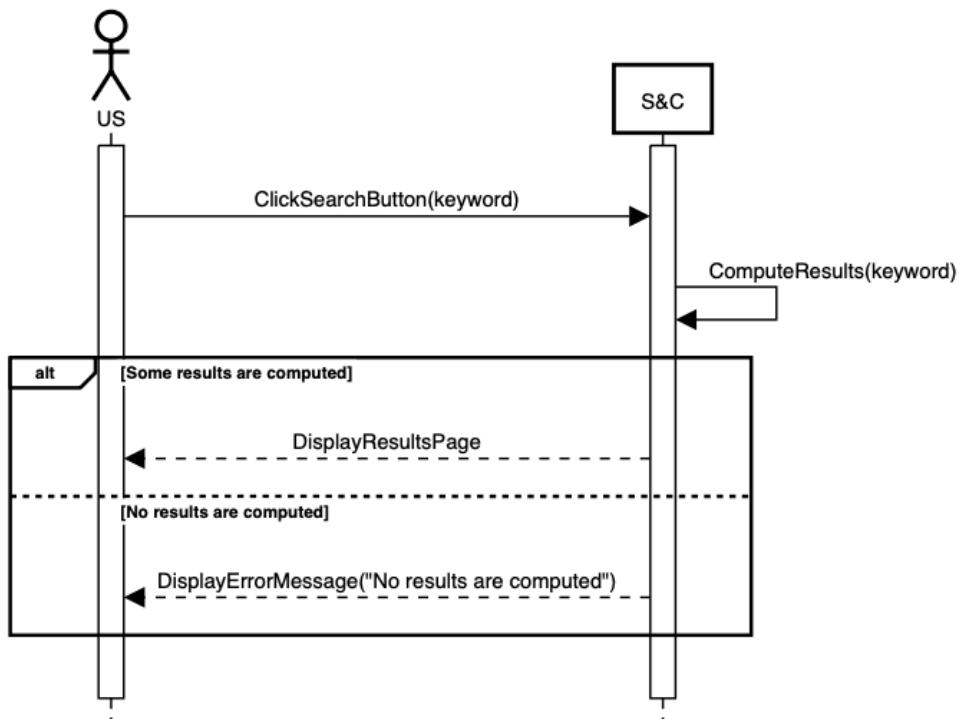


Figure 3.9: UC6 sequence diagram

UC7. Student Views Position	
Actors	US
Entry Condition	The US is logged in and S&C displays the PO name.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the PO name. 2. S&C displays the PO page.
Exit Condition	S&C displays the PO page.
Exceptions	<ul style="list-style-type: none"> • The PO has been removed. • The CO has been deleted. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.9: Use case 7

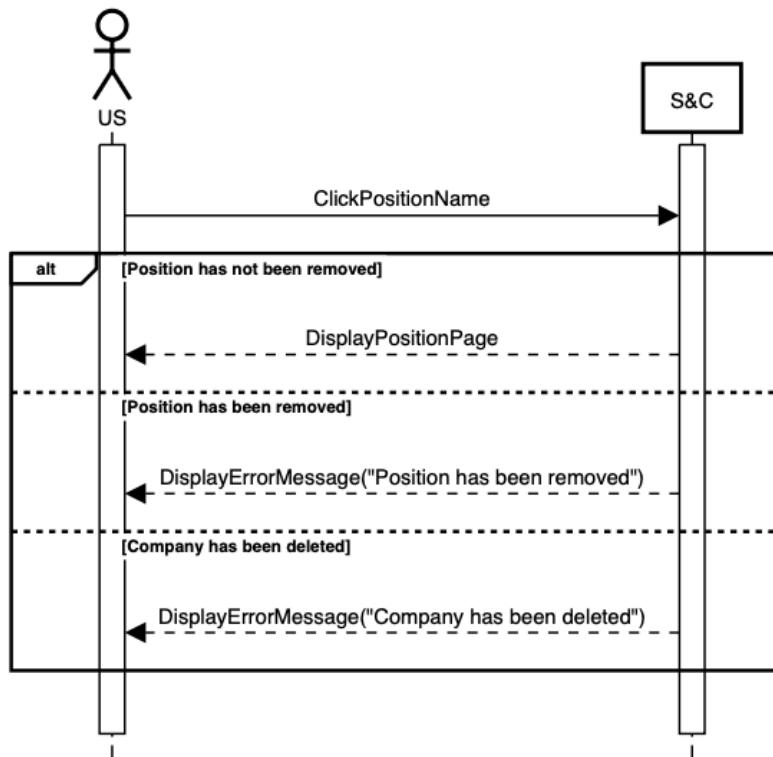


Figure 3.10: UC7 sequence diagram

UC8. Student Views Company	
Actors	US
Entry Condition	The US is logged in and S&C displays the CO name.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the CO name. 2. S&C displays the CO page.
Exit Condition	S&C displays the CO page.
Exceptions	<ul style="list-style-type: none"> • The CO has been deleted. In this case, S&C displays a descriptive error message.

Table 3.10: Use case 8

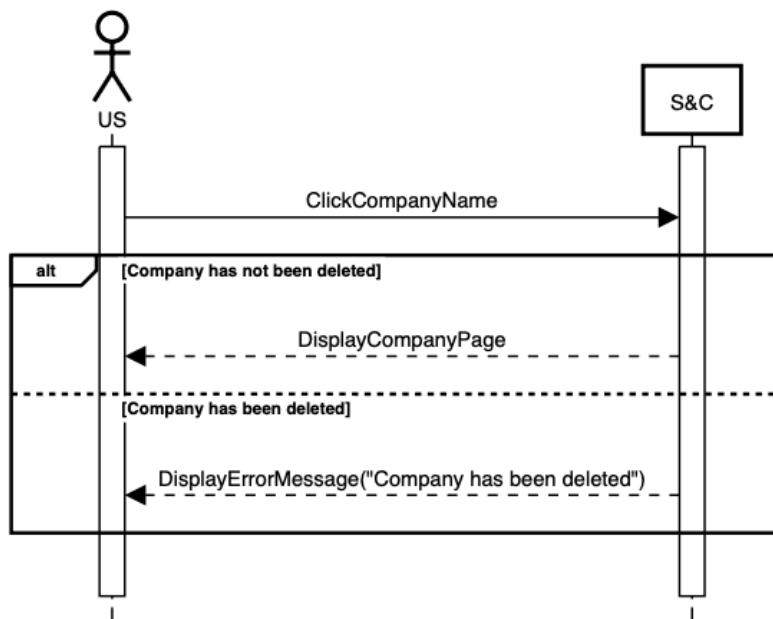


Figure 3.11: UC8 sequence diagram

UC9. Student Applies for Position	
Actors	US, CO, EP
Entry Condition	The US is logged in and S&C displays the PO page.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the "Apply" button. 2. S&C displays the application page. 3. S&C sends a confirmation email to the US via the EP. 4. S&C sends a notification email to the CO via the EP.
Exit Condition	The US has applied for the PO, the CO has been notified and S&C displays the application page.
Exceptions	<ul style="list-style-type: none"> • The US has already applied. • The PO has been removed. • The CO has been deleted. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.11: Use case 9

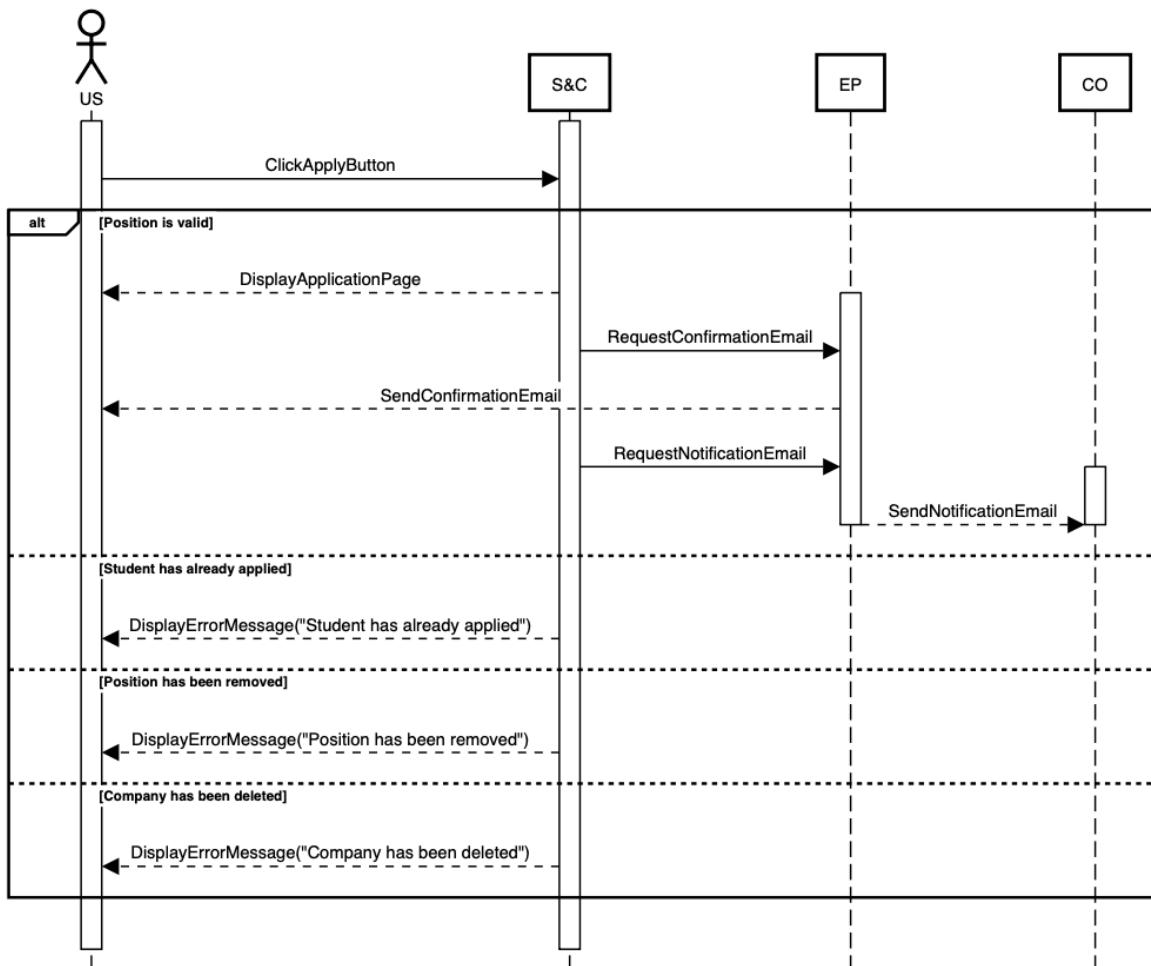


Figure 3.12: UC9 sequence diagram

UC10. Student Withdraws Application	
Actors	US
Entry Condition	The US is logged in, has applied for the PO and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the "My Applications" button. 2. S&C displays the applications page. 3. The US clicks the application name. 4. S&C displays the application page. 5. The US clicks the "Withdraw" button. 6. S&C displays the "My Applications" page.
Exit Condition	The application is withdrawn and S&C displays the "My Applications" page.
Exceptions	None.

Table 3.12: Use case 10

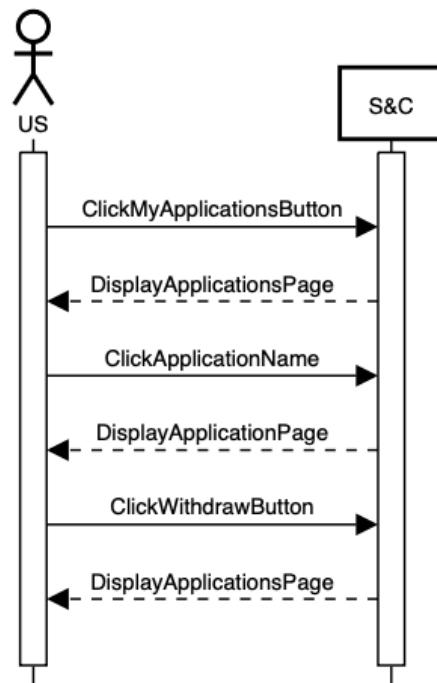


Figure 3.13: UC10 sequence diagram

UC11. Student Accepts Recommendation	
Actors	US, CO, EP
Entry Condition	The US is logged in, has uploaded a CV, entered preferences and ticked the "Keep Me Updated" field.
Event Flow	<ol style="list-style-type: none"> 1. S&C sends a notification email to the US via the EP. 2. The US clicks the "View Position" link on the email. 3. S&C displays the PO page. 4. The US clicks the "Accept" button. 5. S&C displays the application page. 6. S&C checks if a match is identified. 7. S&C sends a notification email to the US via the EP. 8. S&C sends a notification email to the CO via the EP.
Exit Condition	The recommendation is accepted and S&C displays the application page.
Exceptions	<ul style="list-style-type: none"> • The recommendation has already been resolved. • The PO has been removed. • The CO has been deleted. <p>In all cases, S&C displays the home page and a descriptive error message.</p>

