



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Software Engineering 2

Requirements Analysis and

Specification Document

Author(s): **Riccardo Piantoni - 10816545**

Matteo Rossi - 10798975

Jacopo Sacramone - 10752157

Academic Year: 2024-2025

Version: 1.0

Release date: 22/12/2024

Contents

Contents	i
-----------------	----------

1 Introduction	1
1.1 Purpose	1
1.1.1 Goals	1
1.2 Scope	2
1.2.1 World Phenomena	2
1.2.2 Shared Phenomena	3
1.3 Definitions, Acronyms, Abbreviations	5
1.4 Revision history	6
1.5 Reference Documents	6
1.6 Document Structure	7
2 Overall Description	8
2.1 Product Perspective	8
2.1.1 Scenarios	8
2.1.2 Domain-level Class Diagram	12
2.1.3 State Diagrams	14
2.2 Product Functions	15
2.2.1 Requirements	16
2.3 User Characteristics	19
2.4 Assumptions, Dependencies and Constraints	20
2.4.1 Domain Assumptions	20
3 Specific Requirements	22
3.1 External Interface Requirements	22
3.1.1 User Interfaces	22
3.1.2 Software Interfaces	25
3.1.3 Communication Interfaces	25
3.1.4 Hardware Interfaces	26
3.2 Functional Requirements	27
3.2.1 Use Cases and Activity Diagrams	28
3.2.2 Mapping On Requirements	68
3.3 Performance Requirements	68
3.4 Design Constraints	69

3.4.1	Standards Compliance	69
3.4.2	Hardware Limitations	70
3.5	Software System Attributes	70
3.5.1	Reliability	70
3.5.2	Availability	71
3.5.3	Security	71
3.5.4	Maintainability	71
3.5.5	Portability	72
4	Formal Analysis using Alloy	73
4.1	Objectives	73
4.2	The Model	74
4.2.1	Signatures	74
4.2.2	Actions	76
4.2.3	Facts	79
4.2.4	Main	81
4.3	Validation through the Model	84
4.3.1	Instance of the Large Static World	84
4.3.2	Instance of the Example Dynamic World	85
5	Effort Spent	87
6	References	88
6.1	Used Tools	88
List of Figures		89

1 | Introduction

1

1.1. Purpose

The transition from academia to the job market often presents significant challenges for university students, as they face difficulties in aligning their academic skills with industry expectations. Companies, on the other hand, struggle to efficiently connect with young talent and promote internships and job opportunities tailored to their needs. These gaps create inefficiencies in the hiring process, leaving valuable opportunities untapped.

Students&Companies (S&C) is designed to fill this gap. The platform aims to facilitate entrance into the job market for students while enabling companies to effectively reach and recruit emerging talent. By addressing the mismatch between academic preparation and industry requirements, S&C enhances the matching process, creating an ecosystem where education meets practical experience.

1.1.1. Goals

In this section, there are defined the goals that the system has to achieve:

Profile Management

[G1] Students, companies and universities should manage their profiles on the platform.

Internships Publication and Search

[G2] Companies should advertise their internships.

[G3] Students should apply for internships.

Recommendations

[G4] Students should receive recommendations about internship offers matching their CVs.

[G5] Companies should receive recommendations about students matching their preferences.

Selection Process

[G6] Companies should manage the selection process aimed at evaluating candidates for their internship offers.

Monitoring

[G7] Both students and companies should monitor the evolution of the internships they are taking part in.

[G8] Both students and companies should report complaints about the ongoing internships.

[G9] Universities should handle complaints about the internships of their students.

[G10] Both students and companies should provide feedback regarding the concluded internship they have taken part in.

Submission Suggestions

[G11] Students should receive suggestions about how to make their CVs more appealing.

[G12] Companies should receive suggestions about how to make their internship offers' descriptions more appealing.

1.2. Scope

Students&Companies (S&C) is a platform that acts as an intermediary system facilitating the internship matching process between students and companies. It allows companies to advertise internships and students to search, receive customized recommendations, and initiate contact.

The platform defines **Recommendation** as the automated process of identifying suitable internship opportunities for students and potential candidates for companies. Following this, a **Contact** represents the phase in which students and companies communicate via the platform to conduct the selection process, including interviews and candidate selection.

The system automates key activities such as generating recommendations using various mechanisms, coordinating interviews, and collecting feedback to improve its algorithms. Additionally, it provides tools to monitor the progress of contacts and internships, manage issues through communication features, and enable universities to supervise the status of internships, ensuring compliance and resolving possible complaints effectively.

1.2.1. World Phenomena

[WP1] Students redact their CV.

[WP2] Companies make new internship positions available.

[WP3] Students decide to take an internship.

[WP4] Companies select candidates to be interviewed among the applicants for each of their open positions.

[WP5] Companies select the student candidates who fit the most according to the results of

their interviews.

[WP6] Students carry on their internships at their companies.

[WP7] A problem arises in an ongoing internship.

[WP8] Universities handle complaints (eventually interrupting internships).

1.2.2. Shared Phenomena

World Controlled

[SP1] Students, companies and universities sign up to the platform.

[SP2] Students, companies and universities log into the platform.

[SP3] Users provide information about themselves to the system.

[SP4] Companies provide information about their internship positions to the system.

[SP5] Students submit filters to search for suitable internship positions.

[SP6] Students proactively apply for an open internship position.

[SP7] Students track received recommendations about internship offers.

[SP8] Companies track recommendations about potential candidates for their internship offers.

[SP9] Students accept recommendations for internships.

[SP10] Students reject recommendations for internships.

[SP11] Companies accept recommended candidates for their internships.

[SP12] Companies reject recommended candidates for their internships.

[SP13] Companies contact selected students to set up an interview with them.

[SP14] Companies submit questions to students.

[SP15] Students provide the information required by companies during the interview.

[SP16] Companies finalize the selection process.

[SP17] Students track the outcomes of the interviews they have participated in.

[SP18] Students and companies report information about an ongoing internship.

[SP19] Students, companies and universities monitor an ongoing internship.

[SP20] Students and companies report complaints and problems about ongoing internships.

[SP21] Universities report information about the problems they have handled.

[SP22] Students and companies provide feedback and suggestions about internships.

[SP23] Students ask for suggestions about their profiles.

[SP24] Companies ask for suggestions about their internship offers.

Machine Controlled

[SP25] The system shows to the students the internship offers which match their selection criteria.

[SP26] The system shows to the companies the list of all the students who have applied for their internship offers.

[SP27] The system shows to students and companies the list of all their received recommendations along with their status.

[SP28] The system forwards communications about the scheduling of interviews from companies to students (and vice-versa).

[SP29] The system forwards the submitted questions from companies to students.

[SP30] The system forwards the submitted answers from students to companies, which collect them.

[SP31] The system shows to the candidate students the outcome of their interviews after the conclusion of a selection process.

[SP32] The system forwards information about the ongoing internships to students and companies.

[SP33] The system forwards to universities information about problems in the ongoing internships of their students.

[SP34] The system asks for feedback after the conclusion of an internship to improve its recommendation algorithms.

[SP35] The system provides suggestions to students about their profiles.

[SP36] The system provides suggestions to companies about the description of their internship offers.

1.3. Definitions, Acronyms, Abbreviations

- **System, Platform:** these terms are used interchangeably when referring to the system-to-be-developed.
- **Upload a CV:** refers to completing all required fields in the CV section of the user's profile. This isn't an upload of a file to enforce a standardized format and enable the system to efficiently collect and process the information.
- **University, Company:** when the terms are referenced as performing an action, it means that the action is executed by a representative acting on behalf of the respective entity.
- **Party:** the term refers to the entities actively involved in the process of applying, participating and managing internships, so both Student and Company; it doesn't include the University.
- **Notification:** indicates a message appearing in the target page of the recipient, not an actual notification shared via email or received on a device.
- **Platform Guidelines:** a set of rules and policies ensuring standardized behavior across users and maintaining the platform's integrity and usability.
- **In-Platform vs. In-Person Interviews:**
 - **In-Platform Interview:** Conducted entirely through the platform tools, such as structured questionnaires.
 - **In-Person Interviews:** Requires the candidate and interviewer to meet physically at a designated location.
- **Accuracy:** represents the proportion of correct recommendations made by the system, calculated as the ratio of successful matches to the total number of recommendations generated. It provides an overall measure of how well the system performs.
- **F1:** a performance metric that combines precision and recall into a single value, balancing the trade-off between the two. Precision measures the proportion of correct recommendations among all generated recommendations, while recall measures the proportion of relevant matches identified out of all possible relevant matches. The F1 score is particularly useful in evaluating the recommendation system when both false positives and false negatives are significant concerns.

Acronyms	Definition
RASD	Requirements Analysis & Specification Document
API	Application Programming Interface
HTTPS	HyperText Transfer Protocol over Secure Socket Layer
2FA	Two-Factor Authentication

Table 1.1: Acronyms used in the document.

Abbreviations	Definition
S&C	Student&Company
G	Goal
WP	World Phenomena
SP	Shared Phenomena
DA	Domain Assumption
R	Requirement
UC	Use Case

Table 1.2: Abbreviations used in the document.

1.4. Revision history

Revised on	Version	Description
22/12/2024	1.0	Initial Release of the document

Table 1.3: Revision history

1.5. Reference Documents

- [1] Di Nitto, Rossi, Camilli, "*A.Y. 2024-2025 Software Engineering 2 Requirement Engineering and Design Project*", 2024.
- [2] ISO/IEC/IEEE 29148:2018, "*Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*," International Organization for Standardization, International Electrotechnical Commission, and Institute of Electrical and Electronics Engineers, 2018.

1.6. Document Structure

This document is composed of six sections:

- 1st Chapter - Introduction
- 2nd Chapter - Overall Description
- 3rd Chapter - Specific Requirements
- 4th Chapter - Formal Analysis using Alloy
- 5th Chapter - Effort Spent
- 6th Chapter - References

2 | Overall Description

8

2.1. Product Perspective

2.1.1. Scenarios

- **1st Scenario: Sign Up and Profile Creation Performed by a Student**

Marco Stella, a student at Politecnico di Milano, wants to find a suitable summer internship to put into practice the knowledge gained during his Bachelor's Degree in Computer Science. He has discovered the opportunity offered by the S&C platform and has decided to sign up in order to browse the advertised internships. He is required to provide the necessary information in order to subscribe, such as his educational email, password, and other personal details. Once the account verification process has been completed, Marco is asked to upload his CV and potentially add other relevant details about his background, skills and attitudes. Once this information is inserted into the platform, all the functionalities provided by S&C get enabled and he can freely navigate his personal dashboard, finally able to start the search process.

- **2nd Scenario: Sign Up and Profile Creation Performed by a Company**

BrightFuture, a company specializing in AI solutions, is looking to recruit talented interns to support its upcoming projects. The company has heard about the opportunities provided by the S&C platform and decides to register in order to post internship offers and connect with potential candidates. To begin, a representative from BrightFuture Technologies signs up by providing the necessary company details, such as the corporate email address, password, and additional contact information about the organization. Once the account registration is complete, the system sends a confirmation email for verification. After confirming the email, the representative is prompted to create a company profile: this involves adding essential details, such as a description of the organization, its mission, and the fields it operates in. They can also upload the company logo and other branding elements to enhance the profile's appeal to prospective interns. With the profile fully set up, BrightFuture Technologies can now access all the features of the S&C platform, including its personalized dashboard.

- **3rd Scenario: Publication of an Internship Offer by a Company**

TechFuture, a company specializing in the development of innovative software solutions, decides to use the S&C platform to post an internship opportunity for students interested in the tech industry. After logging into their corporate account using their credentials, a representative selects the "Create New Offer" option from the main menu. The platform prompts the company to provide all the necessary details for the internship: the applica-

tion domain, the tasks to be performed, the required skills, the duration of the internship, the application deadline and the compensation terms. The representative carefully completes all the required fields, reviews the information for accuracy, and submits the offer. The system performs an automatic preliminary verification of the provided details to ensure the offer complies with the platform's policies and standards. Once the review is completed, the internship offer is published on the platform, making it accessible to students who can now apply and explore the opportunity further.

- **4th Scenario: Internship Search by a Student**

Davide Bianchi, a second-year Mechanical Engineering student at Politecnico di Milano, is eager to find an internship that aligns with his academic background and career aspirations. He logs into the S&C platform using his credentials and navigates to the "Search Offers" section. Here, the platform presents him with a search interface, allowing him to apply filters to narrow down the opportunities based on his preferences. Davide specifies his criteria: internships related to mechanical design, located in Italy, with a duration of at least three months. Confident in his choices, he submits the search. Within moments, the system displays a list of internship offers that match Davide's selected criteria. Browsing through the options, he identifies a position posted by Innovex Solutions, a company known for its innovative mechanical systems. The platform confirms that the student has been successfully added to the offer's applicants list. Encouraged by this seamless process, he continues exploring other opportunities to further expand his options.

- **5th Scenario: Automatic Internship Recommendation**

Luca Rossi, a second-year Electrical Engineering student at Politecnico di Milano, has recently completed his profile on the S&C platform. His profile includes details about his academic background, skills, and interests, which he hopes will help him find an ideal internship opportunity. As part of its functionality, the S&C platform automatically analyzes Luca's profile. It leverages the information he has provided - such as his field of study, technical skills, and career aspirations - to compare it with the internship offers available on the platform. Using a recommendation algorithm, the system identifies the internships that best align with Luca's qualifications and professional goals. Once the analysis is complete, the platform compiles a list of recommended internships. Luca receives a notification informing him about the new recommendations. He navigates to the "Recommendation" section, where he can review the suggested opportunities and start applying for positions that match his interests and attitudes.

- **6th Scenario: Automatic Student Recommendation**

GreenSpark Energy, a company specializing in renewable energy solutions, has recently posted an internship offer on the S&C platform. The offer outlines the required skills,

academic background, and other criteria for the ideal candidate. Once the offer is published, the platform's recommendation system begins its analysis. The system retrieves the details of the internship, such as the field of study, necessary qualifications, required skills, and desired attributes. Using these parameters, it compares the offer against the profiles of students registered on the platform. Leveraging its recommendation algorithm, the system identifies a list of students whose profiles closely align with the internship requirements. Once the analysis is complete, the platform compiles a list of recommended candidates. The system notifies GreenSpark Energy and provides access to the list of students, allowing the company to review their profiles and directly invite them to apply for the internship.