Table 3.13: Use case 11

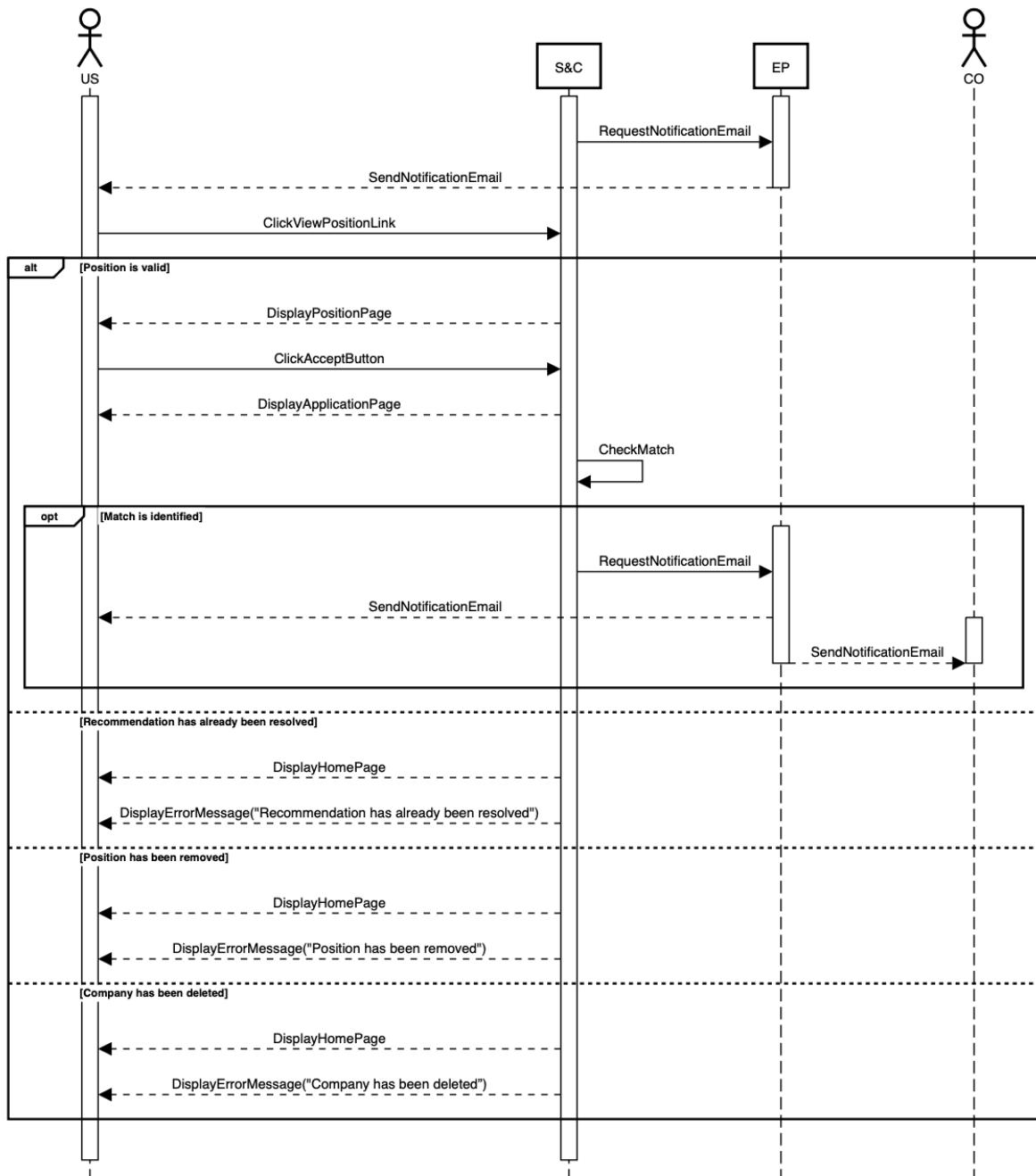


Figure 3.14: UC11 sequence diagram

UC12. Student Fills Out Feedback Form	
Actors	US
Entry Condition	The US is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none">1. The US clicks the "Give Feedback" button.2. S&C displays the feedback form.3. The US enters the fields.4. The US clicks the "Submit" button.5. S&C validates the fields.6. S&C displays the home page.
Exit Condition	The feedback form is submitted and S&C displays the home page.
Exceptions	<ul style="list-style-type: none">• A field is invalid. In this case, S&C displays a descriptive error message.

Table 3.14: Use case 12

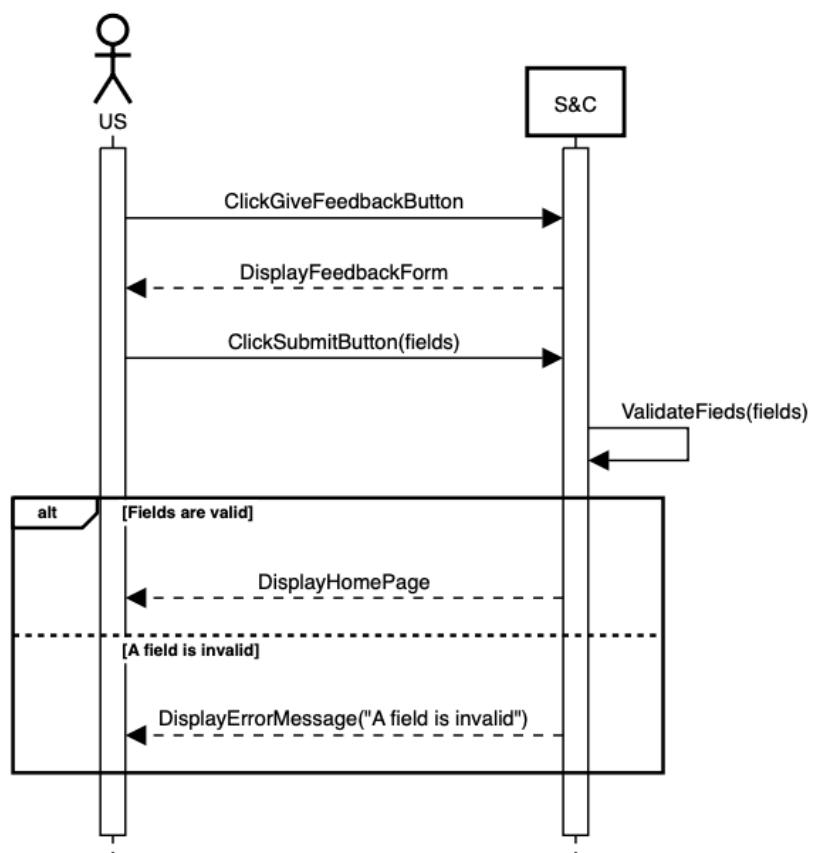


Figure 3.15: UC12 sequence diagram

UC13. Student Fills Out Questionnaire	
Actors	US, CO, EP
Entry Condition	The US is logged in, has received a notification on an added questionnaire and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the "My Applications" button. 2. S&C displays the applications page. 3. The US clicks the application name. 4. S&C displays the application page. 5. The US clicks the "View Questionnaire" button. 6. S&C displays the questionnaire. 7. The US enters the fields. 8. The US clicks the "Submit" button. 9. S&C validates the fields. 10. S&C displays the application page. 11. S&C sends a notification email to the CO via the EP.
Exit Condition	The questionnaire is submitted and S&C displays the application page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. <p>In this case, S&C displays a descriptive error message.</p>

Table 3.15: Use case 13

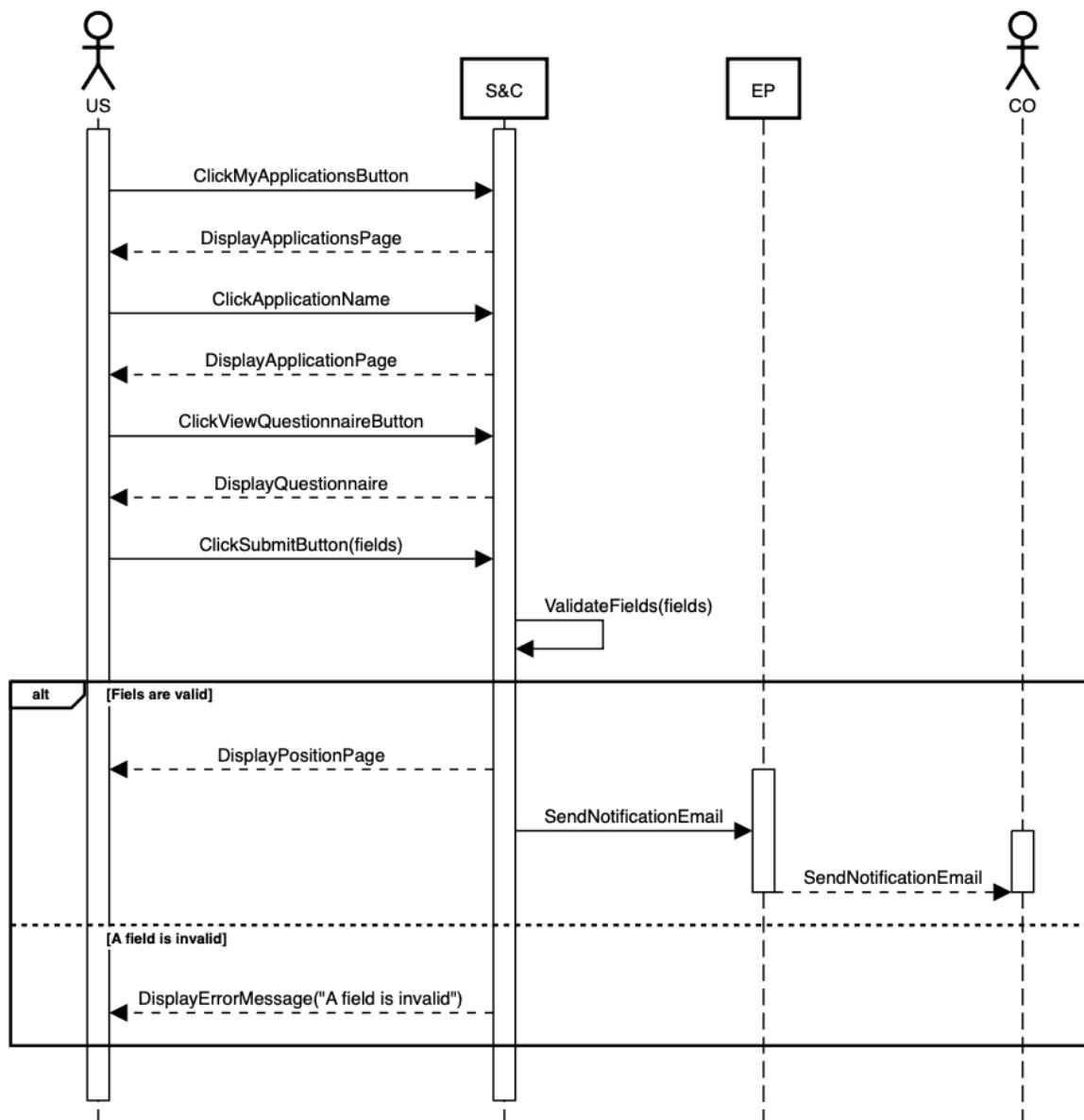


Figure 3.16: UC13 sequence diagram

UC14. Student Accepts Interview	
Actors	US, CO, EP
Entry Condition	The US is logged in, has received a notification on a scheduled interview and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the "My Applications" button. 2. S&C displays the applications page. 3. The US clicks the application name. 4. S&C displays the application page. 5. The US clicks the "View Interview" button. 6. S&C displays the interview page. 7. The US clicks the "Accept" button. 8. S&C displays the application page. 9. S&C sends a notification email to the CO via the EP.
Exit Condition	The interview is scheduled and S&C displays the application page.
Exceptions	None.

Table 3.16: Use case 14

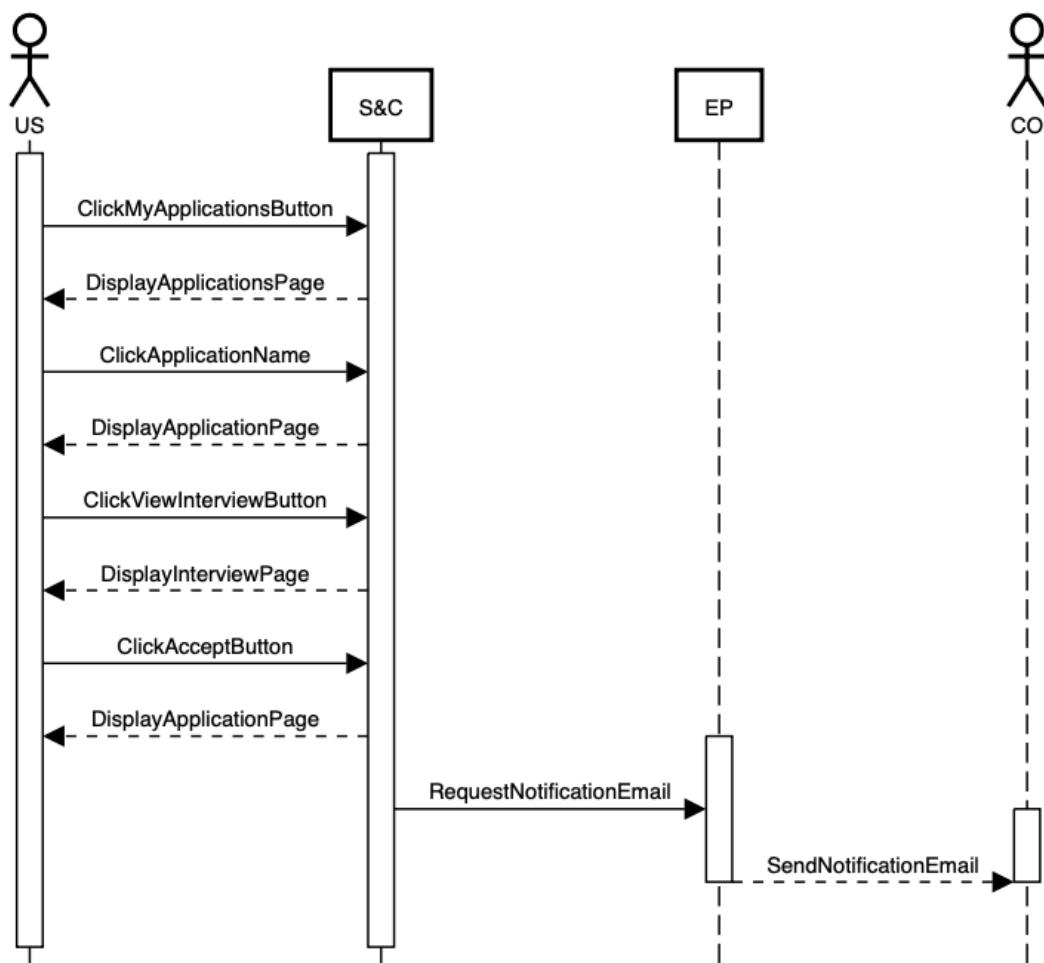


Figure 3.17: UC14 sequence diagram

UC15. Student Comments Internship	
Actors	US, CO, UN, EP
Entry Condition	The US is logged in, doing an IN and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The US clicks the "My Internships" button. 2. S&C displays the internships page. 3. The US clicks the IN name. 4. S&C displays the IN page. 5. The US clicks the "Write a Comment" button. 6. S&C displays the comment form. 7. The US enters the fields. 8. The US clicks the "Send" button. 9. S&C validates the fields. 10. S&C displays the IN page. 11. S&C sends a notification email to the CO via the EP. 12. S&C sends a notification email to the UN via the EP.
Exit Condition	The comment is sent, the CO and the UN are notified and S&C displays the IN page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. In this case, S&C displays a descriptive error message.

Table 3.17: Use case 15

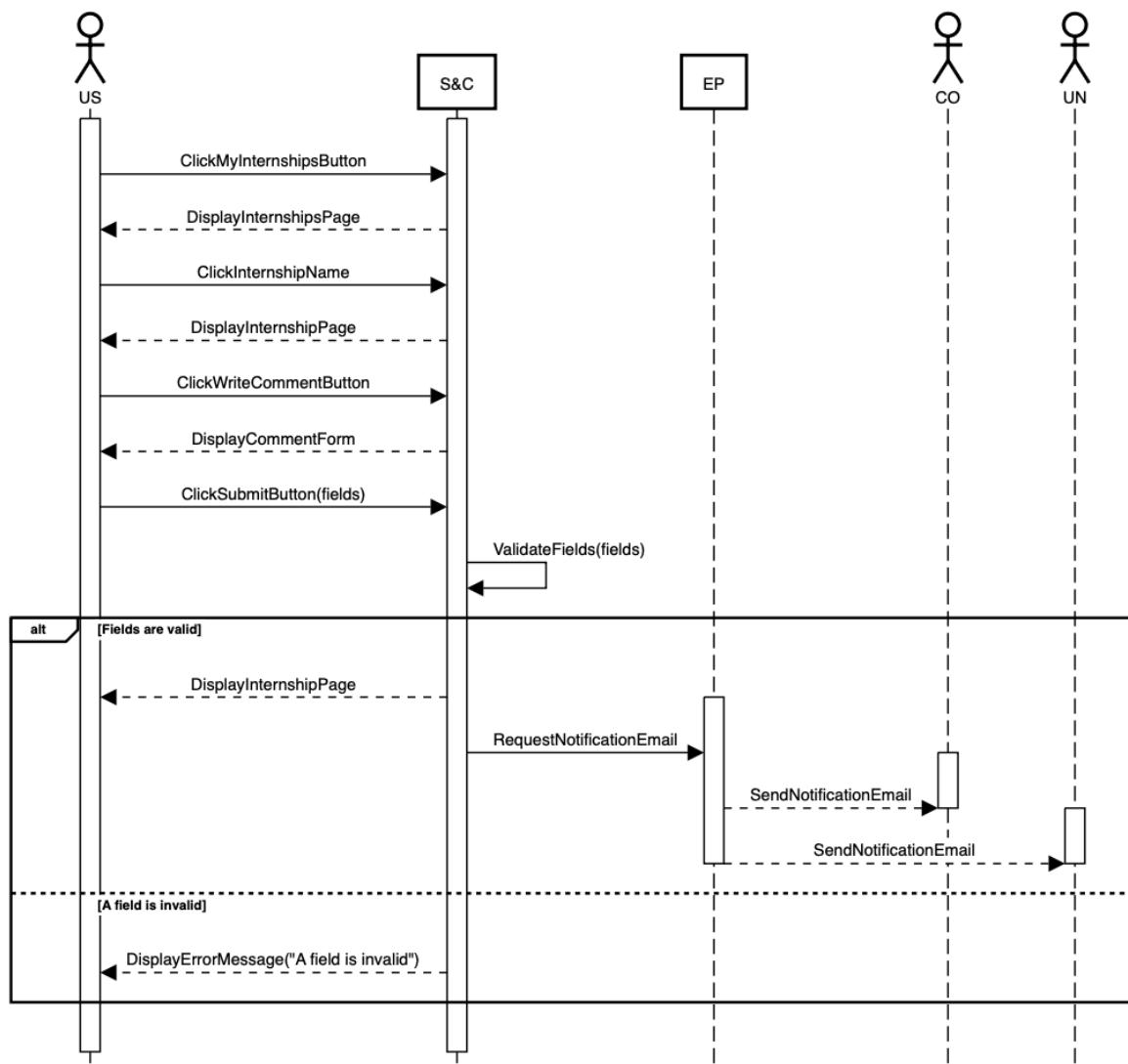


Figure 3.18: UC15 sequence diagram

Company Use Cases

The use case diagram below demonstrates the primary operations a company can perform.

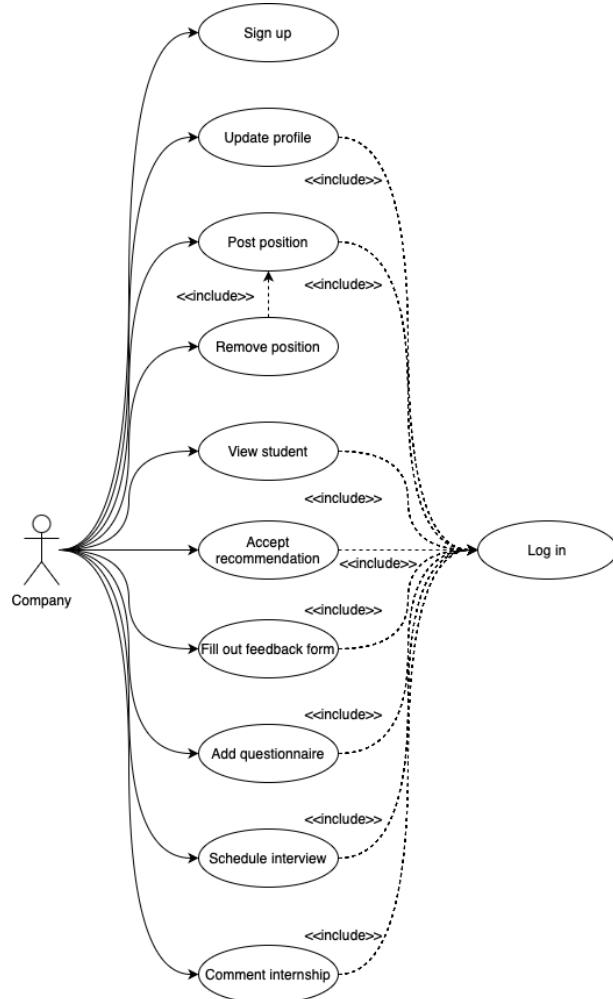


Figure 3.19: Company use cases diagram

UC16. Company Signs Up	
Actors	CO, EP
Entry Condition	The CO is not signed up on S&C.
Event Flow	<ol style="list-style-type: none"> 1. The CO navigates to the landing page. 2. S&C displays the landing page. 3. The CO clicks the "Sign Up as a Company" button. 4. S&C displays the signup page. 5. The CO enters their name, email address, field, password, and confirms the password. 6. The CO can tick the "Keep Me Updated" field. 7. The CO clicks the "Sign Up" button. 8. S&C validates the fields. 9. S&C sends a confirmation email to the CO via the EP. 10. The CO clicks the confirmation link in the email. 11. S&C displays the login page.
Exit Condition	The CO is signed up and S&C displays the login page.
Exceptions	<ul style="list-style-type: none"> • The email is already linked to another profile. • The password is shorter than 8 characters. • The passwords do not match. • Another field is invalid. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.18: Use case 16

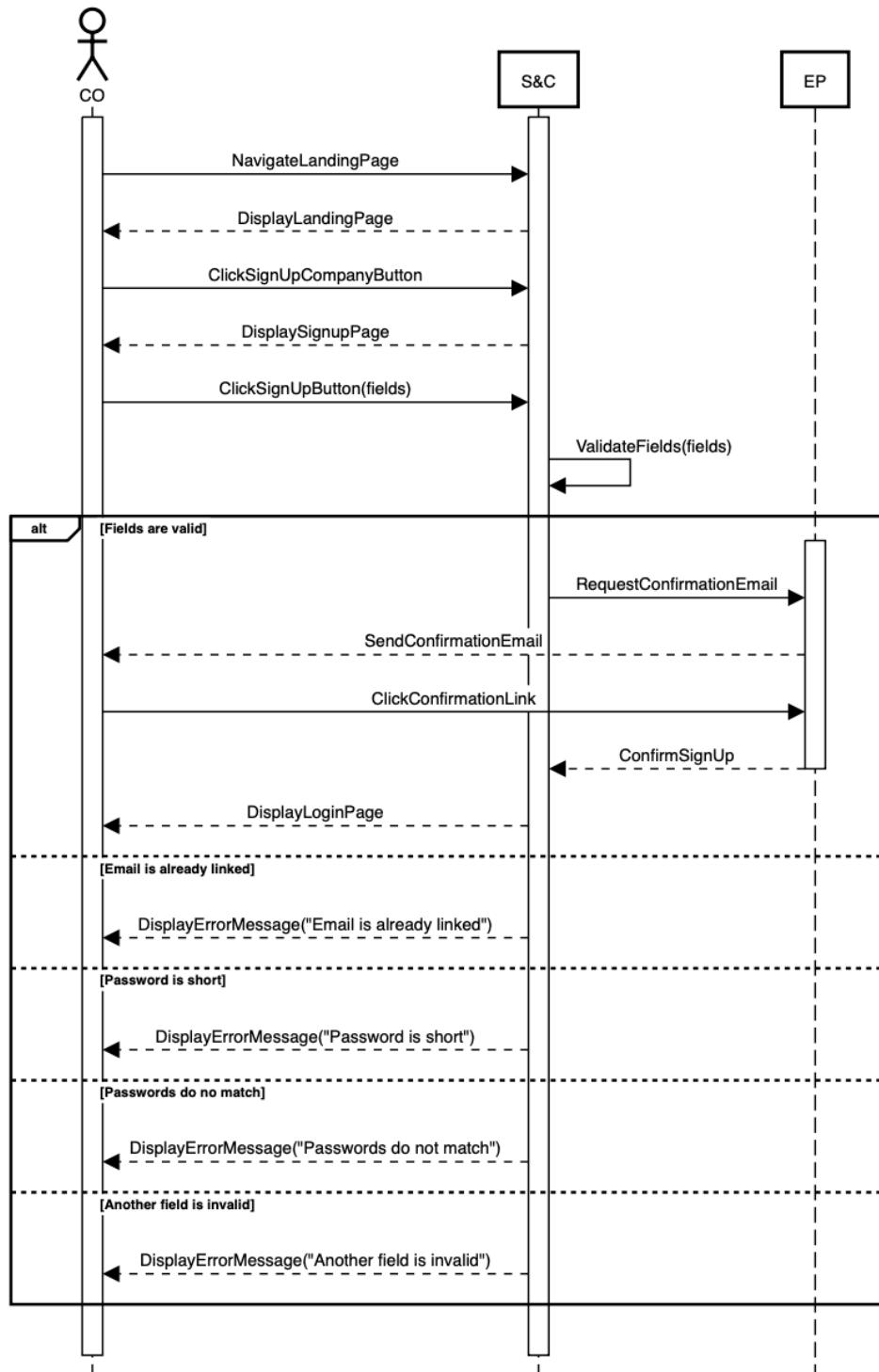


Figure 3.20: UC16 sequence diagram

UC17. Company Updates Profile	
Actors	CO
Entry Condition	The CO is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks the "My Profile" button. 2. S&C displays the profile page. 3. The CO clicks the "Update Profile" button. 4. S&C displays the profile editor. 5. The CO updates the desired fields. 6. The CO clicks the "Save Profile" button. 7. S&C validates the fields. 8. S&C displays the profile page.
Exit Condition	The profile is updated and S&C displays the profile page.
Exceptions	<ul style="list-style-type: none"> • The email is already linked to another profile. • The password is shorter than 8 characters. • The passwords do not match. • Another field is invalid. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.19: Use case 17

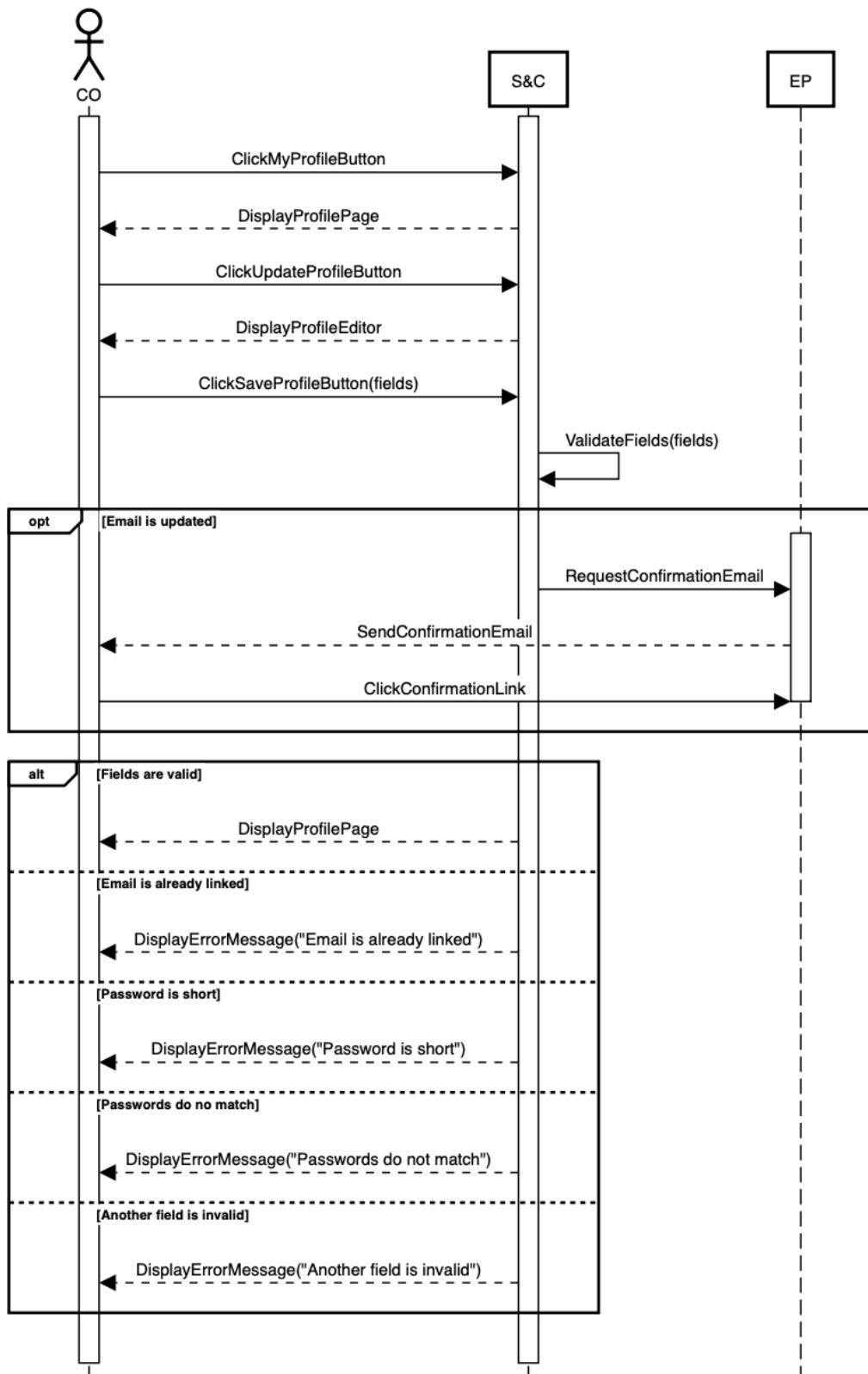


Figure 3.21: UC17 sequence diagram

UC18. Company Posts Position	
Actors	CO
Entry Condition	The CO is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks the "Post a Position" button. 2. S&C displays the PO posting page. 3. The CO enters the PO name, domain, project, tasks, and terms. 4. The CO clicks the "Post" button. 5. S&C validates the fields. 6. S&C displays the home page.
Exit Condition	The PO is posted and S&C displays the home page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. In this case, S&C displays a descriptive error message.