- **7th Scenario: Managing an Interview between a Company and a Student**

Innovex Solutions, a company specializing in innovative mechanical systems, is reviewing applications for a recently posted internship position. Among the candidates, they identify Davide Bianchi, a promising Mechanical Engineering student at Politecnico di Milano, whose profile aligns closely with their requirements. The company decides to invite him for an interview. Through the S&C platform, Innovex Solutions sends an interview invitation to Davide, including the date, time, and format (video call or in-person). Shortly after, Davide receives a notification about the interview and logs into the platform to review the details. Finding the schedule convenient, he accepts the invitation and confirms his availability. On the agreed date, the interview is conducted as planned. During the session, Davide discusses his qualifications, skills, and aspirations, while the company representative provides more information about the role and evaluates his suitability. After the interview, Innovex Solutions uses the platform to post the results of the interview, informing Davide whether he has progressed to the next stage of the selection process or been offered the internship. The system updates Davide's status in the selection process, ensuring he remains informed about the outcome.

- **8th Scenario: Reporting and Handling Problems during an Internship**

Giulia Moretti, a Civil Engineering student at the Politecnico di Milano, is excited to begin her internship at Skyline, a company specializing in urban infrastructure projects. On her first day, Giulia is welcomed by her supervisor, who outlines her responsibilities and assigns her to work on a sustainability-focused bridge design project. The initial weeks of her internship are productive, with Giulia learning new skills and gaining hands-on experience. Throughout the internship, Giulia and her supervisor at Skyline regularly update the S&C platform with progress details. Giulia logs her completed tasks, acquired skills, and challenges faced. These updates help both the university and Skyline monitor the alignment between Giulia's work and her learning objectives. However, midway through

the internship, Giulia encounters a significant issue: the tools and resources promised by Skyline for her project, including access to specific engineering software, are unavailable. This lack of resources makes it difficult for Giulia to complete her assigned tasks effectively. After attempting to resolve the issue internally with her supervisor without success, Giulia decides to report the problem to her university. Using the S&C platform, Giulia navigates to the "Report Problems" section and submits a detailed description of the issue. She explains the nature of the problem, its impact on her project, and the steps she has already taken to address it. The platform immediately makes Giulia's report available to her university's internship coordinator. Upon receiving the report, the university's coordinator reviews the details and contacts both Giulia and Skyline to discuss the issue. During a scheduled meeting, the coordinator gathers additional information from both parties and assesses the situation. After evaluating the options, the university decides to address the issue by coordinating with Skyline to speed up the software licensing process. The problem is resolved within a week, and Giulia is able to continue her internship with minimal disruptions. As the internship progresses, Giulia continues to log her tasks and achievements on the S&C platform. Finally, the internship reaches its scheduled conclusion. Giulia submits final feedback summarizing her work, the skills she has developed, and her overall experience. Skyline also provides formal feedback through the platform, highlighting Giulia's strengths and suggesting areas for further improvement. Both the student's and the company's feedback is stored in the system and fed to the recommendation mechanisms, allowing the evaluation of the quality of internships offered by Skyline and the improvement of future recommendations.

- **9th Scenario: Profile Optimization Suggestions for Students**

Alessandro Romano, a second-year Computer Science student at Politecnico di Milano, logs into the S&C platform to ensure his profile is fully optimized for attracting internship opportunities. Navigating to his profile page, he notices a notification prompting him to review suggestions for improvement. The system automatically analyzes Alessandro's profile, considering his academic background, skills, and documents such as his CV. Based on the analysis, the platform generates a list of personalized suggestions. These include adding programming languages that are in high demand, updating his project portfolio, and providing a more detailed description of his certifications. Alessandro reviews the suggestions carefully and starts implementing the changes directly through the platform.

- **10th Scenario: Internship Offers Optimization Suggestions for Companies**

BlueHorizon Robotics, a company specializing in autonomous systems and AI integration, logs into the S&C platform to ensure its internship offers are effectively attracting top candidates for their internship programs. A representative navigates to the page of one

of their offers and notices a prompt indicating available optimization suggestions. The platform's system analyzes the offer and the company's profile details, including details about past internship postings, descriptions of ongoing projects, and the clarity of their technical requirements. Based on the analysis, the platform generates actionable suggestions, such as refining the job description to better outline the scope of responsibilities or specifying advanced technical skills required for certain roles. The representative carefully reviews the suggestions and begins updating the offer's description to align with the recommendations.

2.1.2. Domain-level Class Diagram

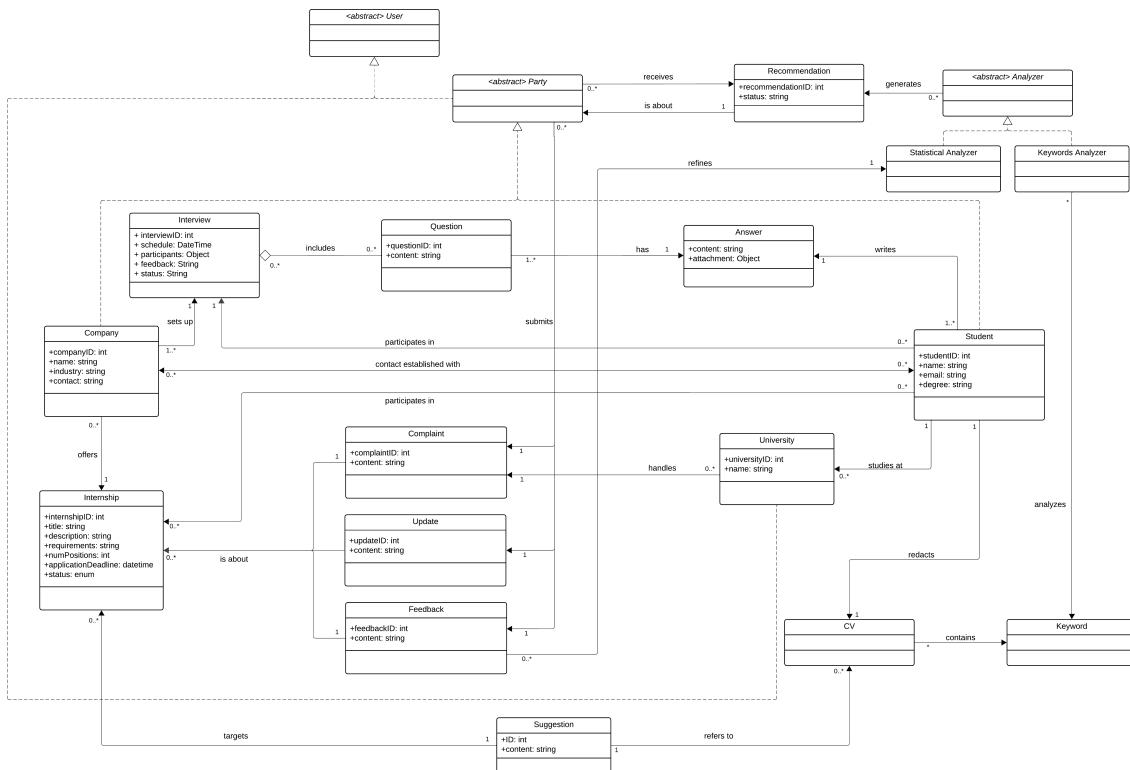


Figure 2.1: Domain-Level Class Diagram.

The entities interacting with the system are modeled through a hierarchical structure to aggregate common functionalities. At the top level, the **User** class generalizes all the users of the system. From it, the **University** and **Party** classes derive, with the latter further specialized into the **Student** and **Company** entities.

The core of the system revolves around the concept of connecting students with internship opportunities: this is reflected in the relationships between the **Student**, **Company**, and **In-**

ternship entities. The Student entity encapsulates candidate data such as personal details and educational background, which combined with the CV class, are essential to matching students with adequate internship opportunities. Companies can publish detailed internship offers that students can eventually search and apply for.

The Recommendation entity serves as a link between CVs and Internships, enabling the system to suggest the most relevant opportunities to students and identify suitable candidates for companies. This process is asymmetric, as recommendations can originate from either party since there's no full intersection in the sets of provided information in CVs and internship offers. For instance, a company might specify a requirement for expertise in a particular software or framework, which may not be explicitly detailed in a student's CV. Despite such mismatches, the system is designed to ensure potential matches are not overlooked. Any missing or unclear details can subsequently be addressed during interviews, which is the proper phase to address further clarification.

The system can generate multiple recommendations for various students for a single internship, and vice-versa, a single CV may be suitable for different internship offers. The recommendation process is powered by specialized Analyzers, which enhance accuracy through techniques such as keyword matching and statistical analysis.

Furthermore, a student remains eligible for new recommendations even while actively enrolled in an internship, since he could be interested in setting up further internships later in time.

When a recommendation is accepted, the system facilitates the next step based on the origin of the recommendation. If the recommendation has originally been sent to a company, the student receives a dual recommendation, inviting him to apply for the given internship, if interested. Otherwise, if the recommendation has been generated for the student, the company receives an application for their internship, similarly to applications spontaneously sent by students after searching. When an internship application deadline is met and both parties have accepted a recommendation, a contact is established and the internship transitions into the interview phase.

The interview process is modeled through the Interview class, which provides a structured framework for evaluating candidates. Each Interview, which is a different entity for every candidate student, is an aggregation of multiple Questions, which are designed to be reusable across different interviews. This approach promotes modularity, allowing companies to build evaluation processes reusable in multiple interviews. The Answer class captures the responses provided by candidates during the interviews. Interviews can be performed in-platform or in-person and, eventually, details are reported in the system.

A key feature of the platform is the facilitation of feedback and communication. During an ongoing internship, Parties can provide Updates, visible only to them, to facilitate communication and share relevant information. At the conclusion of an internship, the Feedback class allows students and companies to provide insights about their experiences, which the system can use to refine future recommendations through the statistical analyzer. The Complaint class offers a mechanism for reporting and addressing issues that may arise during internships. This process is monitored by the University entity, which is the only user able to read them other than the issuer, ensuring that internships comply with established agreements and resolving disputes when necessary.

2.1.3. State Diagrams

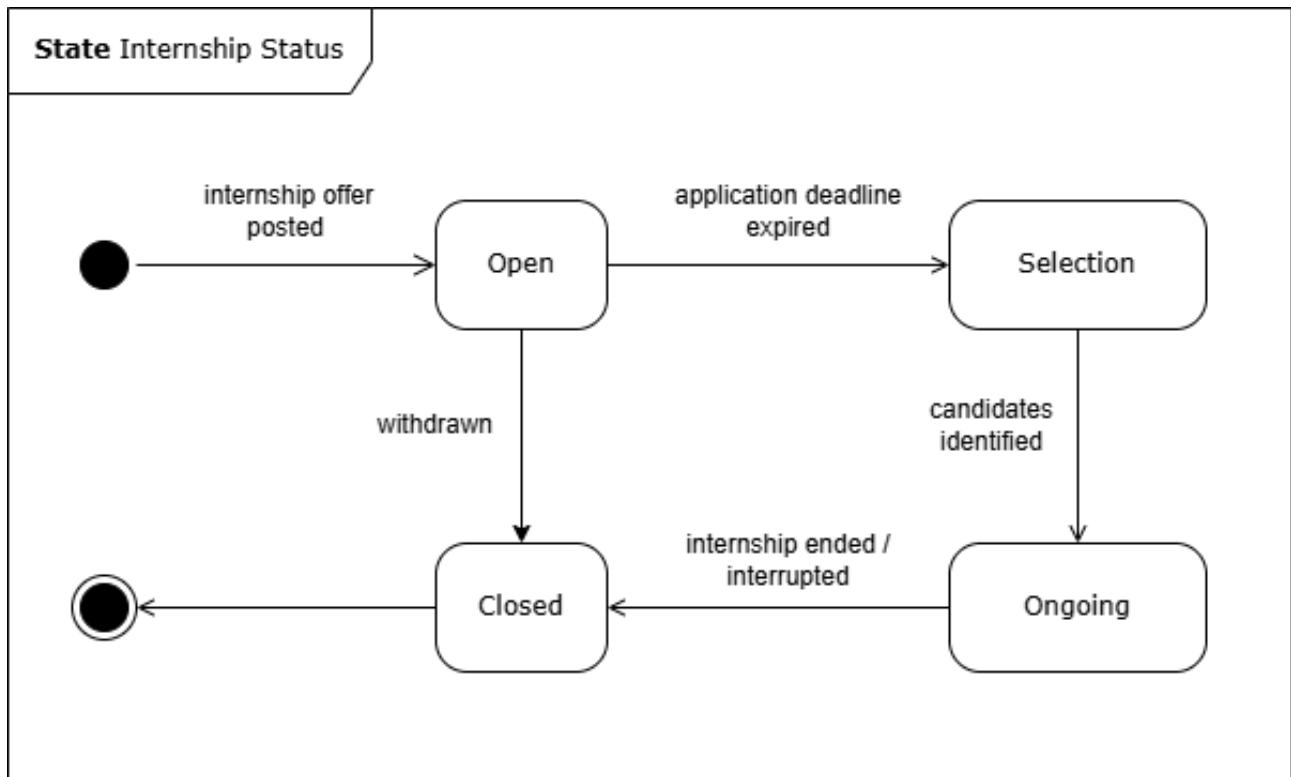


Figure 2.2: Internship Status State Diagram.

This state diagram illustrates the lifecycle of an internship offer on the platform. It begins when an offer is posted, transitioning into the **Open** state, where the internship is available for applications. The offer remains open until either the application deadline expires, leading to the **Selection** phase, or the company withdraws it, moving directly to **Closed**. In the **Selection** state, candidates are evaluated, and if suitable candidates are identified, the process progresses to the **Ongoing** state, where the internship is actively taking place. Finally, upon completion

or interruption of the internship, the offer transitions to the Closed state and it remains stored in the system.

2.2. Product Functions

The S&C platform is designed to streamline the matching, selection and execution of internships, by providing functionalities that support every stage of the process. Key features include internship publication and search, automated recommendations, and monitoring tools. By integrating communication functionalities and feedback mechanisms, the platform ensures an effective experience for all users.

- **Sign Up and Profile Creation**

Users, identified as Students, Companies, and Universities, must sign up to the platform in order to access its functionalities. During the registration process, they create profiles by providing required personal and professional details. For students, details include additional information such as experiences, skills, and attitudes, enabling the platform to generate accurate recommendations. For companies, it also includes the fields in which they operate and eventually their branding elements.