Table 3.20: Use case 18

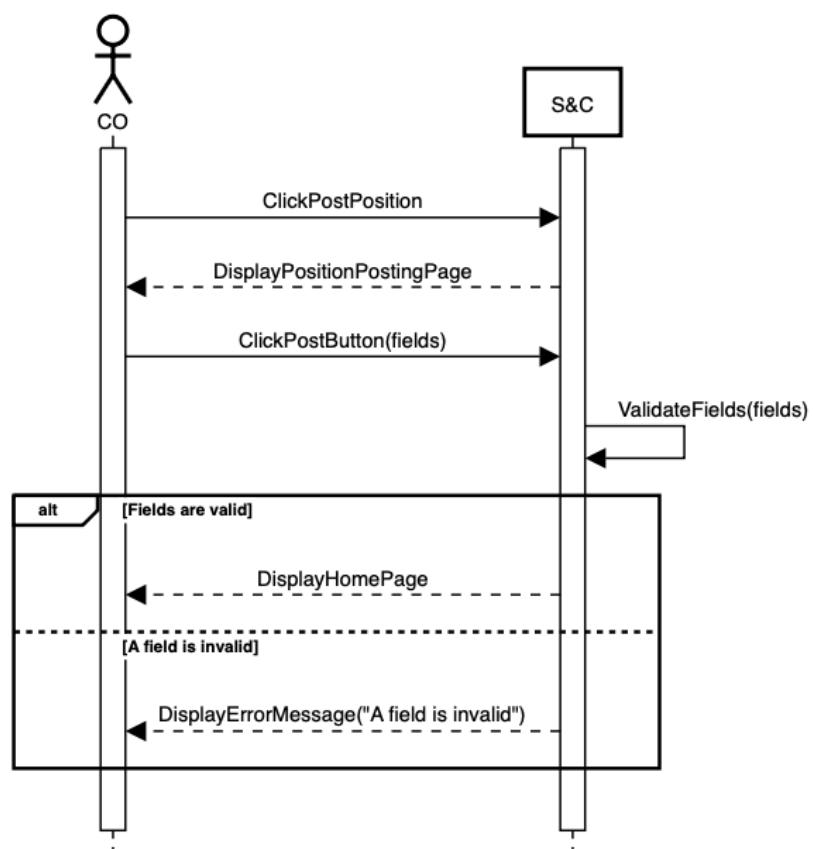


Figure 3.22: UC18 sequence diagram

UC19. Company Removes Position	
Actors	CO
Entry Condition	The CO is logged in, has posted the PO, and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks the "My Positions" button. 2. S&C displays the POs page. 3. The CO clicks the PO name. 4. S&C displays the PO page. 5. The CO clicks the "Remove Position" button. 6. S&C displays the POs page.
Exit Condition	The PO is removed and S&C displays the POs page.
Exceptions	None.

Table 3.21: Use case 19

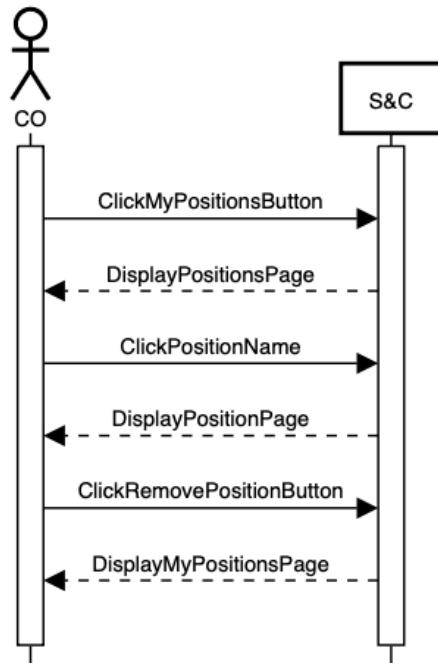


Figure 3.23: UC19 sequence diagram

UC20. Company Views Student	
Actors	CO
Entry Condition	The CO is logged in and S&C displays the US name.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks the US name. 2. S&C displays the US page.
Exit Condition	S&C displays the US page.
Exceptions	<ul style="list-style-type: none"> • The US has been deleted. In this case, S&C displays a descriptive error message.

Table 3.22: Use case 20

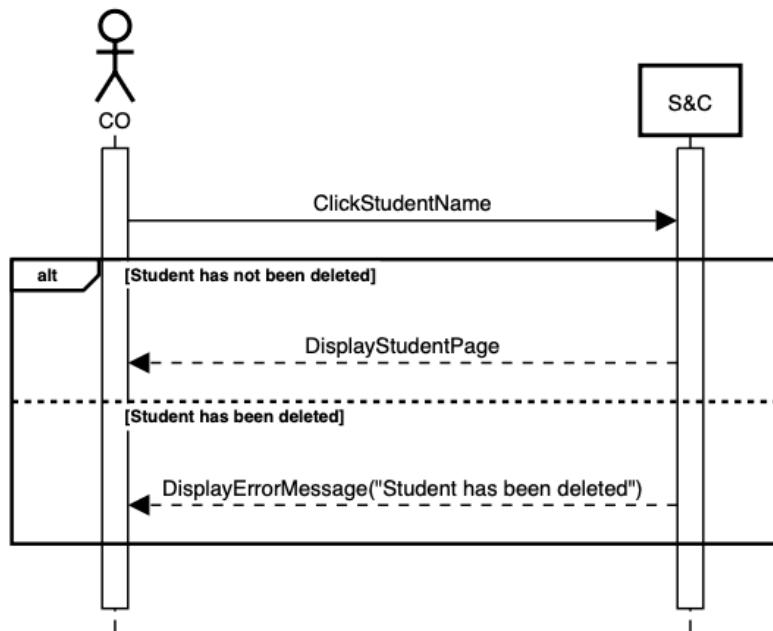


Figure 3.24: UC20 sequence diagram

UC21. Company Accepts Recommendation	
Actors	CO, US, EP
Entry Condition	The CO is logged in and has ticked the "Keep Me Updated" field.
Event Flow	<ol style="list-style-type: none"> 1. S&C sends a notification email to the CO via the EP. 2. The CO clicks the "View Student" link on the email. 3. S&C displays the US page. 4. The CO clicks the "Accept" button. 5. S&C displays the PO page. 6. S&C checks if a match is identified. 7. S&C sends a notification email to the CO via the EP. 8. S&C sends a notification email to the US via the EP.
Exit Condition	The recommendation is accepted and S&C displays the PO page.
Exceptions	<ul style="list-style-type: none"> • The recommendation has already been resolved. • The US has been deleted. • The PO has been deleted. <p>In all cases, S&C displays the home page and a descriptive error message.</p>

Table 3.23: Use case 21

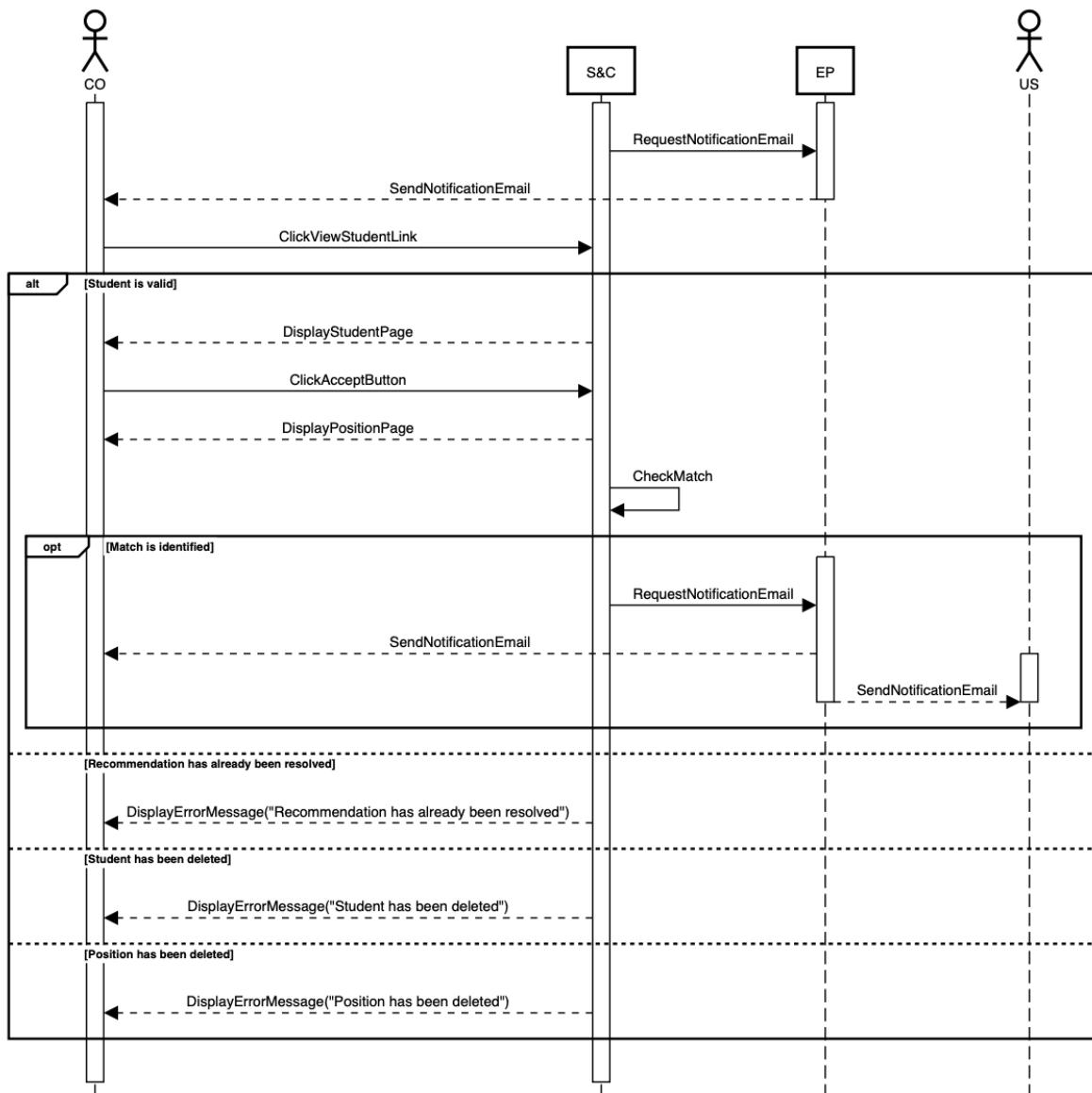


Figure 3.25: UC21 sequence diagram

UC22. Company Fills Out Feedback Form	
Actors	CO
Entry Condition	The CO is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks the "Give Feedback" button. 2. S&C displays the feedback form. 3. The CO enters the fields. 4. The CO clicks the "Submit" button. 5. S&C validates the fields. 6. S&C displays the home page.
Exit Condition	The feedback form is submitted and S&C displays the home page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. In this case, S&C displays a descriptive error message.

Table 3.24: Use case 22

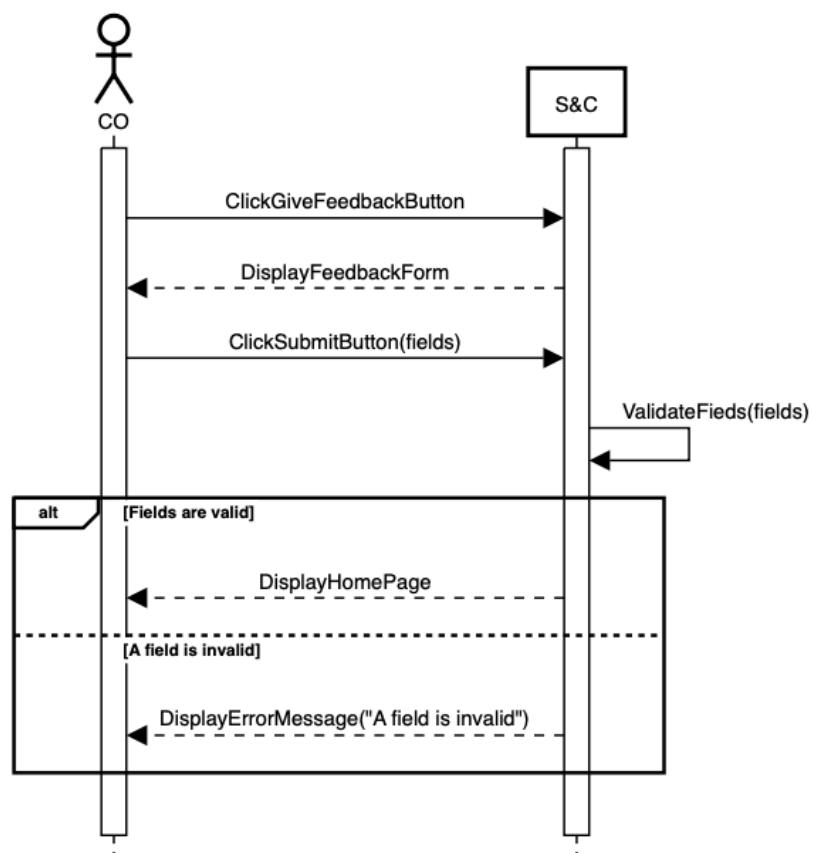


Figure 3.26: UC22 sequence diagram

UC23. Company Adds Questionnaire	
Actors	CO, US, EP
Entry Condition	The CO is logged in, has posted the PO and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks on the "My Positions" button. 2. S&C displays the POs page. 3. The CO clicks on the PO name. 4. S&C displays the PO page. 5. The CO clicks on the "Add Questionnaire" button. 6. S&C displays the questionnaire editor. 7. The CO edits the questionnaire. 8. The CO clicks the "Add" button. 9. S&C displays the PO page. 10. S&C sends a notification email to USs via the EP.
Exit Condition	The questionnaire is added, USs are notified and S&C displays the PO page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. In this case, S&C displays a descriptive error message.

Table 3.25: Use case 23

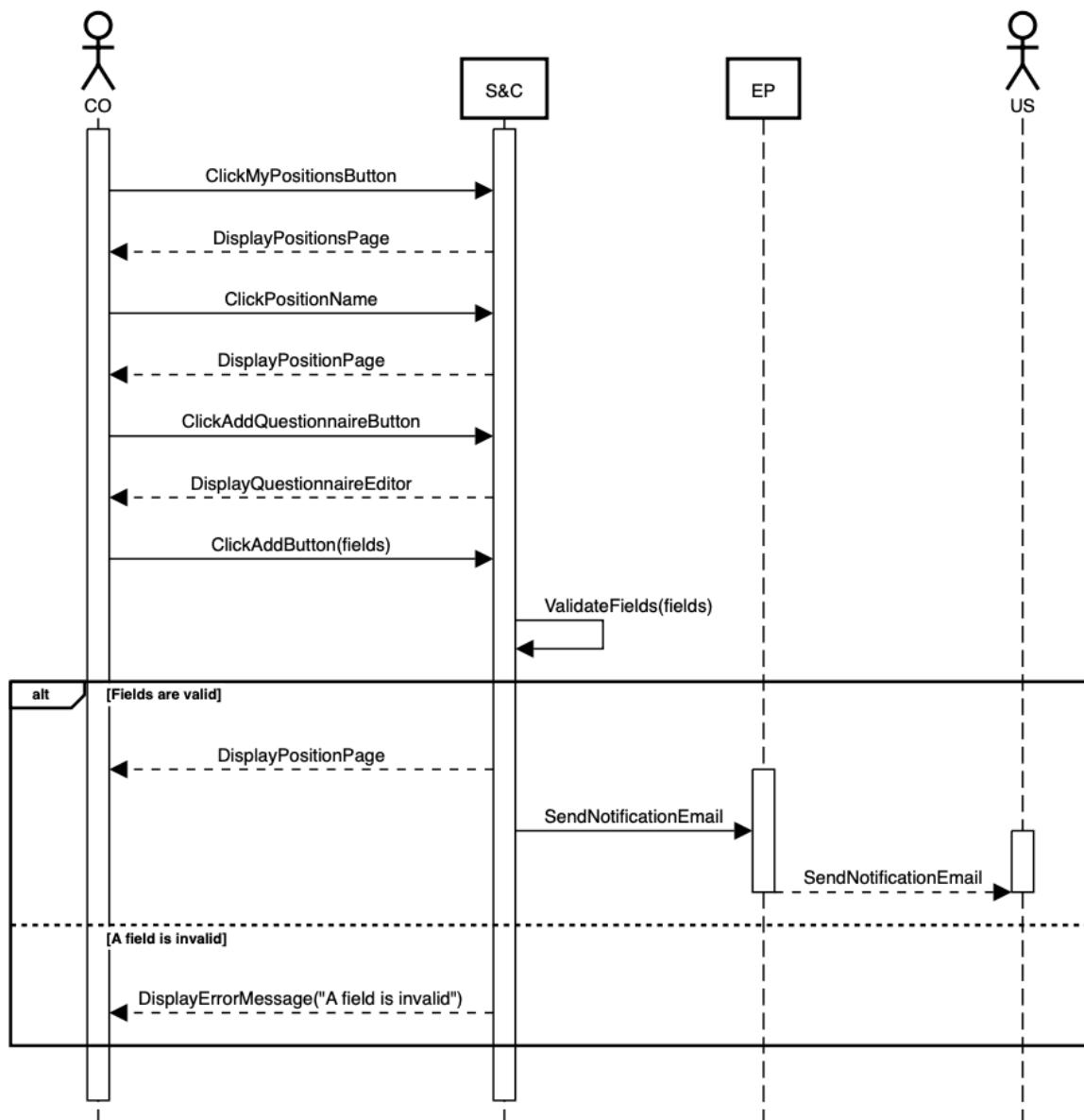


Figure 3.27: UC23 sequence diagram

UC24. Company Schedules Interview	
Actors	CO, US, EP
Entry Condition	The CO is logged in, has posted the PO and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks on the "My Positions" button. 2. S&C displays the POs page. 3. The CO clicks on the PO name. 4. S&C displays the PO page. 5. The CO clicks on the US name. 6. S&C displays the US page. 7. The CO clicks on the "Schedule an Interview" button. 8. S&C displays the interview form. 9. The CO enters the date and the mode. 10. The CO clicks the "Schedule" button. 11. S&C validates the fields. 12. S&C displays the US page. 13. S&C sends a notification email to the US via the EP.
Exit Condition	The interview is scheduled, the US is notified and S&C displays the US page.
Exceptions	<ul style="list-style-type: none"> • The date is in the past. • The mode is invalid. <p>In this case, S&C displays a descriptive error message.</p>

Table 3.26: Use case 24

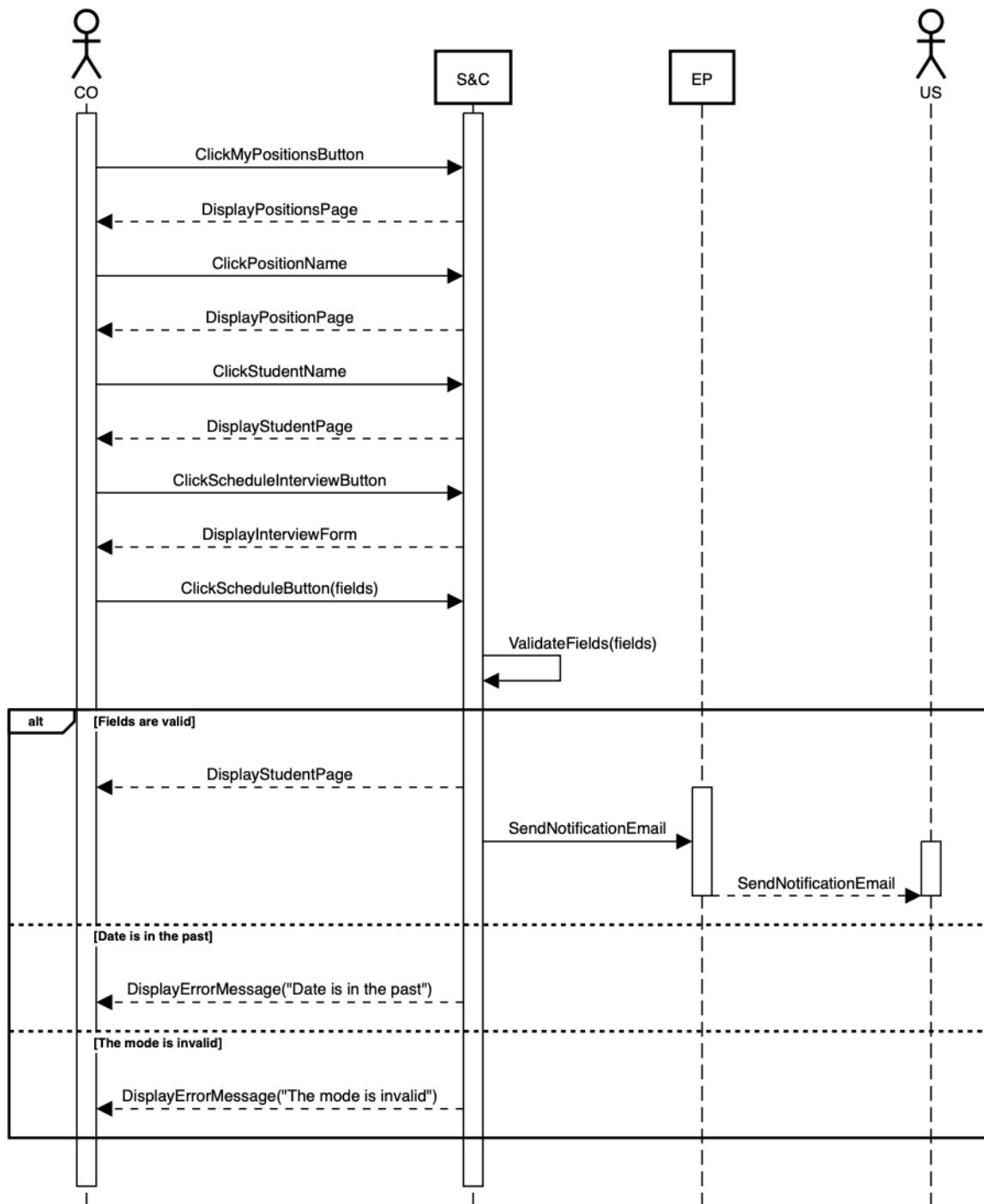


Figure 3.28: UC24 sequence diagram

UC25. Company Comments Internship	
Actors	CO, US, UN, EP
Entry Condition	The CO is logged in, hosting an IN and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The CO clicks the "My Internships" button. 2. S&C displays the internships page. 3. The CO clicks the IN name. 4. S&C displays the IN page. 5. The CO clicks the "Write a Comment" button. 6. S&C displays the comment form. 7. The CO enters the fields. 8. The CO clicks the "Send" button. 9. S&C validates the fields. 10. S&C displays the IN page. 11. S&C sends a notification email to the US via the EP. 12. S&C sends a notification email to the UN via the EP.
Exit Condition	The comment is sent, the US and the UN are notified and S&C displays the IN page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. In this case, S&C displays a descriptive error message.

Table 3.27: Use case 25

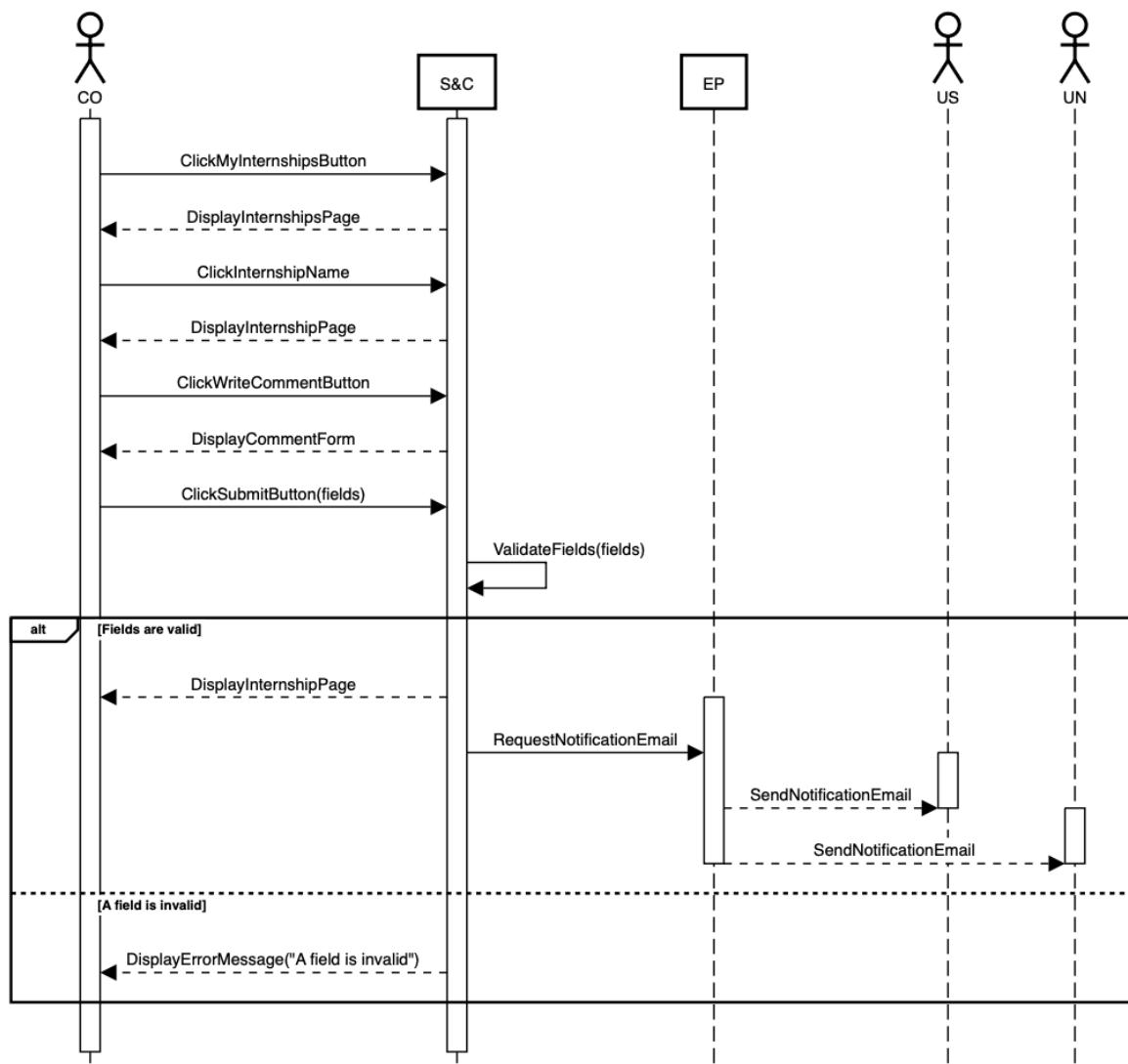


Figure 3.29: UC25 sequence diagram

University Use Cases

The following use case diagram depicts the main activities a university can carry out.