- **Internship Publication**

The platform enables companies to publish detailed and descriptive internship opportunities. Relevant information includes the application domain of the project along with the tasks that the student will be required to perform with the adopted technologies, if applicable. Furthermore, the details include the terms offered by companies: paid internship, training period or other benefits. These details are essential for allowing students to make informed decisions and to improve the recommendation algorithm. Companies are allowed to withdraw internship offers before the application deadline expires.

- **Internship Search**

Students can search for internship opportunities using filtering options based on standardized information required by the platform while posting an internship offer. This functionality ensures a user-friendly browsing experience, allowing students to efficiently find internships aligned with their preferences.

- **Generation of Recommendations**

The platform automatically generates recommendations for students based on their profiles and, asymmetrically, suggests suitable candidates to companies. The system employs various strategies to provide accurate and relevant suggestions. Both parties can evaluate the received recommendations and decide to accept or decline them based on their

interests.

- **Support in the Selection Process**

The platform facilitates the selection process by enabling companies to contact the candidates, schedule interviews and collect responses from students to questions shared by the company for their evaluation. In the end, it reports to students the outcomes of their interviews.

- **Communication Functionalities and Monitoring**

The platform supports communication between students and companies, allowing them to provide updates about the status and progress of their ongoing internships and monitor them. Communication of problems and complaints is made possible through private spaces. Universities can monitor internships to handle complaints and issue them when possible, interrupting internships if necessary.

- **Collecting Feedback**

To improve its recommendation system, the platform gathers feedback from students and companies both during and after internships. This feedback loop allows a continuous refinement, to better address the interests of parties over time.

- **Personalized Suggestions**

The platform assists users by providing personalized suggestions to produce more appealing descriptions. In particular, it provides suggestions to students for improving their profiles and it offers guidance to companies on how to optimize their internship descriptions.

2.2.1. Requirements

In this section, the requirements for the system to be developed are outlined:

[R1] Upon request, the system shall allow the User to sign up to the platform, as long as they submit all the required information, they don't already have a profile in the platform and their identity and role (Student, Company or University) are verified.

[R2] Upon request, the system shall allow the requesting User to log in to the platform, granting him access to their profile as long as their authentication is successful.

[R3] Upon request, the system shall allow the requesting User to update their profile, as long as they provide all the necessary information.

[R4] Upon request, the system shall allow a Company to publish a new internship offer, as long as it provides all the required information and the latter is compliant with platform guidelines.

[R5] Whenever a Company publishes an internship offer, the system shall add it to the list of all the internship offers.

[R6] Upon request, the system shall allow the requesting Company to update information for any of their open internship offers, as long as it provides all the necessary information.

[R7] Upon request, the system shall allow the requesting Company to withdraw any of their open internship offers.

[R8] Upon request, the system shall allow a Student to search for desired internship offers by applying optional filters to the list.

[R9] Whenever receiving a list of filter attributes for searching internship offers, the system shall return the list of all the offers matching the selected criteria.

[R10] Upon request, the system shall allow the requesting Student to apply to an internship offer, as long as that offer's application deadline has not expired.

[R11] Whenever a Student applies for an internship offer, the system shall add them to the list of candidates for that offer.

[R12] Whenever a Student applies for an internship offer, the system shall mark all the "Unhandled" recommendations of that Student about that internship offer as "Accepted".

[R13] Whenever a Student applies for an internship offer, the system shall discard all the "Unhandled" recommendations of the Company offering it about that Student in the context of that offer.

[R14] Whenever an internship offer is withdrawn by its publishing Company, the system shall discard all applications to that offer.

[R15] Whenever an internship offer is withdrawn by its publishing Company, the system shall discard all generated recommendations linked to that offer.

[R16] Whenever a recommendation aimed at a Party is generated, the system shall add that recommendation to that Party's profile, as long as there is not another "Unhandled" recommendation about the other Party in the context of the same offer.

[R17] Whenever a new Student signs up to the platform, the system shall generate, for every internship offer matching that Student's data, a recommendation about them aimed at the Company advertising that offer, as long as the latter's application deadline has not expired.

[R18] Whenever a Student updates their profile, the system shall generate, for every internship offer matching that Student's updated data, a recommendation about them aimed at the Company advertising that offer, as long as the latter's application deadline has not expired.

[R19] Whenever a Company publishes a new internship offer, the system shall generate, for every Student matching with that internship offer's data, a recommendation about it aimed at that Student.

[R20] Whenever a Company updates data for an internship offer, the system shall generate, for every Student matching with that internship offer's updated data, a recommendation about it aimed at that Student.

[R21] Whenever an internship offer is withdrawn by its publishing Company or its application deadline expires, the system shall discard all the recommendations about it, regardless of whether they have been accepted or not.

[R22] Upon request, the system shall allow the requesting Party to manage their received recommendations by accepting or refusing them, if those have not already expired.

[R23] Whenever a Student accepts one of their received recommendations, the system shall apply the requesting Student to the internship offer to which the recommendation refers.

[R24] Whenever a Company accepts one of their received recommendations, the system shall generate a symmetric recommendation to the corresponding Student and add it to the latter's list of recommendations, as long as the generated recommendation is not already present in it.

[R25] After the application deadline of an internship has expired, the system shall allow the publishing Company to contact a Student who had previously applied to that offer in order to plan a future interview with them, if none has been planned yet.

[R26] Whenever a Student receives an interview proposal, the system shall allow that Student to either accept it or refuse it by providing a reason.

[R27] Whenever an interview has to be carried out in-platform, the system shall allow the interviewing Company to submit questions to the Student involved.

[R28] Whenever a Company submits questions to a Student for an in-platform interview, the system shall allow that Student to answer those questions, reporting them to the interviewing Company.

[R29] Upon request, the system shall allow a Company to evaluate the answers received from a Student in one of their interviews, by registering that interview's result.

[R30] Whenever a Company evaluates an interview (both in-platform and in-person), the system shall inform the corresponding Student of the registered outcome.

[R31] Whenever the interview results for all the candidates for an internship offer have been registered into the platform, the system shall close the selection process of that offer.

[R32] Upon request, the system shall allow the requesting Party to provide new information about any of the ongoing internships in which that Party is involved.

[R33] Upon request, the system shall yield to the requesting Party all the information about one of the ongoing internships it is involved in.

[R34] Upon request, the system shall allow the requesting Party to report a problem occurring in one of the ongoing internships it is involved in.

[R35] When receiving a problem report about an internship from a Party, the system shall forward it to the University of the Student involved in that internship.

[R36] Upon request, the system shall allow the requesting University to handle a received problem regarding an ongoing internship in which one of its Students is taking part.

[R37] Upon request, the system shall allow the requesting Party to report feedback about an internship in which it has been actively involved, if that internship has been completed.

[R38] Whenever receiving feedback about a completed internship, the system shall process it in order to improve the process for generating recommendations for the future.

[R39] Upon request, the system shall provide a Student with targeted suggestions for optimizing its profile, enabling the Student to improve its appeal and relevance for obtaining more internship offers in the future, if such optimizations can be found.

[R40] Upon request, the system shall provide a Company with targeted suggestions for optimizing a selected internship offer, enabling the Company to make it more attractive to Students and to improve its visibility for the future, if such optimizations can be found.

2.3. User Characteristics

The S&C platform is designed for three categories of users:

Students University students seeking internships. They are typically proficient users, familiar with app navigation and common interaction patterns such as registration, profile management, and document uploads (e.g., CVs). Their primary motivation is to find internships aligned with their skills and career goals efficiently. Students generally access the platform a few times each week while searching for internships and reviewing recommendations, with increased frequency during the interview phases.

Companies Human Resources professionals responsible for posting internship opportunities, reviewing candidate recommendations, conducting the selection process, and providing feedback on ongoing internships. These users may vary in technical proficiency with the platform,

but expect a straightforward interface to accomplish core tasks efficiently. They typically interact with the platform multiple times daily during work hours, particularly during active recruitment periods.

Universities University staff responsible for monitoring internship outcomes and addressing their student-related complaints during internships. While their technical expertise may vary, they require tools for monitoring and complaint management. They have minimal involvement in day-to-day activities on the platform. The typical number of interactions can be a few times per week, depending on the number of students they manage.

All users expect a user-friendly interface, reliable notifications, and support features. The system accommodates varying levels of technical proficiency and ensures accessibility for all user groups.

2.4. Assumptions, Dependencies and Constraints

2.4.1. Domain Assumptions

[DA1] Users always interact with the system only through a device with a reliable connection to the Internet.

[DA2] People who sign up to the platform have an active email address that they can provide to the system.

[DA3] People who sign up to the platform can access the mailbox of the email address provided during the registration process.

[DA4] People who are not Students, Companies or Universities never disguise themselves as being one of those.

[DA5] Students signing up to the platform belong to one of the Universities which have already registered on the platform.

[DA6] Users never provide false information about themselves.

[DA7] Companies never provide false information about their internship offers.

[DA8] Students never withdraw any of their applications to internship offers (equivalently, they apply for internships only if those match their interests and if they really want to undertake them).

[DA9] For each of their internship offers, Companies always try to set up interviews with all and only the Students who have submitted an application.

[DA10] Companies always finalize the selection process and bring it to completion, even if they haven't found any fitting candidate among the interviewed students.

[DA11] Complaints inserted into the system always refer to events that have happened during the internship they are related to.

[DA12] Users regularly upload information about the ongoing internships

[DA13] Universities always act on received complaints and solve exposed issues, promptly terminating the internship if necessary.

[DA14] Feedback inserted into the system after performing an internship is always meaningful to be fed to the statistical analysis.

3 | Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

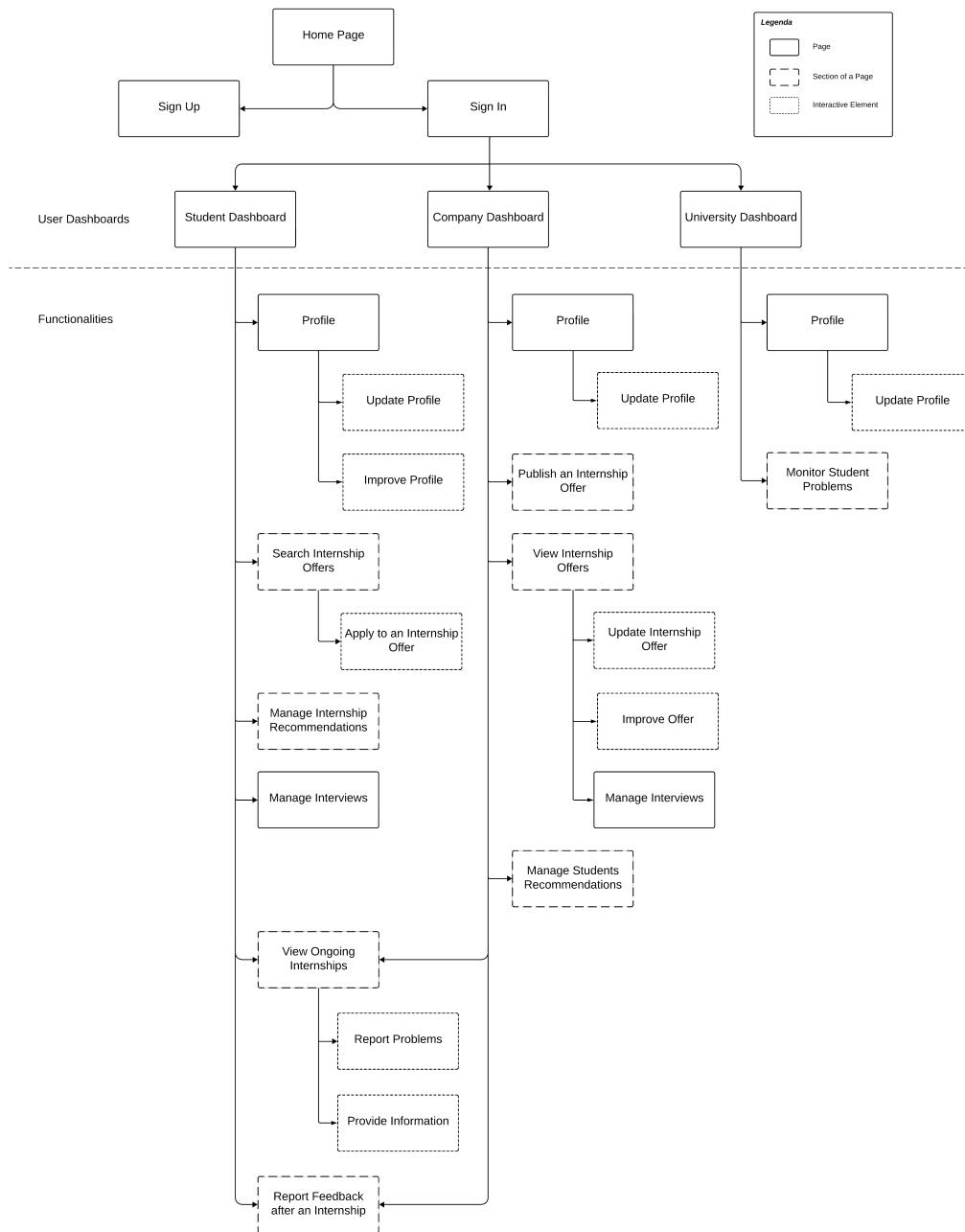


Figure 3.1: WebApp Map

The platform's user interface shall be designed to ensure usability and interactivity, adhering to web standards, briefly discussed in [3.4.1. Standards Compliance](#). The interface shall adapt responsively to various device displays, providing an intuitive layout and functionality across all the supported devices. Interface elements shall be organized in a layout that prioritizes clarity and ease of navigation for all the users.

Below is a description of the primary interfaces needed, as visually displayed in the above map, and how they should be grouped and structured:

Home Page This page serves as the entry point to the platform and offers:

- **Sign-Up (Register):** A page to start the personalized registration for the three distinct user types.
- **Sign-In (Login):** A page to perform login that redirects authenticated users to their respective dashboards.

User Dashboards Each user type is provided with a personalized dashboard, divided into sections offering different functionalities:

1. Student Dashboard

- **Profile:** a page displaying all the student's information and including:
 - **Update Profile:** an interactive element for updating personal and academic information.
 - **Improve Profile:** an interactive element suggesting optimizations based on the content of the profile and CV.
- **Search Internship Offers:** a section that includes a search bar with filtering options to search offers. Each result of the query is an interactive element displaying an overview of the information related to an internship offer; clicking on the internship offer expands its details and enables an interactive element to perform the application.
- **Manage Internship Recommendations:** a section listing the received internship recommendations. Each recommendation is an interactive element, and students can visualize details about the internship offer associated to each recommendation and accept, reject, or postpone the recommendations.
- **Manage Interviews:** a page displaying interview invitations. Each invitation is an interactive element, and Students can confirm or decline based on their availability.