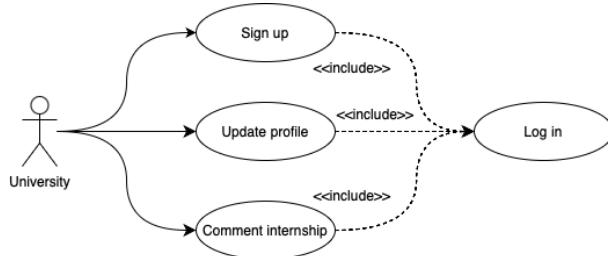


Figure 3.30: University use cases diagram

UC26. University Signs Up	
Actors	UN, EP
Entry Condition	The UN is not signed up on S&C.
Event Flow	<ol style="list-style-type: none"> 1. The UN navigates to the landing page. 2. S&C displays the landing page. 3. The UN clicks the "Sign Up as a University" button. 4. S&C displays the signup page. 5. The UN enters their name, institutional email address, password and confirms the password. 6. The UN can tick the "Keep Me Updated" field. 7. The UN clicks the "Sign Up" button. 8. S&C validates the fields. 9. S&C sends a confirmation email to the UN via the EP. 10. The UN clicks the confirmation link in the email. 11. S&C displays the login page.
Exit Condition	The UN is signed up and S&C displays the login page.
Exceptions	<ul style="list-style-type: none"> • The email is not a valid institutional email address. • The email is already linked to another profile. • The password is shorter than 8 characters. • The passwords do not match. • Another field is invalid. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.28: Use case 26

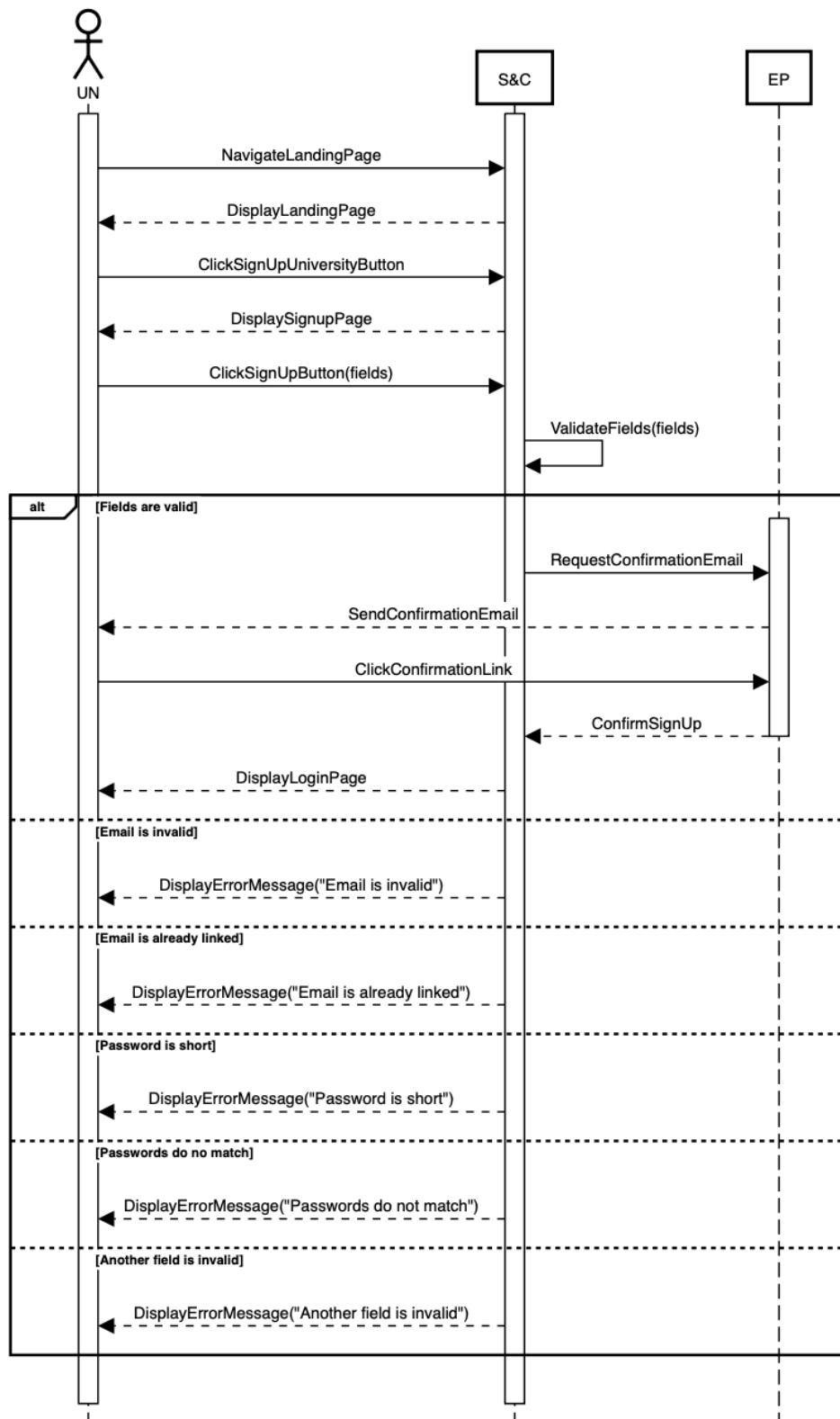


Figure 3.31: UC26 sequence diagram

UC27. University Updates Profile	
Actors	UN
Entry Condition	The UN is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The UN clicks the "My Profile" button. 2. S&C displays the profile page. 3. The UN clicks the "Update Profile" button. 4. S&C displays the profile editor. 5. The UN updates the desired fields. 6. The UN clicks the "Save Profile" button. 7. S&C validates the fields. 8. S&C displays the profile page.
Exit Condition	The profile is updated and S&C displays the profile page.
Exceptions	<ul style="list-style-type: none"> • The email is not a valid institutional email address. • The email is already linked to another profile. • The password is shorter than 8 characters. • The passwords do not match. • Another field is invalid. <p>In all cases, S&C displays a descriptive error message.</p>

Table 3.29: Use case 27

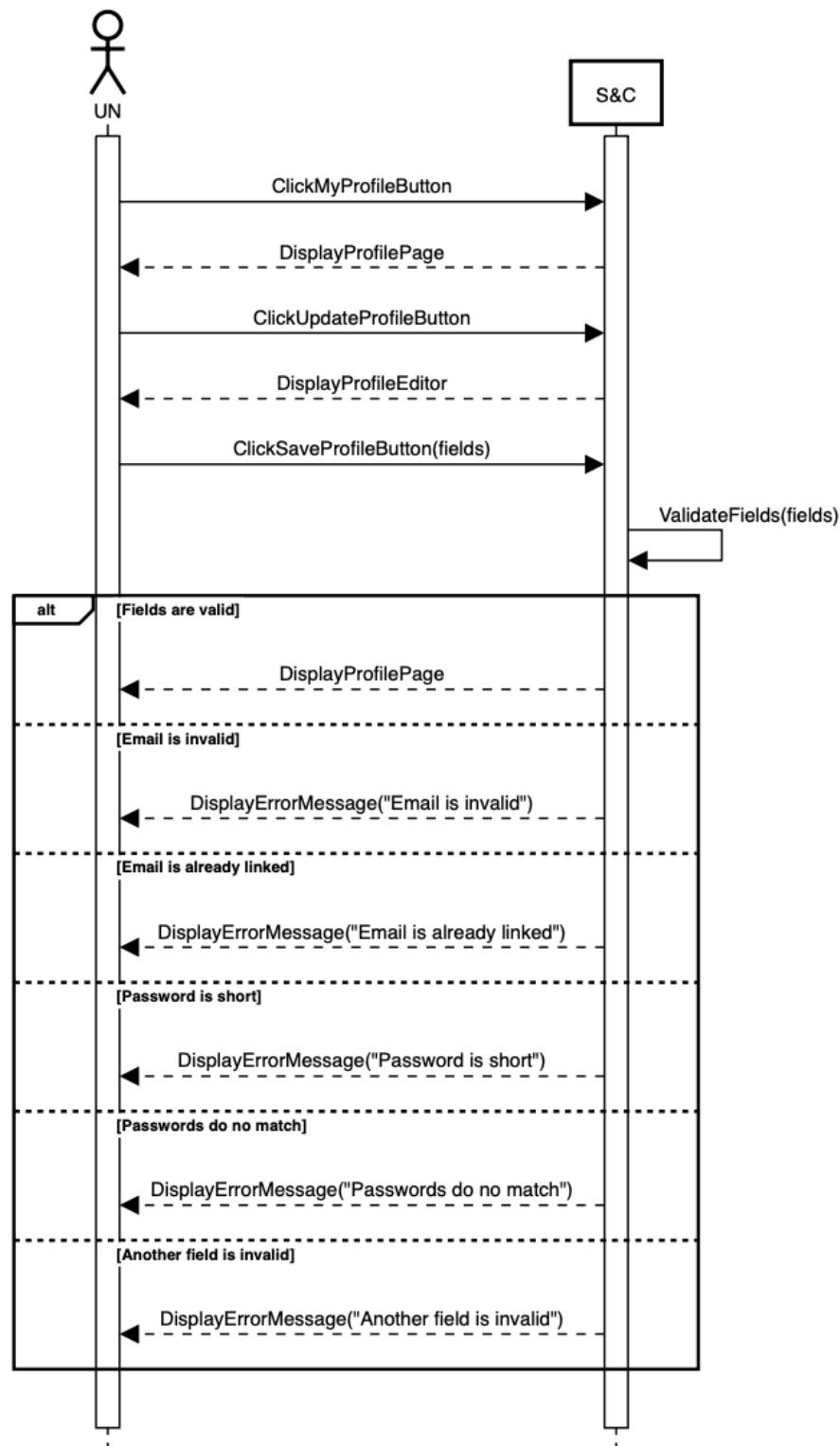


Figure 3.32: UC27 sequence diagram

UC28. University Comments Internship	
Actors	UN, US, CO, EP
Entry Condition	The UN is logged in and S&C displays the home page.
Event Flow	<ol style="list-style-type: none"> 1. The UN clicks the US name. 2. S&C displays the US page. 3. The UN clicks the IN name. 4. S&C displays the IN page. 5. The UN clicks the "Write a Comment" button. 6. S&C displays the comment form. 7. The UN enters the fields. 8. The UN clicks the "Send" button. 9. S&C checks the fields. 10. S&C displays the IN page. 11. S&C sends a notification email to the US via the EP. 12. S&C sends a notification email to the CO via the EP.
Exit Condition	The comment is sent, the US and the CO are notified and S&C displays the IN page.
Exceptions	<ul style="list-style-type: none"> • A field is invalid. <p>In this case, S&C displays a descriptive error message.</p>

Table 3.30: Use case 28

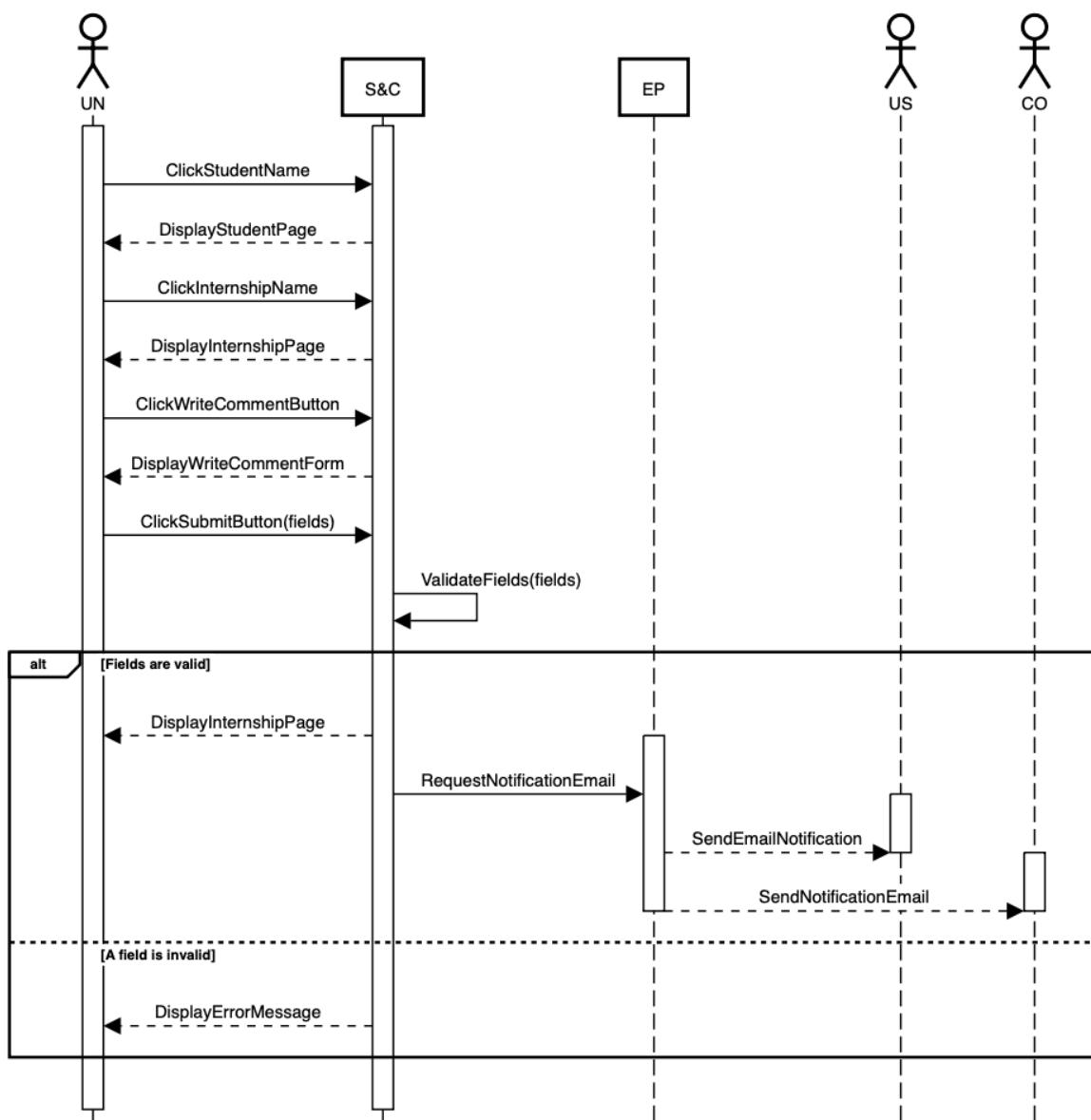


Figure 3.33: UC28 sequence diagram

3.3. Performance Requirements

To ensure an efficient and interactive experience, Students&Companies must meet stringent performance requirements that address scalability, data management and responsiveness under varying loads.

Concurrent Users

The platform is expected to handle a significant user base, as it will cater to multiple universities. To ensure a seamless experience, S&C must support at least 20% of active users simultaneously, which could mean up to 50,000 users at peak times. This guarantees stability and reliability during critical periods, such as application deadlines.

Data Storage

The platform must maintain records of both student profiles and company postings. Historical data such as past internships, feedback scores and selection outcomes should be preserved for analytics and future reference. In general, no data should be deleted without explicit consent, ensuring transparency and trust.

Response Time

Core operations like user authentication, profile updates and search queries should execute within two seconds under normal conditions. During high-load scenarios, response times may increase but must remain reasonable.

Furthermore, the platform's matching algorithm should generate internship recommendations within three seconds, ensuring an enjoyable experience for users and avoiding their disengagement. Feedback mechanisms must also operate swiftly, delivering results to stakeholders within 10 seconds after computation.

3.4. Design Constraints

This section outlines the compliance requirements and the constraints that define the operational boundaries of Students&Companies.

3.4.1. Standard Compliance

Before using S&C, all users must explicitly accept the platform's privacy policy. The platform must comply fully with GDPR regulations, ensuring transparency in how data is collected, stored and processed. This includes providing users with tools to manage their personal data, such as viewing, editing or deleting their information. Secure handling is particularly important for documents like CVs, which may contain highly sensitive personal information.

3.4.2. Hardware Limitations

Users are required to access the platform via devices equipped with stable internet connections and modern web browsers, namely Chrome, Safari and Edge. While desktops provide the best user experience for uploading documents and adding questionnaires, the platform must also ensure functionality on mobile devices. Additionally, the platform must account for varying device capabilities, including lower bandwidth environments, by employing lightweight designs and caching where appropriate.

3.5. Software System Attributes

This section describes the essential software qualities that the Students&Companies platform must maintain to deliver a reliable, secure and interactive experience, capable of supporting its users effectively while accommodating future growth and enhancements.

3.5.1. Reliability

The platform must be fault tolerant, capable of preventing error propagation and ensuring continuous usability. Mechanisms such as database replication, automated failover and regular backups must be in place to maintain reliability. Critical tasks like applications, postings and updates must remain operational even during system failures, with immediate recovery protocols.

3.5.2. Availability

S&C must achieve a minimum availability rate of 99.9%, limiting downtime to approximately 8.76 hours per year. Maintenance should be minimized and scheduled during low-traffic periods, such as nighttime, to prevent disruptions. Special care must be taken to ensure platform availability during peak usage periods, such as application deadlines or high-profile internship postings.

3.5.3. Security

The platform must implement robust access control, ensuring authentication to verify user identities and authorization to confirm permissions for specific actions. All communication must be encrypted using protocols like HTTPS and TLS 1.3 or higher to prevent data breaches. The database must employ measures to defend against vulnerabilities and all user credentials and personal data must be securely stored using encryption techniques. Regular security audits, penetration testing and monitoring systems will ensure the platform's resilience against cyber threats.

3.5.4. Maintainability

S&C must be designed using scalable and reusable models, allowing the addition of new features or improvements with minimal effort. A modular architecture is essential to

enable independent updates to components without impacting the entire system. Maintenance windows must be scheduled during off-peak hours, typically at night, to ensure uninterrupted access during high-traffic periods. Detailed documentation and automated testing must support maintenance and future development.

3.5.5. Portability

The platform must be accessible from all major web browsers, on both desktop and mobile devices. The design must be responsive, ensuring usability across various screen sizes and resolutions. While there are no current requirements, the platform should support infrastructure migrations if necessary, such as moving to cloud-based solutions for enhanced scalability and flexibility.

4 | Formal Analysis

This chapter provides a formal analysis of the system using the Alloy language and a couple of example worlds.

4.1. Model

Below is a code snippet of the model, including signatures, facts and assertions.

```
-- SIGNATURES

sig Email {}

sig Password {}

abstract sig User {
    email: one Email,
    password: one Password
}

sig Name {}

sig Surname {}

sig CV {}

sig Preference {}

sig Student extends User {
    name: one Name,
    surname: one Surname,
    cv: lone CV,
    preferences: set Preference,
    university: one University,
    applications: set Application,
    internships: set Internship,
    feedbacks: set Feedback
}

sig Field {}
```

```

sig Company extends User {
    name: one Name,
    field: one Field,
    positions: set Position,
    feedback: set Feedback
}

sig University extends User {
    name: one Name,
    students: some Student
}

sig Domain {}

sig Project {}

sig Task {}

sig Term {}

sig Position {
    name: one Name,
    domain: one Domain,
    project: some Project,
    tasks: some Task,
    terms: some Term,
    company: one Company,
    acceptedStudents: set Student,
    rejectedStudents: set Student,
    internship: one Internship
}

abstract sig ApplicationMode {}

sig Direct, Recommended extends ApplicationMode {}

abstract sig ApplicationStatus {}

sig ApplicationAccepted, ApplicationRejected extends ApplicationStatus {}

sig Application {
    status: one ApplicationStatus,
    mode: one ApplicationMode,
    position: one Position,
    student: one Student
}

abstract sig Bool {}

```

```
sig True, False extends Bool {}

sig Recommendation {
    studentAccepted: one Bool,
    companyAccepted: one Bool,
    student: one Student,
    position: one Position
}

sig Questionnaire {}

abstract sig SelectionOutcome {}

sig Accepted, Rejected extends SelectionOutcome {}

sig Selection {
    outcome: one SelectionOutcome,
    position: one Position,
    student: one Student,
    interviews: set Interview,
    questionnaires: set Questionnaire
}

abstract sig InterviewMode {}

sig Remote, Onsite extends InterviewMode {}

sig Date {}

sig Interview {
    date: one Date,
    mode: one InterviewMode
}

abstract sig InternshipStatus {}

sig Ongoing, Finished extends InternshipStatus {}

sig Internship {
    status: one InternshipStatus,
    company: one Company,
    student: one Student,
    selection: set Selection,
    comments: set Comment
}

sig Comment {
    internship: one Internship,
```

```

    user: one User
}

sig Feedback {
    user: one User
}

-- FACTS

-- All emails are unique
fact EmailsAreUnique {
    no disj u1, u2: User |
        u1.email = u2.email
}

-- All CVs are unique
fact StudentCVsAreUnique {
    no disj s1, s2: Student |
        s1.cv = s2.cv
}

-- Two distinct universities cannot have the same name
fact UniversityNamesAreUnique {
    no disj u1, u2: University |
        u1.name = u2.name
}

-- Students, companies, and universities cannot share the same name
fact EntityNamesAreDistinct {
    all s: Student, c: Company |
        s.name != c.name
    all s: Student, u: University |
        s.name != u.name
    all c: Company, u: University |
        c.name != u.name
}

-- Two different universities cannot have students in common
fact UniversitiesDoNotShareStudents {
    all u1, u2: University |
        u1 != u2 implies #(u1.students & u2.students) = 0
}

-- An application belongs to only one student
fact ApplicationHasSingleOwner {
    all a: Application |
        lone s: Student |
            a in s.applications
}

```

```

-- For each (student, position) pair, there is at most one application
fact UniqueApplicationPerStudentPosition {
    all s: Student, p: Position |
        lone a: Application |
            a.student = s and a.position = p
}

-- All positions in a student's internship selections appear in the student's
-- applications
fact PositionsInInternshipsAreInApplications {
    all i: Internship |
        all sel: i.selection |
            some a: Application |
                a.student = i.student
                and a.position = sel.position
                and sel.student = a.student
}

-- A student can have at most one ongoing internship
fact SingleOngoingInternship {
    all s: Student |
        lone i: s.internships |
            i.status = Ongoing
}

-- A student's internships must match the internships listing that student
fact StudentsHaveOnlyTheirInternships {
    all s: Student |
        all i: Internship |
            i.student = s iff i in s.internships
}

-- A student's applications must match the applications listing that student
fact StudentApplicationsAreConsistent {
    all s: Student |
        all a: Application |
            a.student = s iff a in s.applications
}

-- If an application for a position is accepted that student must be in the set
-- of the accepted ones for that position
fact AcceptedApplicationsMatchAcceptedSets {
    all a: Application |
        (a.status = ApplicationAccepted) iff (a.student in
            a.position.acceptedStudents)
}

-- If an application for a position is rejected the student must be in the set

```

```

    of the rejected ones for that position
fact RejectedApplicationsMatchRejectedSets {
    all a: Application |
        (a.status = ApplicationRejected) iff (a.student in
            a.position.rejectedStudents)
}

-- If a student is in a positions accepted or rejected set, that student must
-- have applied
fact ConsistentApplicationStatus {
    all s: Student, p: Position |
        (s in p.acceptedStudents or s in p.rejectedStudents)
        iff some a: s.applications | a.position = p
}

-- No position may list the same student in both accepted and rejected sets
fact PositionAcceptedAndRejectedSetsAreDisjoint {
    all p: Position |
        (p.acceptedStudents & p.rejectedStudents) = none
}

-- Positions are unique by at least one differing field
fact UniquePositions {
    all p1, p2: Position |
        p1 != p2 iff (
            p1.name != p2.name or
            p1.domain != p2.domain or
            p1.project != p2.project or
            p1.tasks != p2.tasks or
            p1.company != p2.company or
            p1.internship != p2.internship
        )
}

-- A direct application exists iff there is no matching recommendation
fact DirectApplicationsHaveNoRecommendations {
    all a: Application |
        (a.mode = Direct) iff
            no r: Recommendation |
                r.student = a.student and r.position = a.position
}

-- A recommended application exists iff there is a matching recommendation
fact RecommendedApplicationsHaveRecommendations {
    all a: Application |
        (a.mode = Recommended) iff
            some r: Recommendation |
                r.student = a.student and r.position = a.position
}

```

```

-- Each (student, position) recommendation is unique
fact RecommendationsAreUnique {
    all r1, r2: Recommendation |
        r1 != r2 iff (
            r1.student != r2.student or
            r1.position != r2.position or
            r1.position.company != r2.position.company
        )
}

-- A recommended and accepted application matches a recommendation where both
-- student and company accepted
fact AcceptedRecommendedApplicationsMatchRecommendations {
    all a: Application |
        a.mode = Recommended and a.status = ApplicationAccepted iff
            some r: Recommendation |
                r.student = a.student
                and r.position = a.position
                and r.studentAccepted = True
                and r.companyAccepted = True
}

-- A recommended and rejected application matches a recommendation where
-- student or company (or both) did not accept
fact RejectedRecommendedApplicationsMatchRecommendations {
    all a: Application |
        (a.mode = Recommended and a.status = ApplicationRejected) iff
            some r: Recommendation |
                r.student = a.student
                and r.position = a.position
                and (
                    (r.studentAccepted = False and r.companyAccepted = True) or
                    (r.studentAccepted = False and r.companyAccepted = False) or
                    (r.studentAccepted = True and r.companyAccepted = False)
                )
}

-- An accepted application must have a corresponding selection
fact AcceptedApplicationsRequireSelection {
    all a: Application |
        (a.status = ApplicationAccepted) implies
            one s: Selection |
                s.student = a.student and s.position = a.position
}

-- Accepted applications must have one selection; rejected applications have
-- none
fact AcceptedAppsHaveSelectionsRejectedAppsHaveNone {

```

```

all a: Application |
  (a.status = ApplicationAccepted implies
    one s: Selection |
      s.student = a.student and s.position = a.position
  )
  and
  (a.status = ApplicationRejected implies
    no s: Selection |
      s.student = a.student and s.position = a.position
  )
}

-- Selections exist only for accepted applications
fact SelectionExistsOnlyForAcceptedApplications {
  all s: Selection |
    some a: Application |
      a.student = s.student
      and a.position = s.position
      and a.status = ApplicationAccepted
}

-- For each (student, position) pair, there is at most one selection
fact UniqueSelectionForStudentPosition {
  all s: Student, p: Position |
    lone sel: Selection |
      sel.student = s and sel.position = p
}

-- For each position, at most one selection that ends with a positive outcome
fact SelectionConsistency {
  all p: Position |
    lone s: Selection |
      s.position = p and s.outcome = Accepted
}

-- Each interview and questionnaire belongs to exactly one selection
fact InterviewsAndQuestionnairesHaveUniqueAssociations {
  all i: Interview |
    one s: Selection |
      i in s.interviews
  all q: Questionnaire |
    one s: Selection |
      q in s.questionnaires
}

-- All interviews have different dates
fact InterviewDatesAreUnique {
  all disj i1, i2: Interview |
    i1.date != i2.date
}

```

```

}

-- An internship exists iff there is an accepted selection for the same
student and a position from that company
fact InternshipExistsIfSelectionAccepted {
    all i: Internship |
        (some s: Selection |
            s.position in i.company.positions
            and s.student = i.student
            and s.outcome = Accepted
        ) iff i in i.student.internships
}

-- Each position has exactly one corresponding internship
fact SingleInternshipPerPosition {
    all p1, p2: Position |
        p1 != p2 iff p1.internship != p2.internship
}

-- No two internships are identical in student, company, or selection
fact InternshipsAreUnique {
    all disj i1, i2: Internship |
        i1.student != i2.student
        or i1.company != i2.company
        or i1.selection != i2.selection
}

-- No two internships can involve the same student for the same position
fact UniqueInternshipPerStudentAndPosition {
    all i1, i2: Internship |
        i1 != i2 implies (
            i1.student != i2.student or
            no p: Position |
                (some s1: i1.selection | s1.position = p)
                and (some s2: i2.selection | s2.position = p)
        )
}

-- A university lists a student only if that student actually belongs to it
and has an ongoing internship
fact StudentMonitorsCondition {
    all u: University, s: Student |
        s in u.students iff
            (s.university = u and some i: Internship | i.student = s)
}

-- Each comment relates to exactly one internship
fact CommentsRelateToAtMostOneInternship {
    all c: Comment |

```

```

lone i: Internship |
  c.internship = i
}

-- Students can only comment on their own internships
fact StudentsCanOnlyCommentOnTheirOwnInternships {
  all c: Comment |
    c.user in Student implies
      c.internship in c.user.internships
}

-- Companies can only comment on internships they provide
fact CompanyCommentsOnProvidedInternships {
  all c: Comment |
    c.user in Company implies
      c.internship.company = c.user
}

-- Universities can only comment on internships of their students
fact UniversityCommentsOnMonitoredInternships {
  all u: University, c: Comment |
    c.user = u implies
      c.internship.student in u.students
}

-- Internships contain all and only the comments related to them
fact InternshipsContainOnlyTheirComments {
  all i: Internship |
    i.comments = {c: Comment | c.internship = i}
}

-- The author of a feedback is either a student or a company
fact FeedbackAuthor {
  all f: Feedback |
    f.user in Student + Company
}

-- Companies list all and only their feedback
fact CompanyHaveOnlyTheirFeedback {
  all co: Company |
    co.feedback = {f: Feedback | f.user = co}
}

-- Students list all and only their feedback
fact StudentHaveOnlyTheirFeedback {
  all s: Student |
    s.feedbacks = {f: Feedback | f.user = s}
}