Questions posted by companies are also interactive elements and are displayed here.

2. Company Dashboard

- **Profile:** a page displaying all the company's information and including:
 - **Update Profile:** an interactive element for updating company information.
- **Publish an Internship Offer:** an interactive element to publish new internship offers.
- **View Internship Offers:** a section listing all published internship offers. Each internship offer is an interactive element displaying an overview of the information related to an internship offer; clicking on the internship offer expands its details and enables:
 - **Update Internship Offer:** an interactive element for updating offer information.
 - **Improve Offer:** an interactive element to receive optimizations based on the offer details.
 - **Manage Interviews:** a page allowing companies to handle interview invitations and post questions to candidates.
- **Manage Student Recommendations:** a section listing student's recommendations. Each recommendation is an interactive element, and the company can visualize its profile and accept, reject, or postpone it.

Common Functionalities in Student and Company Dashboards

- **View Ongoing Internships:** a section displaying all the ongoing internships. Each internship is an interactive element and by clicking on it, it enables:
 - **Report Problems:** an interactive element to report problems to be handled by the university.
 - **Provide Information:** an interactive element to write relevant information during an internship.
- **Report Feedback after an Internship:** a section to give feedback upon completion of an internship.

3. University Dashboard

- **Profile:** a page displaying all university-related information and including:

- **Update Profile:** an interactive element for updating information.
- **Monitor Student Problems:** a section displaying issues reported by students during internships. Each report is an interactive element and provides tools to review, resolve, and update statuses.

3.1.2. Software Interfaces

The system is meant to be a platform-independent WebApp, which does not rely on external APIs or essential third-party software to perform its core functionalities. However, the system shall adhere to the following requirements:

- It shall function across the most widely adopted operating systems, including but not limited to Microsoft Windows, MacOSX, Linux, Android, iOS and ChromeOS.
- It shall function across the most widely used browsers, including, but not limited to, Google Chrome, Opera, Mozilla Firefox, Safari, and Microsoft Edge, without requiring additional software installation on the user's device.
- It shall interact appropriately with the chosen services or network protocols for email management, in order to carry on sign-up and sign-in functionalities.

No additional software installations or configurations are required on the user's device beyond the availability of a supported web browser.

3.1.3. Communication Interfaces

The system shall support standard communication protocols for reliable interaction between the components. Specifically, the following requirements apply:

- The system shall operate over standard internet connections, including wired and wireless networks (e.g. Ethernet, Wi-Fi, mobile data).
- The system shall support encrypted communication/secure sessions (e.g. through TLS/SSL) to protect sensitive data, including user credentials and personal information, during transmission.
- The system shall utilize a widely adopted web communication protocol (such as HTTPS) to ensure secure and reliable data transmission between the user's browser and the web server.
- The system shall not require users to install additional software to facilitate communication.

- The system shall minimize communication latency to provide a responsive user experience, with server response times aligning with industry standards for web applications.

3.1.4. Hardware Interfaces

The system shall ensure compatibility with common user devices and hosting infrastructure, designed to provide performance and accessibility.

User Devices

The platform shall support access from a wide range of devices commonly used by end-users, including but not limited to: desktop and laptop computers, tablets and smartphones.

The platform shall ensure responsiveness and usability across different device types and display resolutions, requiring hardware with free space for caching that can run modern web applications efficiently.

Hosting Infrastructure

The system shall be deployed on an infrastructure capable of supporting:

- Simultaneous access by multiple users with minimal latency
- Data processing and storage necessary for handling user interactions and background operations

The hosting environment shall be scalable to accommodate growth in usage and computational demands while maintaining reliable service availability.

3.2. Functional Requirements

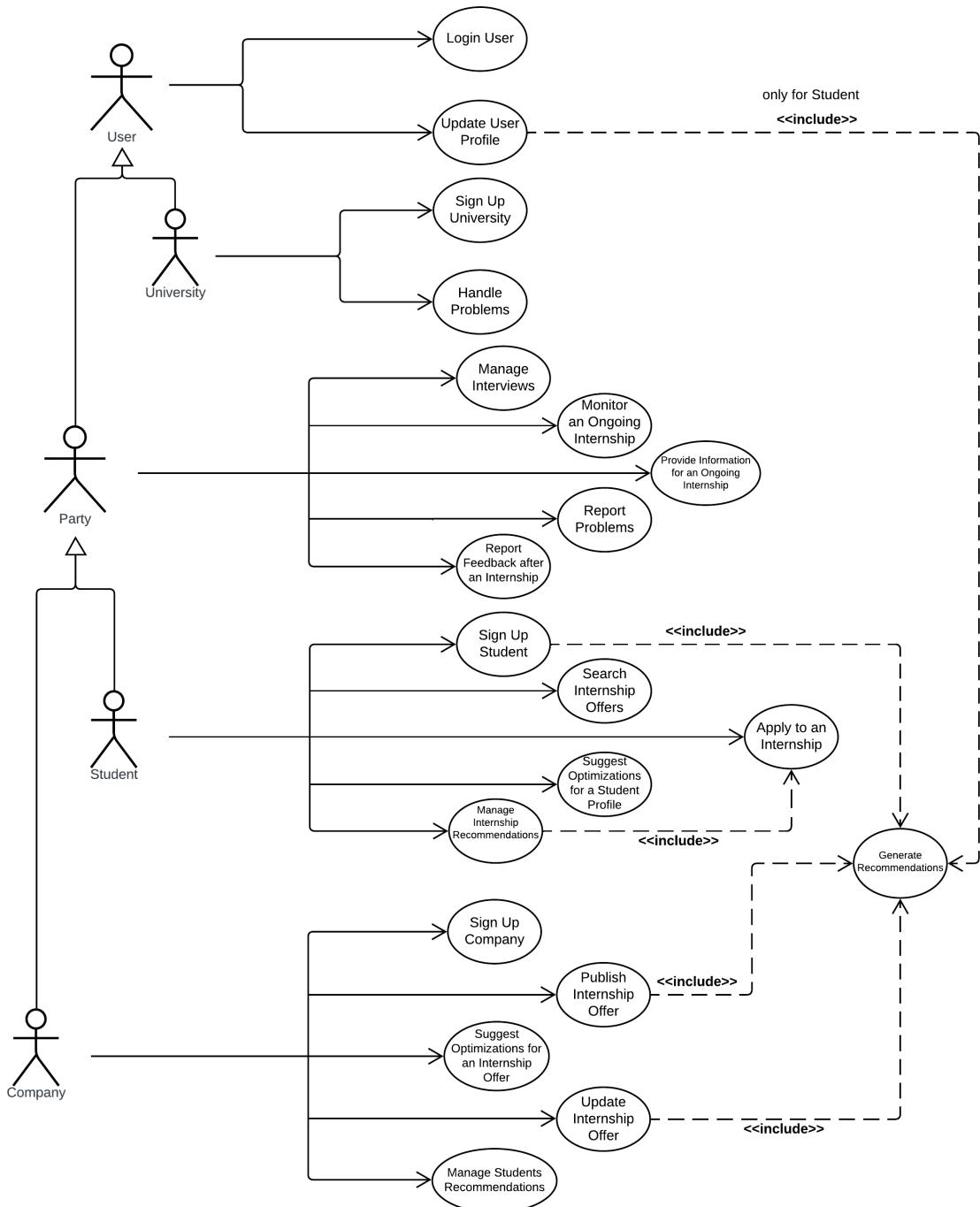


Figure 3.2: Use Case Diagram

3.2.1. Use Cases and Activity Diagrams

UC1. Sign Up by a Student

Actor	Student
Entry Conditions	The Student is not logged into the S&C platform.
Flow of Events	<ol style="list-style-type: none"> 1. On the homepage, the Student clicks the "Sign Up" button, entering the students' registration page. 2. The Student provides the required details: name, surname, date of birth, gender, email, and password. 3. The Student confirms the provided information by clicking the "Register" button. 4. The system sends to the indicated mailbox a confirmation email with a link that expires in 24 hours for account verification purposes. 5. The Student clicks the link in the confirmation email before it expires and logs into the platform. 6. On the profile page, the Student completes their profile by uploading their CV, selecting their university from a drop-down list and adding relevant details: skills, education, and career aspirations. 7. The system starts an instance of the process for identifying new recommendations via the <u>UC. Generate Recommendations</u> functionality.
Exit Conditions	The profile is complete and the Student has access to its functionalities.
Exceptions	<ul style="list-style-type: none"> • The email address is already linked to an existing account: an error message is shown, and the Student is redirected to the login page. • The password does not meet the platform's security requirements: an error message is displayed, and the Student is prompted to correct the password. • Some mandatory fields are missing: the system doesn't allow the Student to complete the procedure until all the mandatory fields are filled out. • The confirmation link sent to the indicated mailbox expires: all the information previously inserted into the system by the Student is discarded, and the link is invalidated.

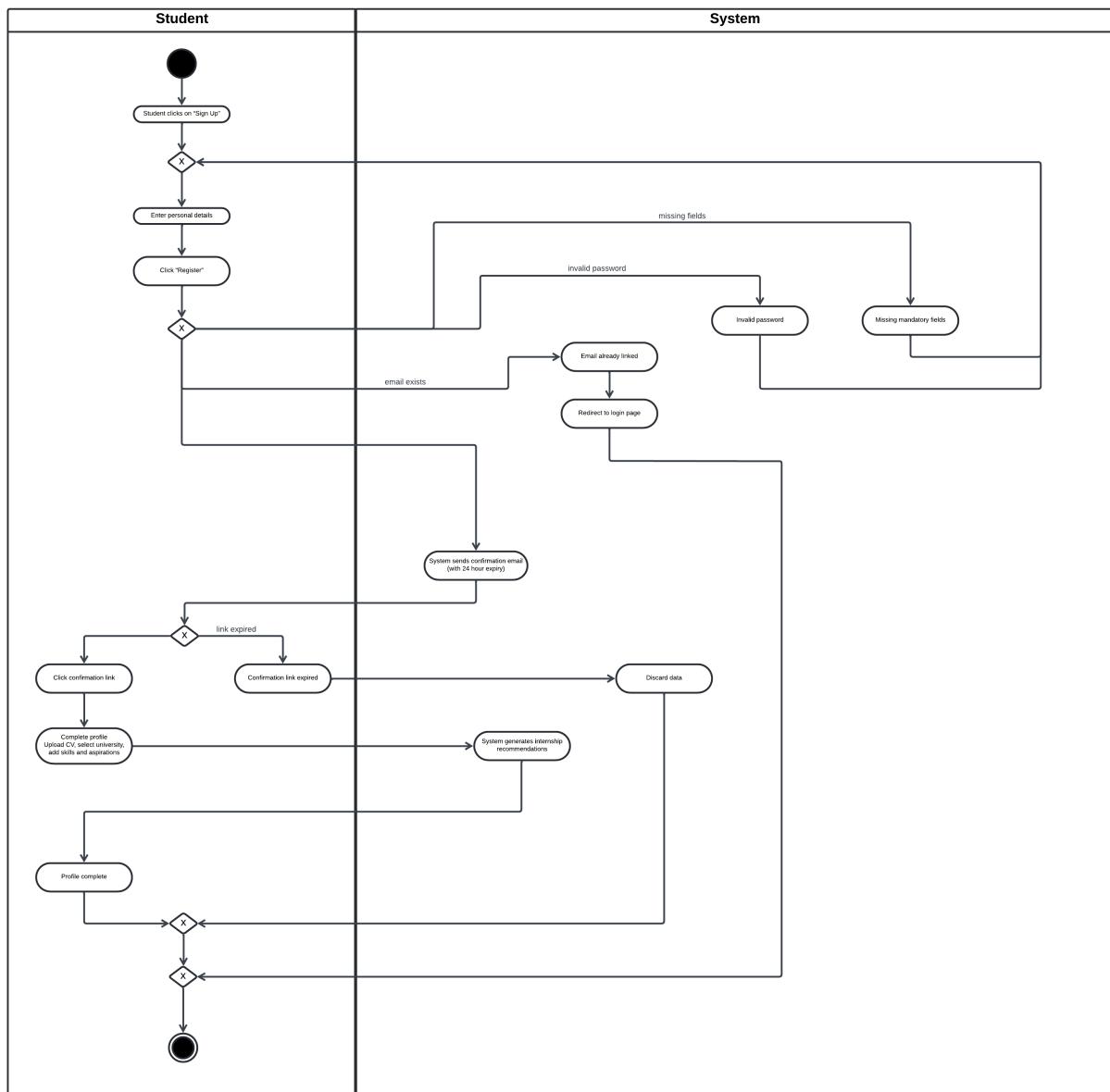


Figure 3.3: Sign Up by a Student

UC2. Sign Up by a Company

Actor	Company
Entry Conditions	The Company is not logged into the S&C platform.
Flow of Events	<ol style="list-style-type: none"> 1. On the homepage, the Company clicks the "Sign Up" button, entering the companies' registration page. 2. The Company provides the required details: company name, location, email, password, phone number. 3. The Company confirms the provided information by clicking the "Register" button. 4. The system sends to the indicated mailbox a confirmation email with a link that expires in 24 hours for account verification purposes. 5. The Company clicks the link in the confirmation email and logs into the platform. 6. On the profile page, the Company completes the company's profile by adding relevant details: company description, mission, vision, and field in which it operates, and uploading its logo.
Exit Conditions	The profile is complete and the Company has access to all its functionalities.
Exceptions	<ul style="list-style-type: none"> • The email address is already linked to an existing account: an error message is shown, and the Company is redirected to the login page. • The password does not meet the platform security requirements: An error message is displayed, and the Company is prompted to correct the password. • Some mandatory fields are missing: the system does not allow the Company to complete the procedure until all the mandatory fields are filled out. • The confirmation link sent to the indicated mailbox expires: all information previously inserted into the system by the Company is discarded and the link is invalidated.