```

```
-- ASSERTIONS --

assert NoDuplicateEmails {
    all u1, u2: User |
        u1 != u2 implies u1.email != u2.email
}
check NoDuplicateEmails for 5

assert NoDuplicateCVs {
    all s1, s2: Student |
        s1 != s2 implies s1.cv != s2.cv
}
check NoDuplicateCVs for 5

assert NoDuplicateRecommendations {
    all r1, r2: Recommendation |
        r1 != r2 implies (
            r1.student != r2.student or
            r1.position != r2.position or
            r1.position.company != r2.position.company
        )
}
check NoDuplicateRecommendations for 5

assert NoDuplicateApplications {
    all s: Student |
        all disj a1, a2: s.applications |
            a1.position != a2.position
}
check NoDuplicateApplications for 5

assert SelectionsMatchApplications {
    all sel: Selection |
        some a: sel.student.applications |
            a.position = sel.position
}
check SelectionsMatchApplications for 5

assert NoDuplicatePositionsOutcome {
    all p: Position |
        no s: Student |
            s in p.acceptedStudents and s in p.rejectedStudents
}
check NoDuplicatePositionsOutcome for 5

assert ApplicationsAcceptedMatchPositionsOutcome {
    all a: Application |
        (a.status = ApplicationAccepted) implies
```

```

        a.student in a.position.acceptedStudents
}
check ApplicationsAcceptedMatchPositionsOutcome for 5

assert ApplicationsRejectedMatchPositionsOutcome {
    all a: Application |
        (a.status = ApplicationRejected) implies
            a.student in a.position.rejectedStudents
}
check ApplicationsRejectedMatchPositionsOutcome for 5

assert InternshipsMatchApplications {
    all i: Internship |
        all sel: i.selection |
            some a: Application |
                a.student = i.student
                and a.position = sel.position
                and a.student = sel.student
}
check InternshipsMatchApplications for 5

assert UniqueOngoingInternships {
    all s: Student |
        no disj i1, i2: s.internships |
            i1.status = Ongoing and i2.status = Ongoing
}
check UniqueOngoingInternships for 5

--PRED

--Base world: 2 students, 1 company, 1 position. Both students apply, one
--through recommendation. Only one students obtains the internship.
pred BaseWorld {
    #Company =1
    #Student= 2
    #Position = 1
    #Application =2
    #Selection =1
    #Recommendation=1
    #Internship=1
}

run BaseWorld for 4

--Second world: 2 students, 1 company, 1 position. Both students apply, one
--through recommendation. They are both accepted for the selection process,
--but only one obtains the internship.
pred SecondWorld {

```

```
#Company =1  
#Student= 2  
#Position = 1  
#Application =2  
#Selection =2  
#Recommendation=1  
#Internship=1  
}  
  
run SecondWorld for 4
```

4.1.1. Example Worlds

Below are several examples of the model, demonstrating its coherence and correctness with respect to its key functionalities.

4.1.2. Base World

The following base world sets the stage by outlining the domain: two students apply for the same position, one through a recommendation and the other through a direct application. However, only one progresses to the selection process.

Second World

In this world, two students apply for the same position. Both successfully advance to the selection process, but only one ultimately secures the internship.

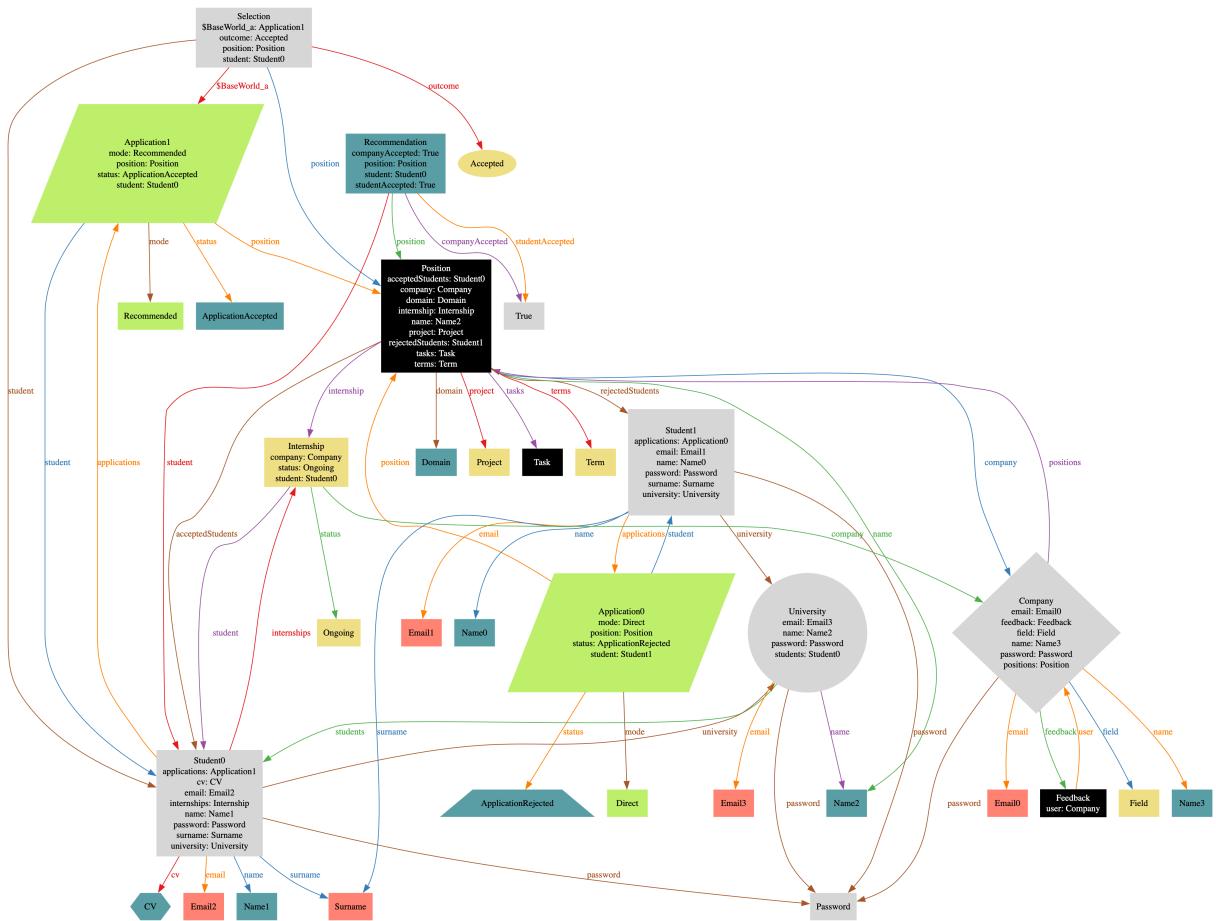


Figure 4.1: Base world

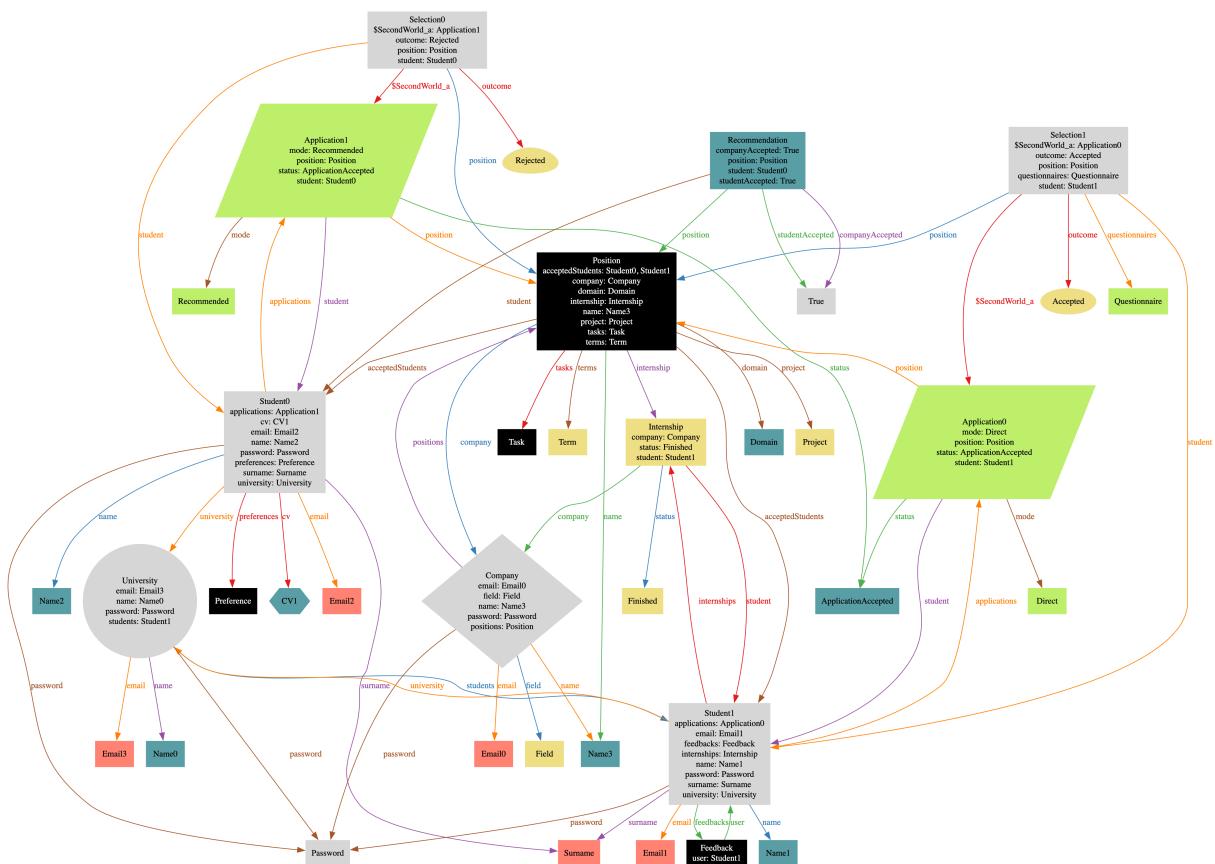


Figure 4.2: Second world

5

Workload

This chapter quantifies the workload of the authors in writing the document.

Andrea Carrara

Task	Hours
Purpose	3
Scope	4
Class Diagram	3
Product Functions	4
Assumptions and Constraints	1
External Interface Requirements	3
Requirements	3
Sequence Diagrams	16
Performance Requirements	1
Design Constraints	1
Software System Attributes	1
Peer Review	20
Total	60

Table 5.1: Workload of Andrea Carrara

Federica Curò Dossi

Task	Hours
Purpose	3
Scope	4

Scenarios	4
State Diagrams	2
User Characteristics	1
Use Cases	14
Formal Analysis	12
Peer Review	20
Total	60

Table 5.2: Workload of Federica Currò Dossi

Bibliography

- [1] IEEE. *IEEE 29148:2018: Systems and Software Engineering, Life Cycle Processes and Requirements Engineering*. IEEE, 2018.
- [2] Micheal Jackson. *The World and the Machine*. IEEE, 1995.
- [3] Elisabetta Di Nitto. *Requirements Engineering and Design Project*. Software Engineering 2, 2024.

List of Figures

2.1	Class diagram	12
2.2	Recommendations state diagram	13
2.3	Selection process state diagram	13
3.1	Wireframe	18
3.2	User use cases diagram	25
3.3	UC1 sequence diagram	27
3.4	UC2 sequence diagram	28
3.5	UC3 sequence diagram	29
3.6	Student use cases diagram	30
3.7	UC4 sequence diagram	32
3.8	UC5 sequence diagram	34
3.9	UC6 sequence diagram	35
3.10	UC7 sequence diagram	36
3.11	UC8 sequence diagram	37
3.12	UC9 sequence diagram	39
3.13	UC10 sequence diagram	40
3.14	UC11 sequence diagram	42
3.15	UC12 sequence diagram	44
3.16	UC13 sequence diagram	46
3.17	UC14 sequence diagram	48
3.18	UC15 sequence diagram	50
3.19	Company use cases diagram	51
3.20	UC16 sequence diagram	53
3.21	UC17 sequence diagram	55
3.22	UC18 sequence diagram	57
3.23	UC19 sequence diagram	58
3.24	UC20 sequence diagram	59
3.25	UC21 sequence diagram	61
3.26	UC22 sequence diagram	63
3.27	UC23 sequence diagram	65
3.28	UC24 sequence diagram	67
3.29	UC25 sequence diagram	69
3.30	University use cases diagram	70
3.31	UC26 sequence diagram	71
3.32	UC27 sequence diagram	73
3.33	UC28 sequence diagram	75

4.1	Base world	92
4.2	Second world	93

List of Tables

1.1	Goals	2
1.2	World phenomena	2
1.3	Shared phenomena controlled by the world and observed by the machine . .	5
1.4	Shared phenomena controlled by the machine and observed by the world . .	6
1.5	Glossary	7
2.1	Domain assumptions	15
2.2	Constraints	16
3.1	Requirements	22
3.2	Traceability matrix	25
3.3	Use case 1	26
3.4	Use case 2	28
3.5	Use case 3	29
3.6	Use case 4	31
3.7	Use case 5	33
3.8	Use case 6	35
3.9	Use case 7	36
3.10	Use case 8	37
3.11	Use case 9	38
3.12	Use case 10	40
3.13	Use case 11	41
3.14	Use case 12	43
3.15	Use case 13	45
3.16	Use case 14	47
3.17	Use case 15	49
3.18	Use case 16	52
3.19	Use case 17	54
3.20	Use case 18	56
3.21	Use case 19	58
3.22	Use case 20	59
3.23	Use case 21	60
3.24	Use case 22	62
3.25	Use case 23	64
3.26	Use case 24	66
3.27	Use case 25	68
3.28	Use case 26	70

3.29 Use case 27	72
3.30 Use case 28	74
5.1 Workload of Andrea Carrara	95
5.2 Workload of Federica Currò Dossi	96