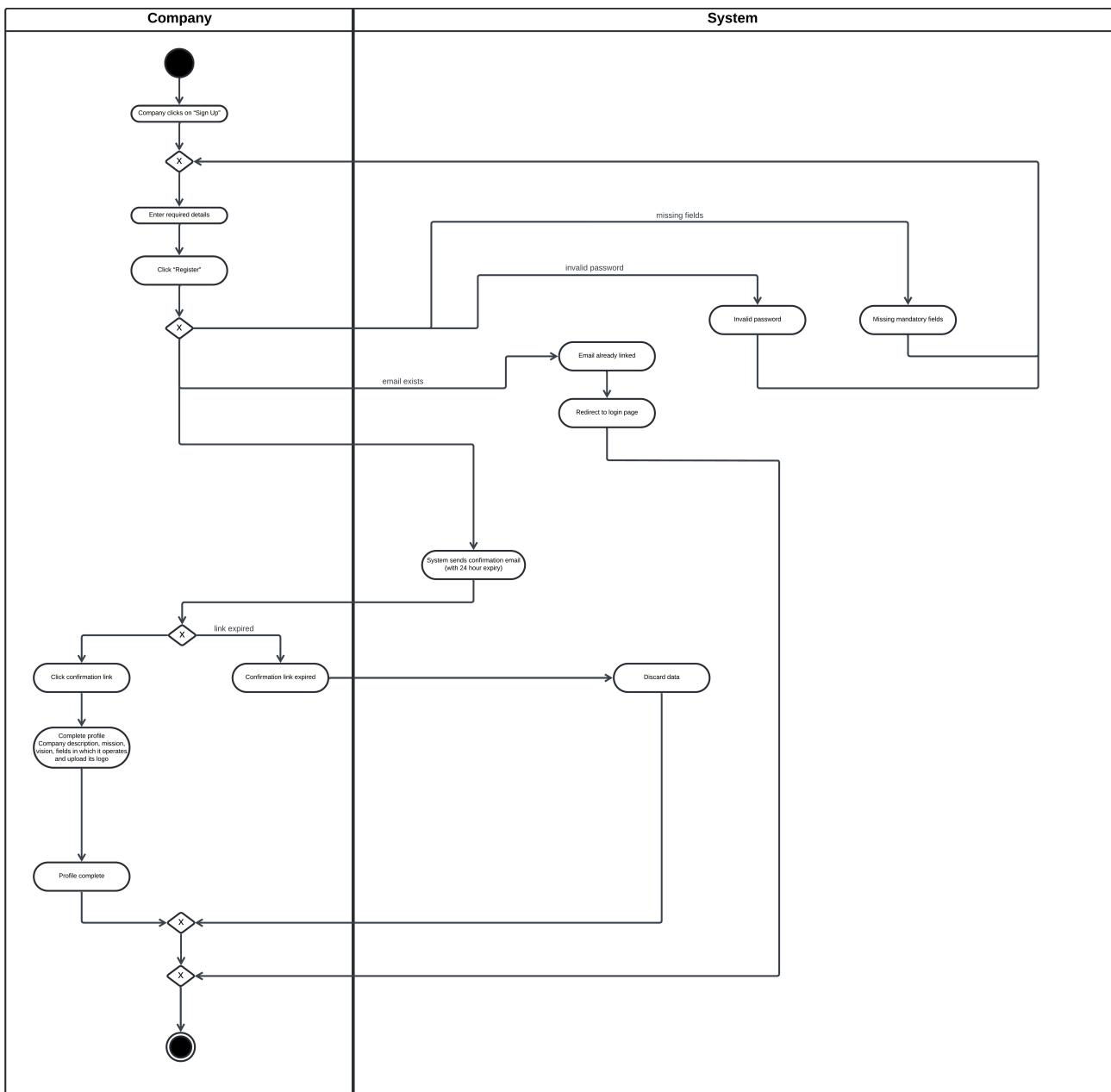


Figure 3.4: Sign Up by a Company

UC3. Sign Up by a University

Actor	University
Entry Conditions	The University is not logged into the S&C platform.
Flow of Events	<ol style="list-style-type: none"> 1. On the homepage, the University clicks the "Sign Up" button, entering the companies' registration page. 2. The University provides the required details: university name, location, email, password, phone number. 3. The University confirms the provided information by clicking the "Register" button. 4. The system sends to the indicated mailbox a confirmation email with a link that expires in 24 hours for account verification purposes. 5. The University clicks the link in the confirmation email and logs into the platform. 6. On the profile page, the University completes the university's profile by adding relevant details: faculties, key contacts for internships, university website
Exit Conditions	The profile is complete and the University has access to all its functionalities.
Exceptions	<ul style="list-style-type: none"> • The email address is already linked to an existing account: an error message is shown, and the University is redirected to the login page. • The password does not meet the platform security requirements: An error message is displayed, and the University is required to correct the password. • Some mandatory fields are missing: the system does not allow the University to complete the procedure until all the mandatory fields are filled out. • The confirmation link sent to the indicated mailbox expires: all information previously inserted into the system by the University is discarded and the link is invalidated.

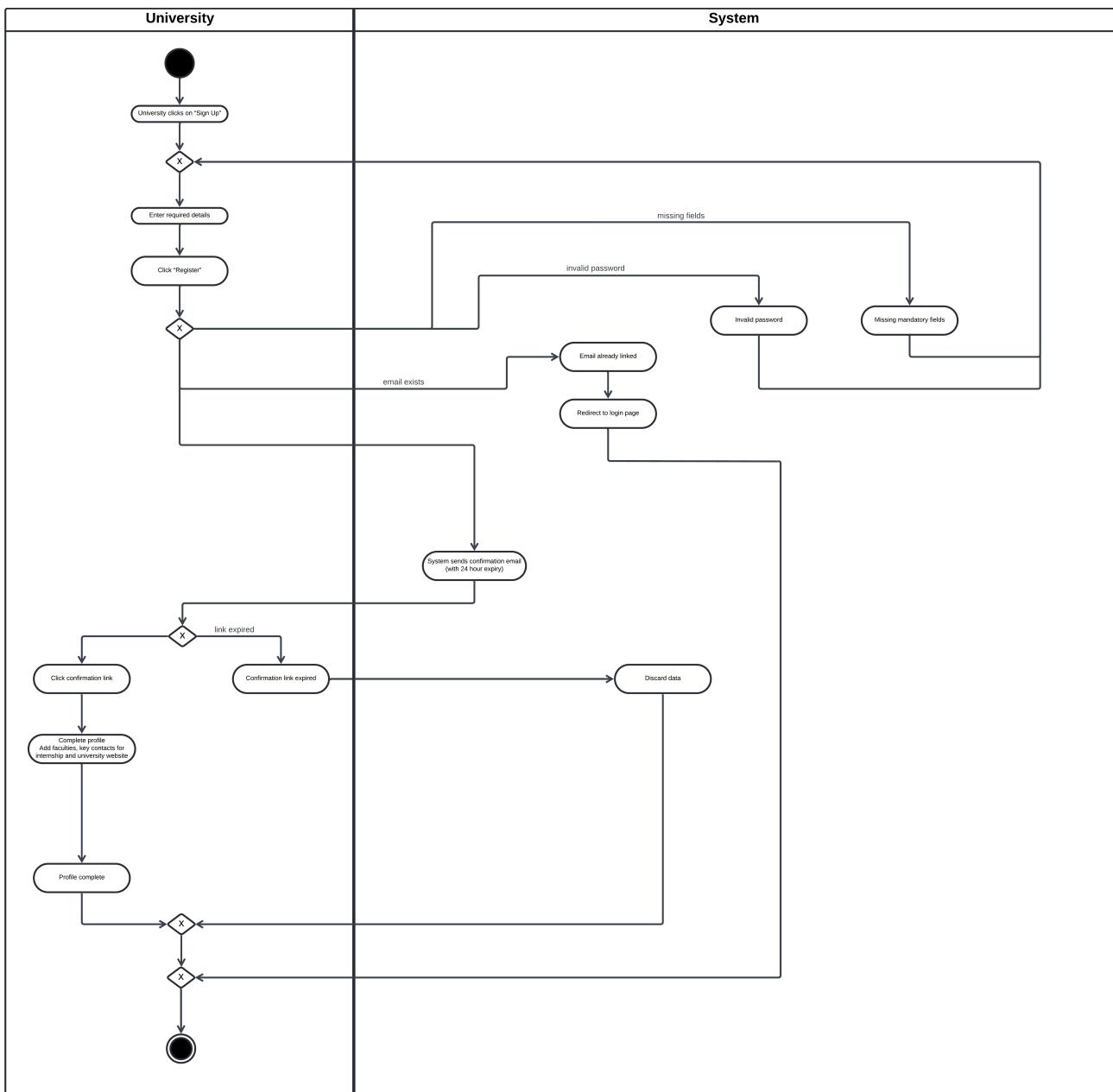


Figure 3.5: Sign Up by a University

UC4. Log In by a User

Actor	User (Student, Company or University)
Entry Conditions	The User is not already logged into the S&C platform.
Flow of Events	<ol style="list-style-type: none"> 1. On the homepage, the User clicks the "Login" button, which displays the login form. 2. The User enters their email and password into the designated fields. 3. The User clicks the "Login" button. 4. The system validates the provided credentials. 5. The system redirects the User to the dashboard page.
Exit Conditions	The User is successfully logged in.
Exceptions	<ul style="list-style-type: none"> • The inserted credentials are incorrect: the system displays an error message indicating that the credentials are invalid and the User remains on the login page. • The account hasn't been verified yet: if the User has not confirmed their email and the confirmation link has not expired yet, the system shows a message requesting to complete the verification process.

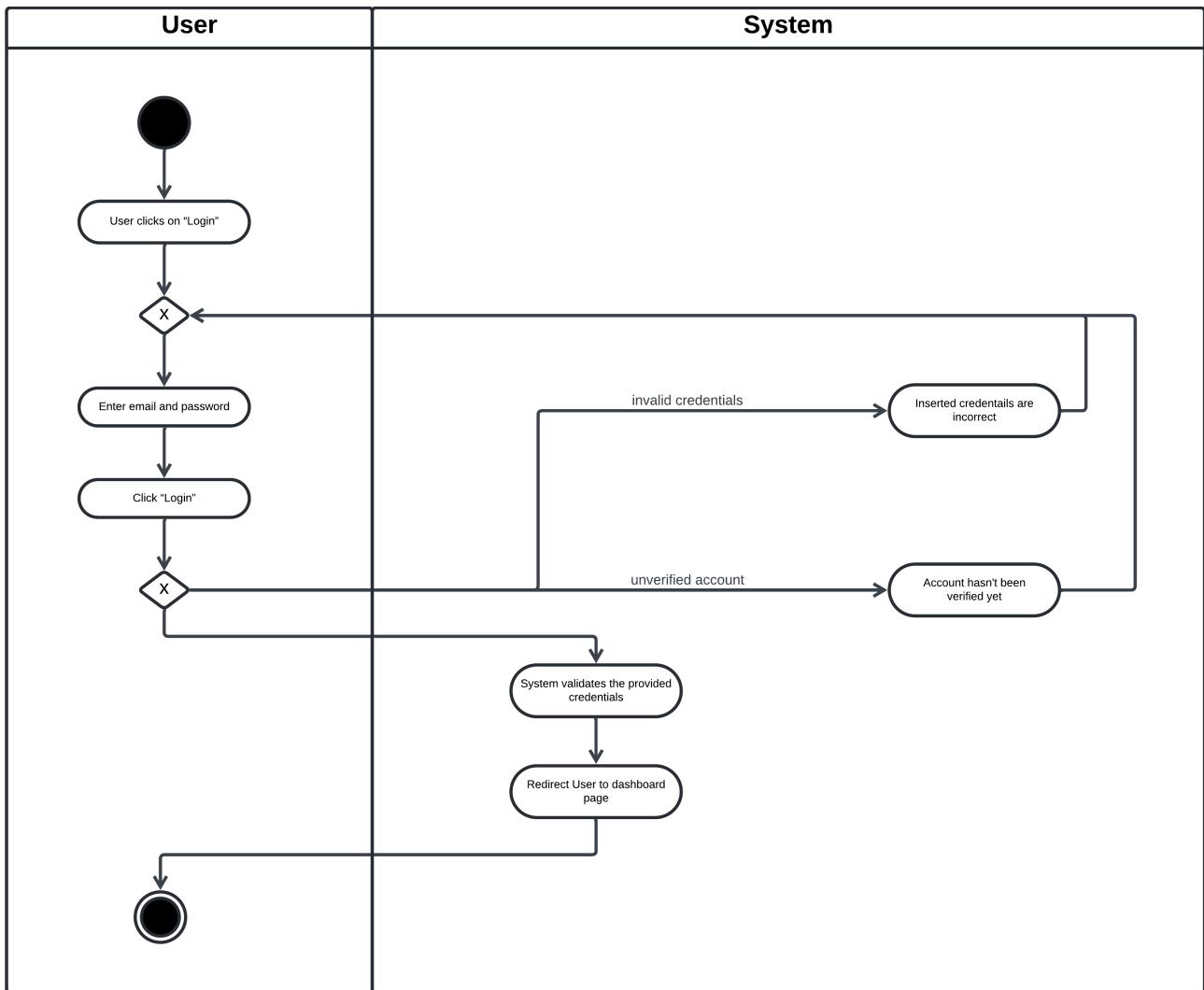


Figure 3.6: Login by a User

UC5. Update User Profile

Actor	User
Entry Conditions	The User is logged into the S&C platform.
Flow of Events	<ol style="list-style-type: none"> 1. In their "Profile" section, the User clicks the "Edit" button. 2. The User updates the desired details of its profile. 3. The User confirms the applied changes by clicking the "Apply Changes" button. 4. If the User is a Student and the updated details are relevant, the system starts an instance of the process for identifying new recommendations via the UC. Generate Recommendations functionality.
Exit Conditions	The User's profile is updated and its changes are recorded in the system.
Exceptions	<ul style="list-style-type: none"> • Some mandatory fields are left empty: the system doesn't allow the User to complete the procedure until all the mandatory fields are filled out.

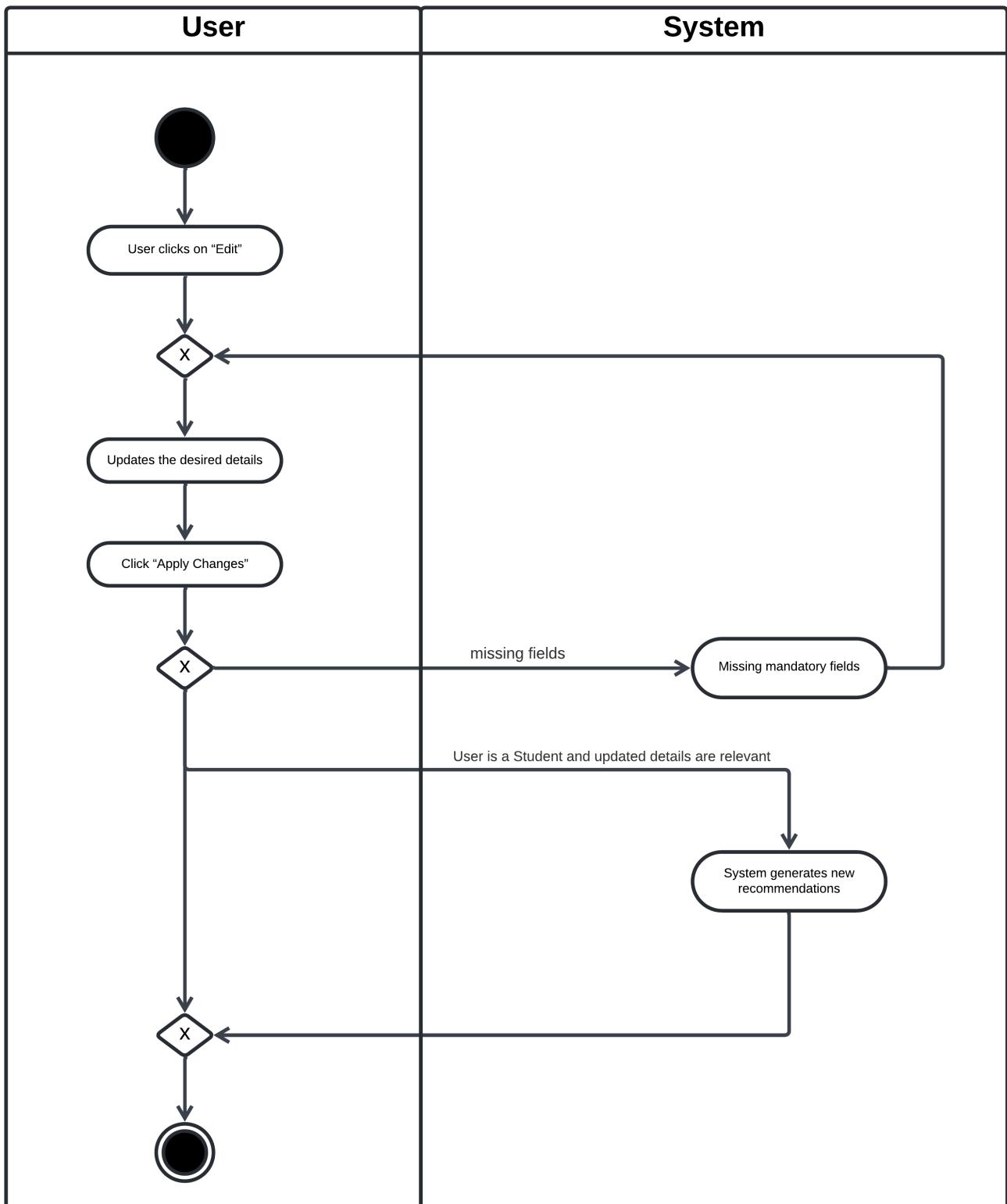


Figure 3.7: Update User profile

UC6. Publish an Internship Offer

Actor	Company
Entry Conditions	The Company is logged into the S&C platform.
Flow of Events	<p>1. In the dashboard, the Company clicks the "Create New Offer" button, entering the internship creation page.</p> <p>2. The Company fills out the internship creation form, inserting:</p> <ul style="list-style-type: none"> • Application domain • Tasks to be performed • Required skills • Internship duration • Compensation terms • Location (on-site, hybrid, or remote) • Application deadline <p>3. The Company clicks the "Submit" button, waiting for the automatic data verification.</p> <p>4. The system verifies the provided data to ensure the provided information is compliant with platform guidelines, and the data is consistent and accurate.</p> <p>5. The system publishes the internship offer on the platform, making it visible to all students.</p> <p>6. The system confirms to the Company that its offer has been published.</p> <p>7. If the updated details are relevant, the system starts an instance of the process for identifying new recommendations via the <u>UC. Generate Recommendations</u> functionality.</p>
Exit Conditions	The internship offer is published on the platform and accessible to students.
Exceptions	<ul style="list-style-type: none"> • Some mandatory fields are missing: the system doesn't allow the Company to complete the procedure until all the mandatory fields are filled out. • Some information is not compliant with platform guidelines: the system notifies the Company about the issue and requires revisions.

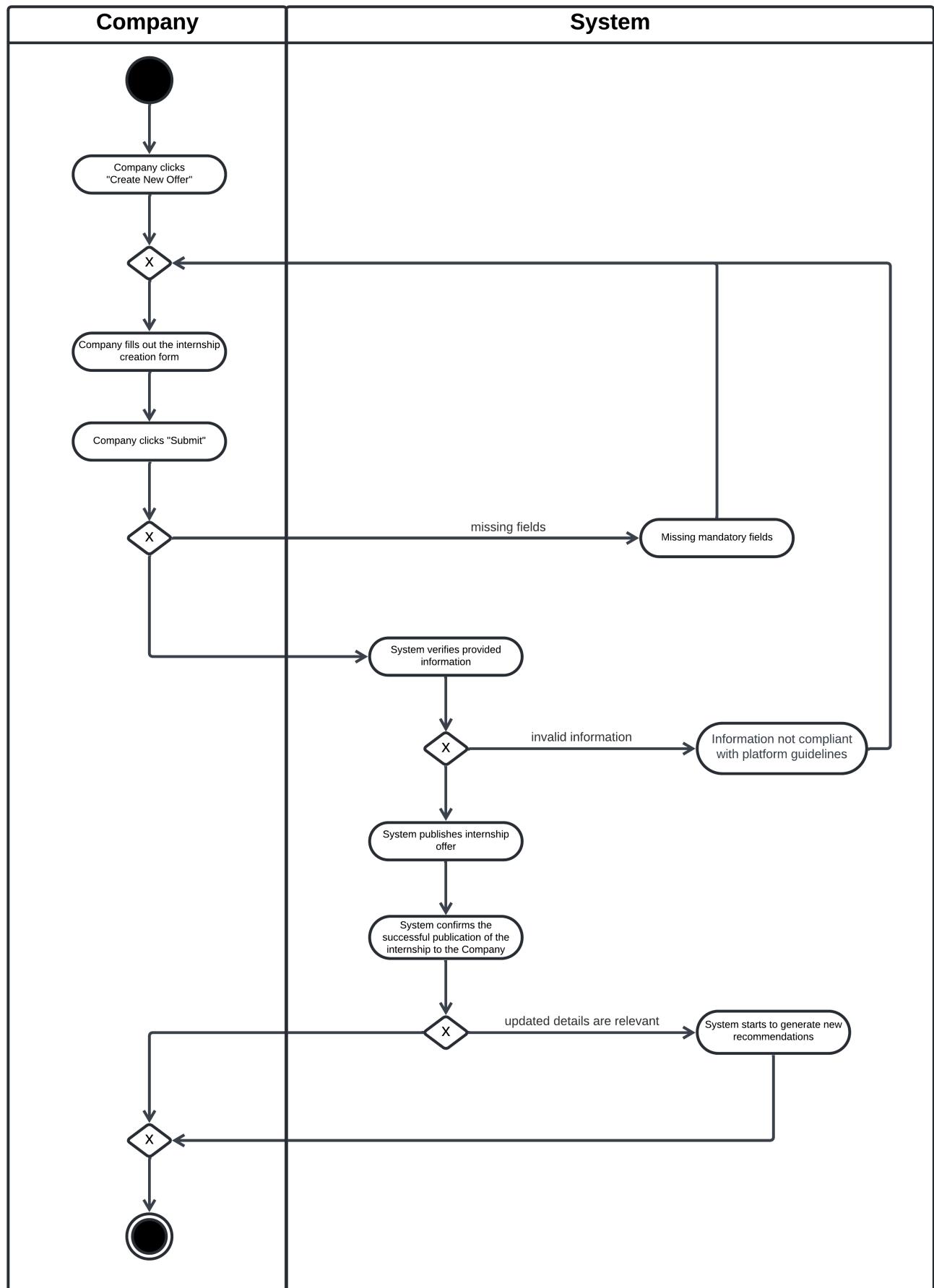


Figure 3.8: Publish an Internship Offer

UC7. Update Internship Offer

Actor	Company
Entry Conditions	The Company is logged into the S&C platform and the selected internship offer's application deadline has not expired yet.
Flow of Events	<ol style="list-style-type: none"> 1. On the selected internship offer's page, the Company clicks the "Edit" button. 2. The Company updates the desired details of the internship offer or chooses to withdraw it entirely by clicking the "Withdraw Offer" button. 3. The Company acts accordingly based on their decision: <ol style="list-style-type: none"> 3.1. If the Company updates the details, it confirms the applied changes by clicking the "Apply Changes" button. 3.2. If the Company chooses to withdraw the offer, a confirmation prompt appears, asking to finalize the withdrawal. Upon confirmation, the internship offer is marked as "Closed" and becomes inaccessible to students. 4.1. If updates are confirmed, the system records the changes. 4.2. If the offer is withdrawn, the system cancels all pending processes related to the offer, such as identifying new recommendations or managing applications. 5. For an updated offer, the system starts an instance of the process for identifying new recommendations via the <u>UC. Generate Recommendations</u> functionality.
Exit Conditions	The selected internship offer is updated and its changes are recorded in the system.
Exceptions	<ul style="list-style-type: none"> • Some mandatory fields are missing: the system doesn't allow the Company to complete the procedure until all the mandatory fields are filled out. • Some information is not compliant with platform guidelines: the system notifies the Company about the issue and requires revisions.

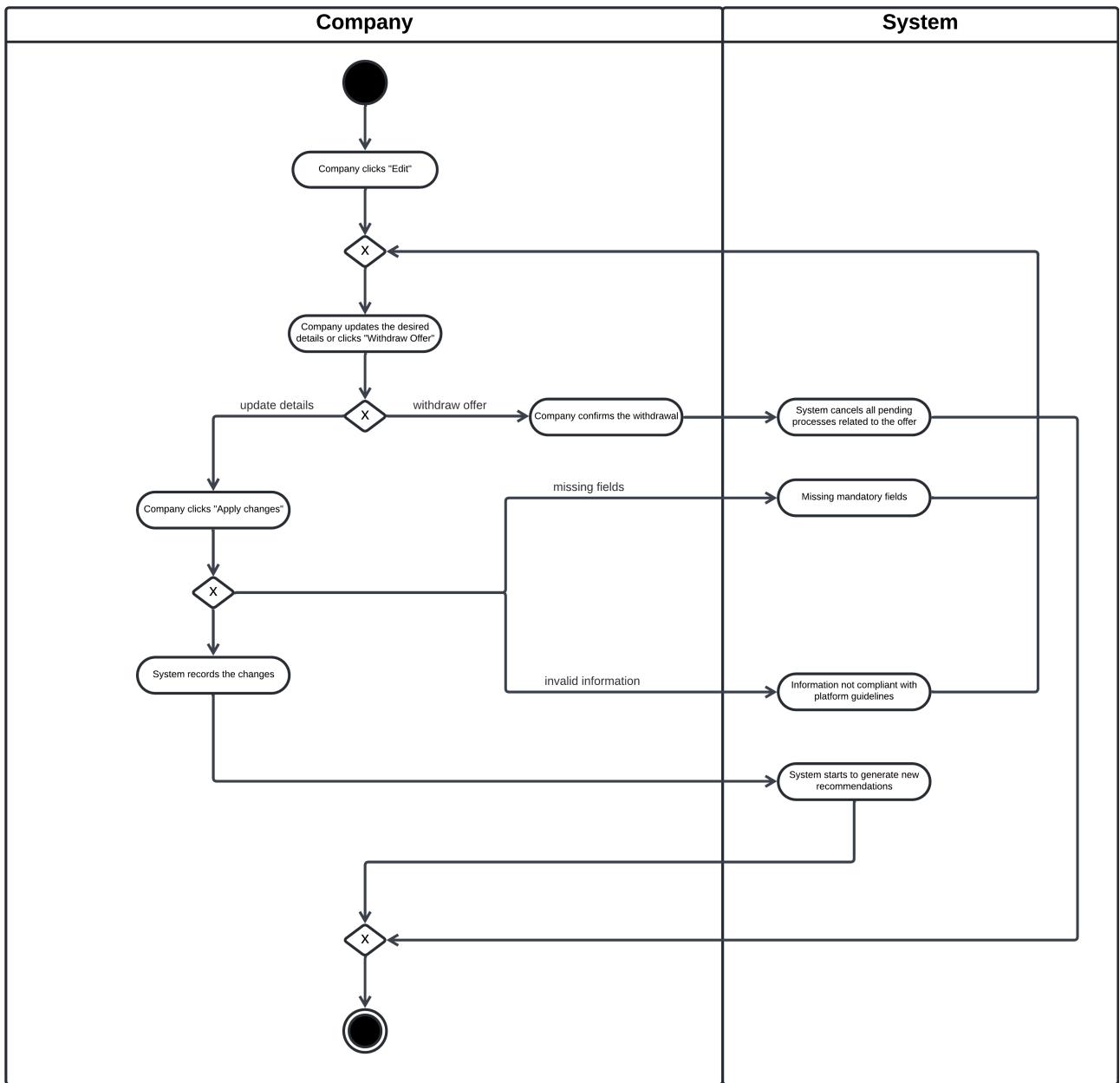


Figure 3.9: Update Internship Offer

UC8. Search Internship Offers

Actor	Student
Entry Conditions	The Student is logged into the S&C platform.
Flow of Events	<p>1. In the dashboard, the Student navigates to the "Search Offers" section.</p> <p>2. The system displays a search interface which has the following optional filters:</p> <ul style="list-style-type: none"> • Domain of interest • Location (on-site, hybrid, or remote) • Internship duration • Compensation terms • Keywords (an offer matches a keyword if it contains that keyword or a semantically similar one in any of its fields). <p>3. The Student selects the desired filters and submits the query to the system.</p> <p>4. The system retrieves and displays a list of internship offers that match the selected criteria.</p> <p>5. Optionally, the Student applies one or more ordering criteria: newest, oldest, most relevant, or most number of applications, by selecting the desired ones from a drop-down list.</p>
Exit Conditions	The system displays the list of internship offers matching the given criteria.
Exceptions	<ul style="list-style-type: none"> • The system doesn't find any matching result: if none of the active internship offers matches the selected criteria, the system displays a message indicating it.

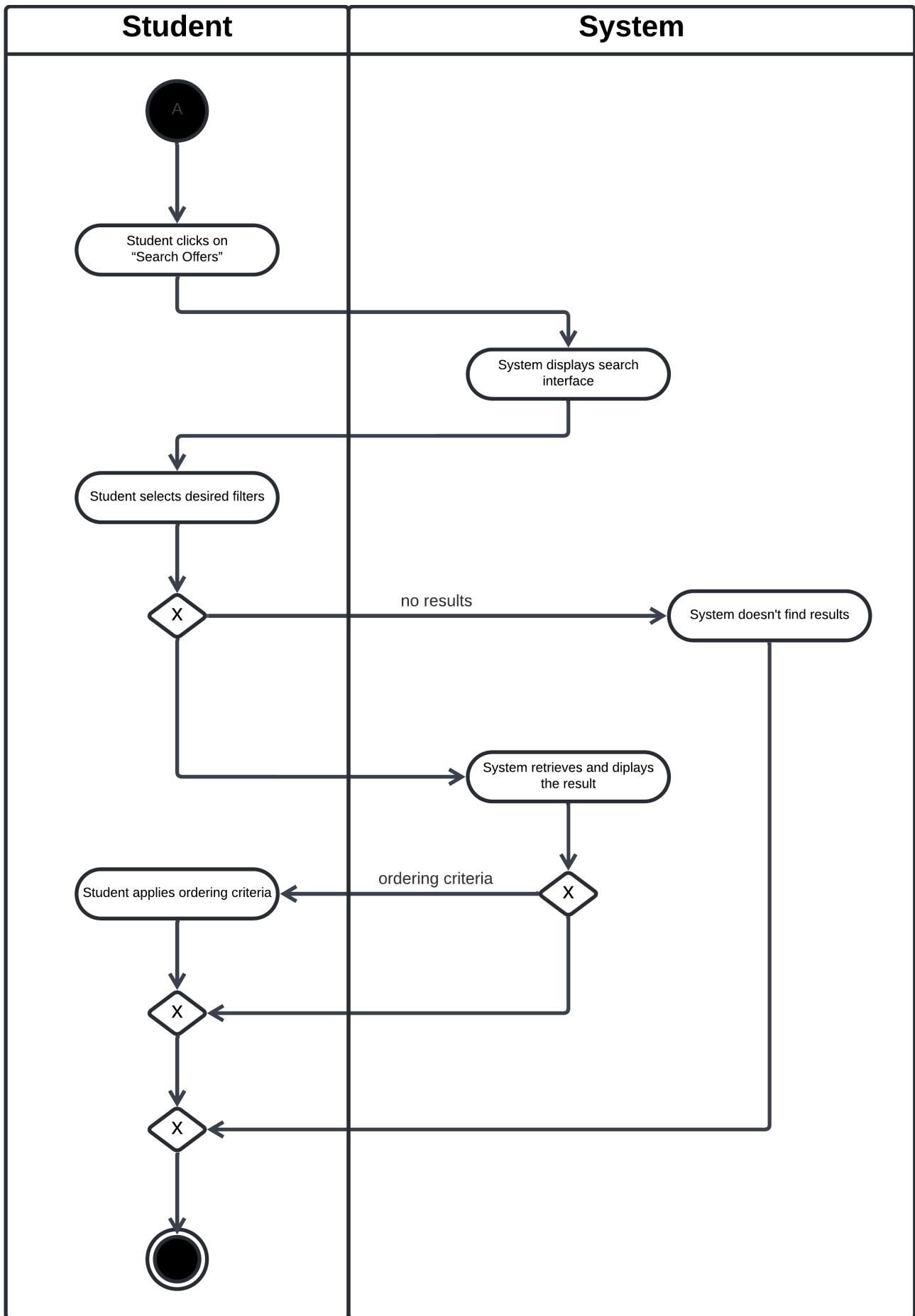


Figure 3.10: Search Internship offer

UC9. Apply to an Internship Offer

Actor	Student
Entry Conditions	The Student is logged into the S&C platform and is on the page of an internship offer.
Flow of Events	<ol style="list-style-type: none"> 1. On the internship offer page, the Student clicks the "Apply" button. 2. The system adds the application to the internship offer's candidates list. 3. The system marks all the "Unhandled" recommendations of the Student about the selected internship offer as "Accepted". 4. The system sends a notification to the profile of the Company to whom the internship offer belongs. 5. The system confirms to the Student that the application has been successfully registered into the system.
Exit Conditions	The Student has successfully made an application to an internship offer of his choice.
Exceptions	<ul style="list-style-type: none"> • The deadline for applying to the internship has expired: the "Apply" button cannot be clicked and the Student has to give up on applying.

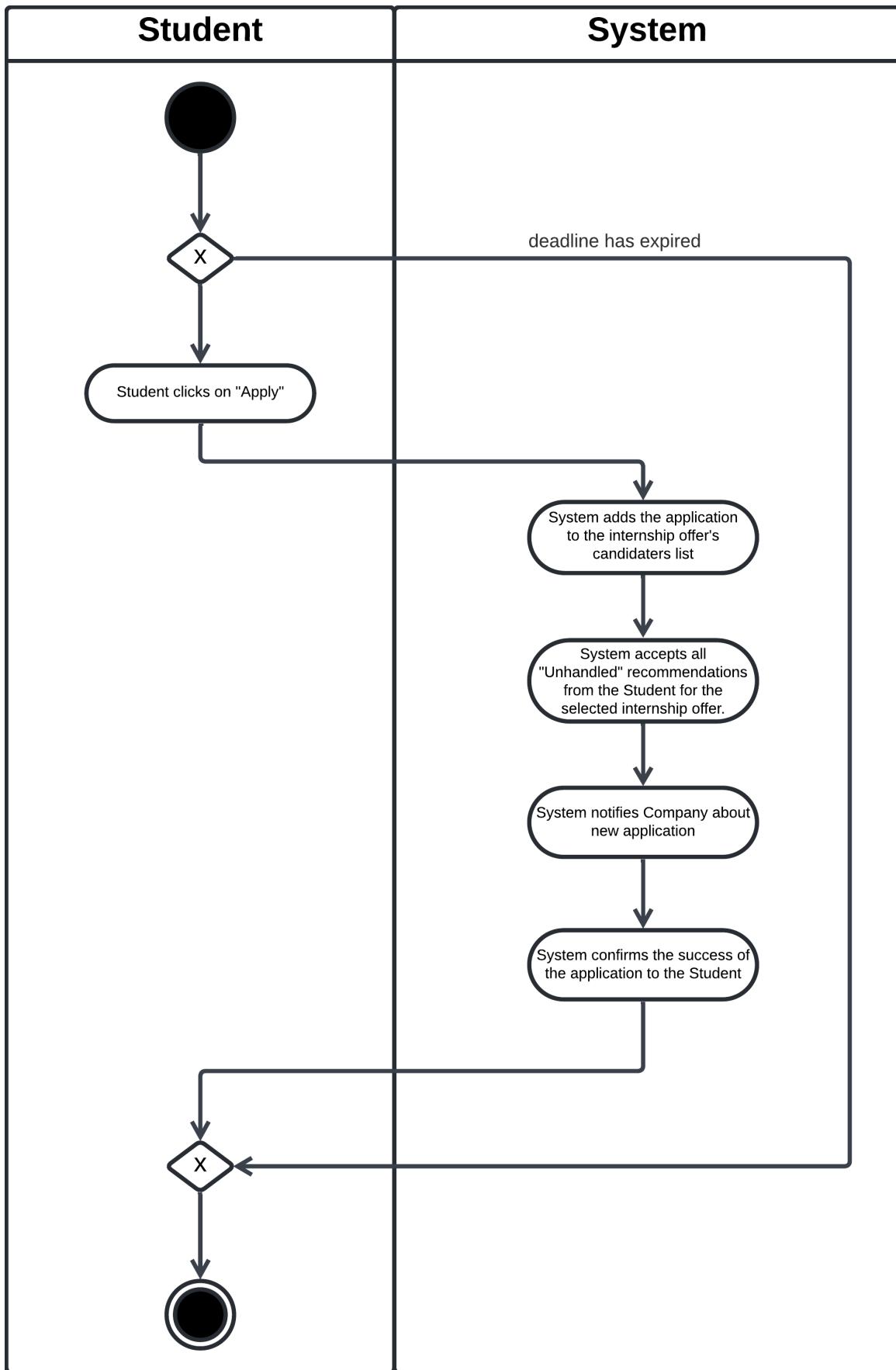


Figure 3.11: Apply to an internship offer

UC10. Generate Recommendations

Actor	None
Entry Conditions	A relevant change in the system's information set has occurred, with the possibility of new matches between a Student and a Company emerging.
Flow of Events	<p>1. By taking into account the newly updated information and a subset of information from all the students, internship offers and companies, including feedback previously expressed by the parties on the platform, the system identifies every new potential match between a Student and an internship offer by a Company that has arisen as a consequence of the update of the information set and had never been identified before. In order to be considered, a match that had already been identified previously and subsequently discarded by a Party needs to have been generated by a substantial change in the information set (with the threshold possibly increasing).</p> <p>2. For every new match, the system generates a recommendation for the identified Parties, which could be only the Student, only the Company or both. Then, the system adds it to the list of recommendations of such Parties.</p>
Exit Condition	The generated recommendations have been posted on the respective Parties' dashboards.
Exceptions	<ul style="list-style-type: none"> • The system doesn't find any match: it terminates the process silently.

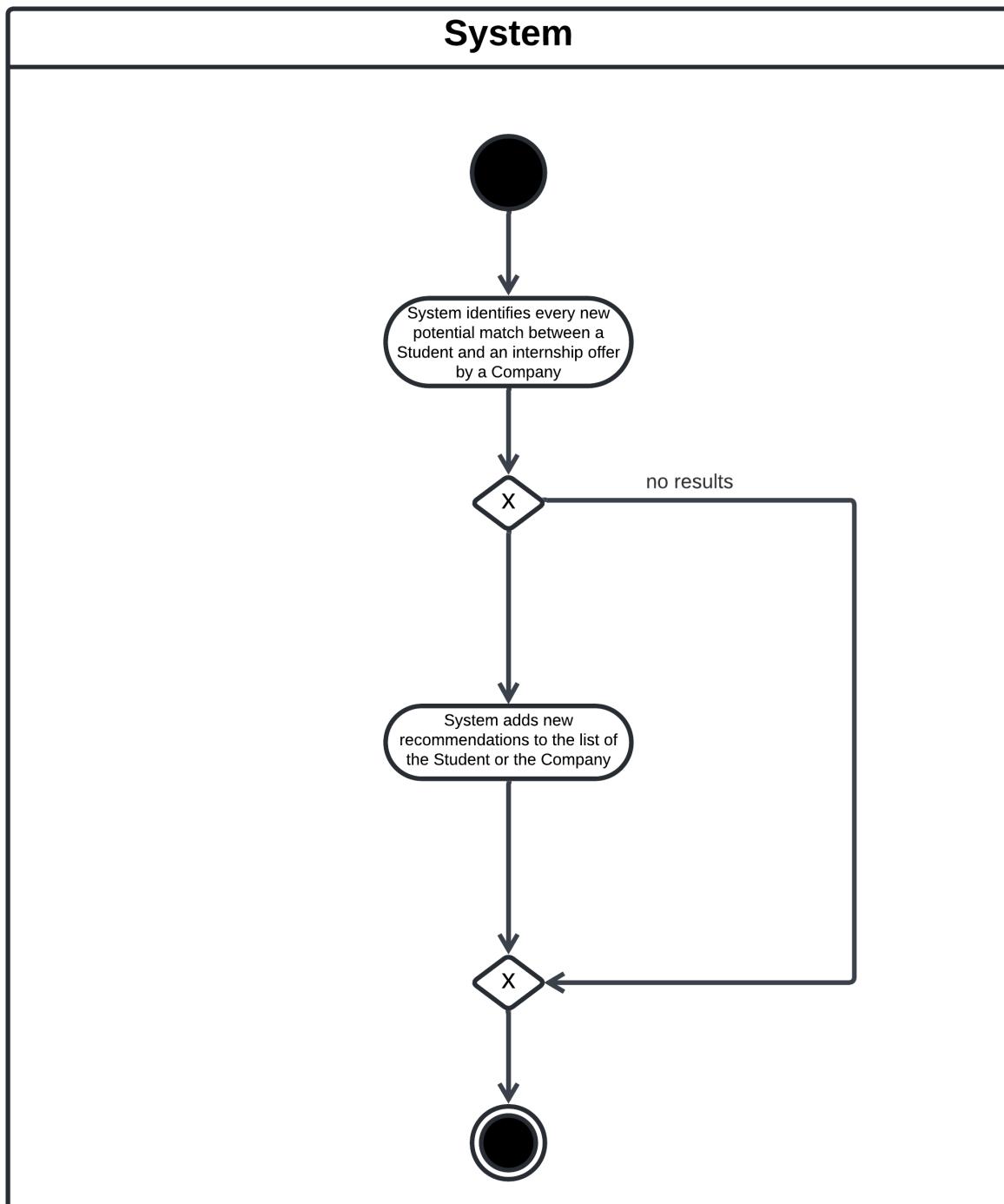


Figure 3.12: Generate Recommendations

UC11. Manage Internship Recommendations

Actor	Student
Entry Conditions	The Student is logged in and has received at least one recommendation about an internship offer.
Flow of Events	<ol style="list-style-type: none"> 1. In the dashboard, the Student navigates to the "Manage Recommendations" section. 2. The system displays a list of recommended internship offers for the Student, along with their and the offering Company's information. 3. The Student reviews the list of recommended internship offers, evaluating them. 4. The Student selects a specific recommendation, which expands showing the internship offer details. 5. The student can perform one of the following actions: accept, reject, or postpone the decision. <ul style="list-style-type: none"> 5.1. If the Student accepts the recommendation, the system automatically performs an "Apply" operation via the <u>UC. Apply to an Internship Offer</u> to the corresponding internship offer; the system also removes the recommendation from visualization. 5.2. If the Student rejects the recommendation, the latter is simply removed from visualization by the system. 5.3. If the Student chooses to postpone the decision, the system doesn't perform any operation. 6. The Student may repeat this process until there are no pending recommendations.
Exit Conditions	The decision on the selected recommendations is recorded, and the internship offers' statuses are updated accordingly in the system.
Exceptions	None

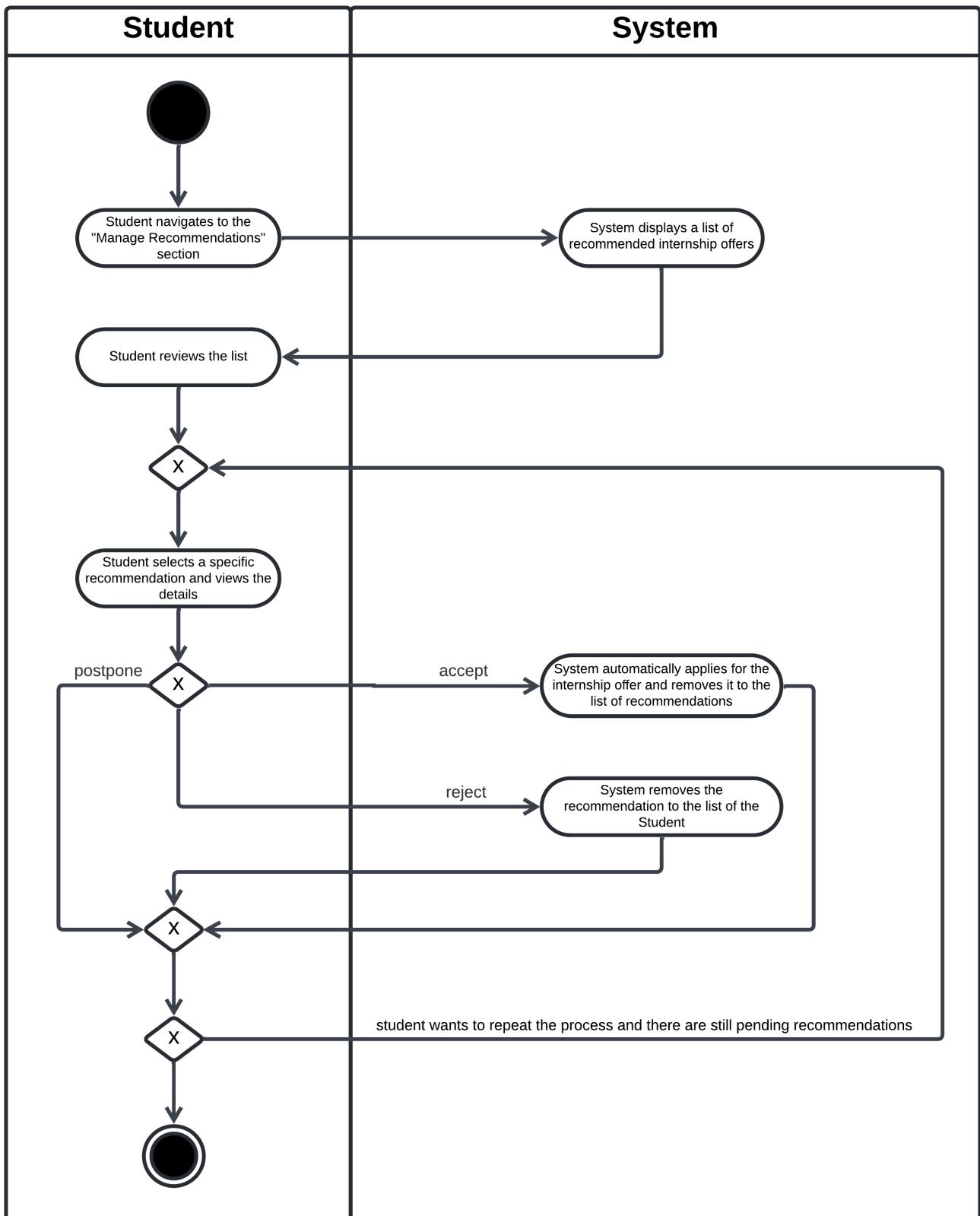


Figure 3.13: Manage Internship Recommendations

UC12. Manage Students Recommendations

Actor	Company
Entry Conditions	The Company is logged in and has received at least one recommendation about a Student for any of their internship offers.
Flow of Events	<ol style="list-style-type: none"> 1. In the dashboard, the Company navigates to the "Manage Recommendations" section. 2. The system displays a list of recommended students for the available internship offers, along with their profile information. 3. The Company reviews the list of recommended students, evaluating them. 4. The Company selects a specific recommendation, which expands showing the student's profile. 5. The Company can perform one of the following actions: accept, reject, or postpone the decision. <ul style="list-style-type: none"> 5.1. If the Company accepts the recommendation, the system marks it as "Selected" and the corresponding Student is notified through another recommendation, on which they will be able to decide; the system also removes the recommendation from visualization. 5.2. If the Company rejects the recommendation, the latter is simply removed from visualization by the system. 5.3. If the Company chooses to postpone the decision, the system doesn't perform any operation. 6. The Company may repeat this process until there are no pending recommendations.
Exit Conditions	The decision on the selected recommendations is recorded, and the students' statuses are updated accordingly in the system.
Exceptions	None

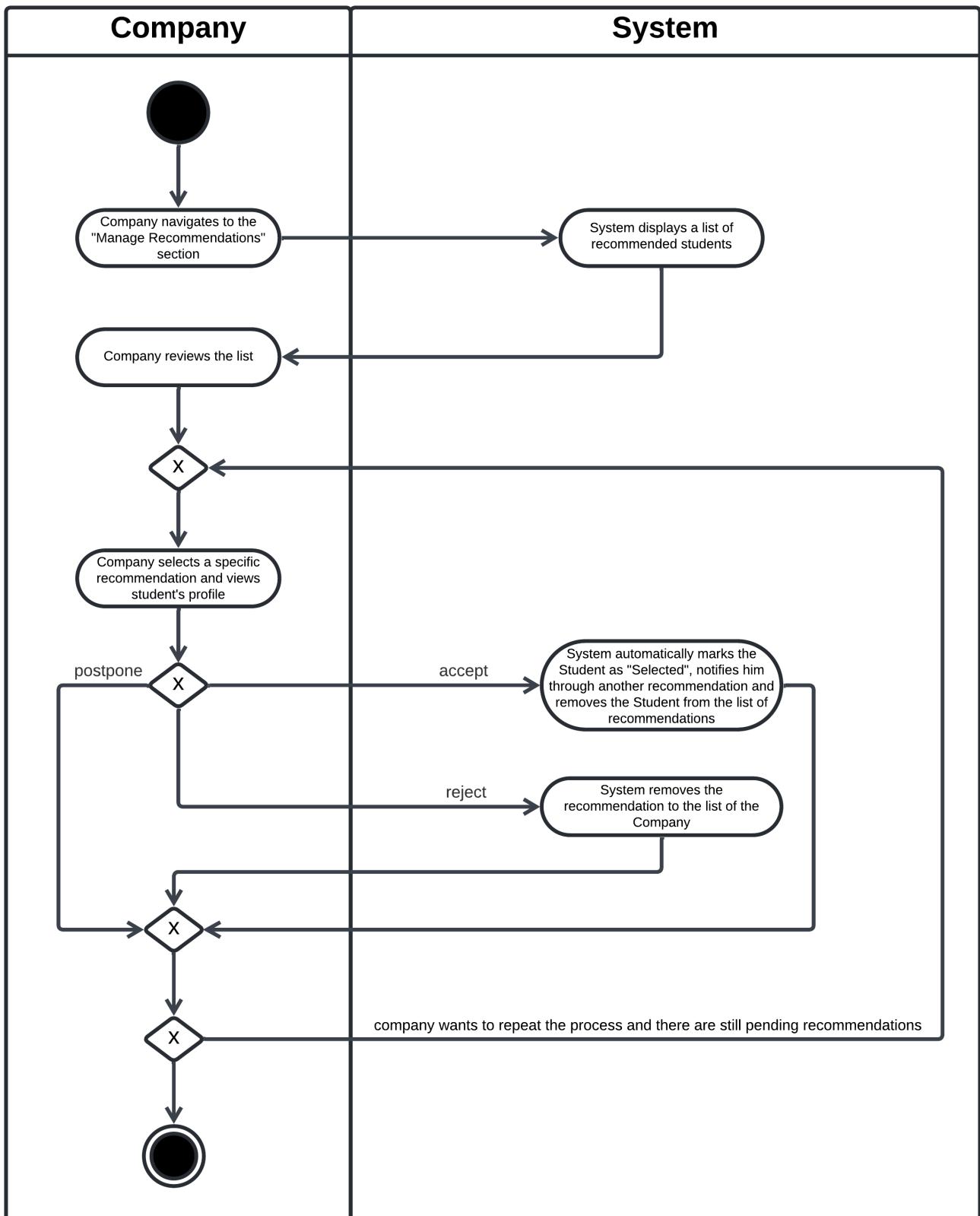


Figure 3.14: Manage Students Recommendations

UC13. Manage an Interview

Actor	Company, Student
Entry Conditions	The application deadline for a specific internship offer has expired and the Company has accepted at least one application to that offer, establishing a contact with the accepted Student.
Flow of Events	<ol style="list-style-type: none"> 1. From the dashboard page, the Company navigates to the page of one of its internship offers. 2. On the internship offer page, the Company selects a Student from the list of candidates for that offer, opening a new page to arrange the interview. 3. The Company creates an interview invitation to the Student, specifying the date, time, format (in-platform or in-person) and an optional text description with more details. 4. The Company reviews and forwards the interview invitation by confirming its information through the "Confirm" button. 5. The Student receives the invitation to the interview on the notifications section of its dashboard page and navigates to the platform to review the invitation details. 6. The Student accepts or declines the interview invitation, confirming their preference. If declining, he is able to provide a reason behind the choice, so that the Company may reschedule the interview. <ul style="list-style-type: none"> 6.1. If the Student accepts the invitation, the interview is conducted according to the specified details, enabling the Company to assess the student's suitability for the internship either by sharing multiple questions through the platform or by interviewing the candidate in person and later recording the results on the system. 6.2. If the Student declines the invitation, the Company receives a notification, and the system marks the interview as declined. The Company can decide whether to reschedule based on the given reason. 7. After the interview, the Company provides an evaluation of the student's performance, inserting strengths, weaknesses, and suitability for the role, and submits the results through the platform. 8. The system records the evaluation and updates the student's status in the selection process: "Interview Completed", "Selected" or "Rejected".

Exit Conditions	The interview is completed, and the Student's status in the selection process is updated in the system, becoming available for consulting.
Exceptions	None

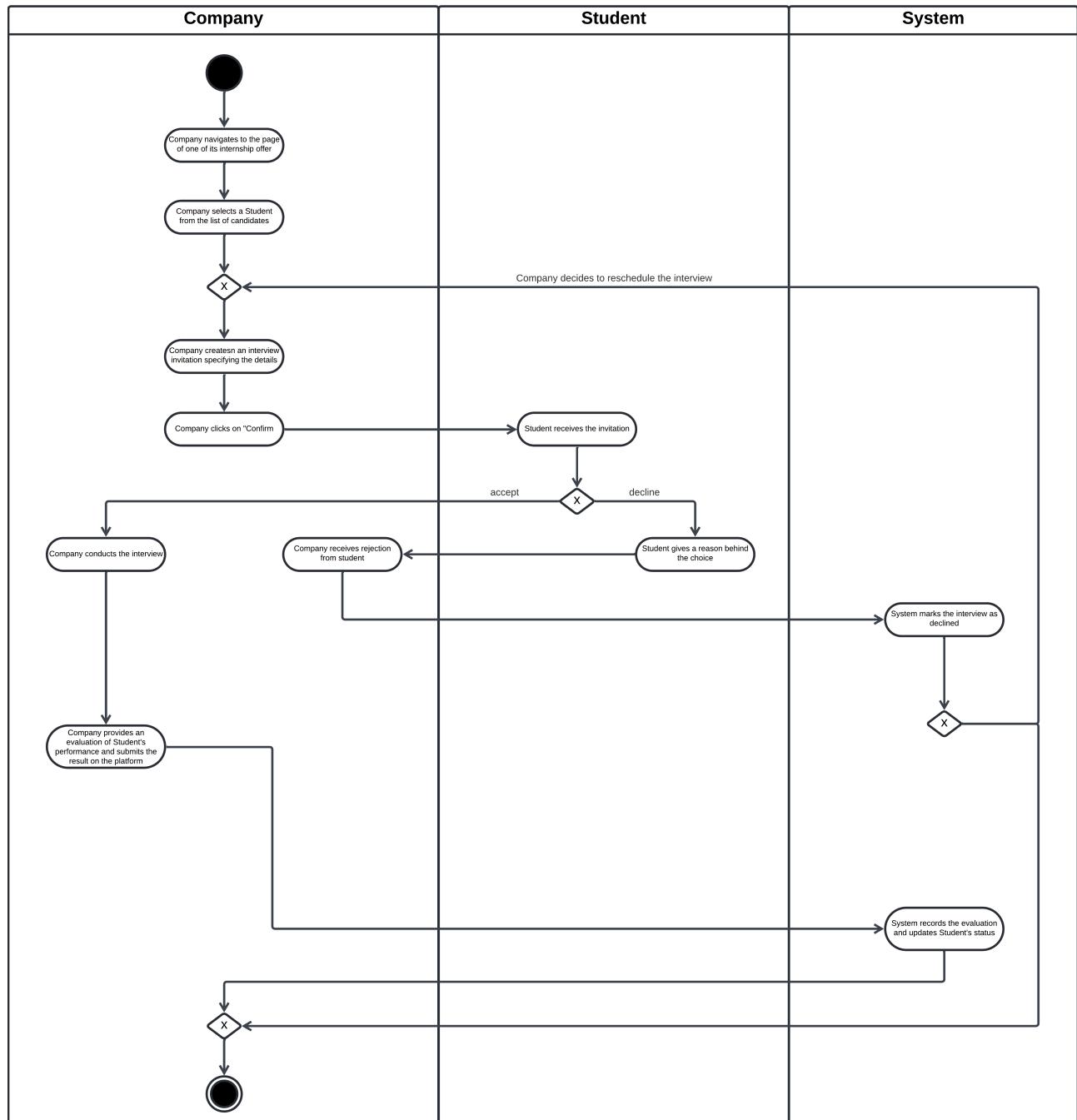


Figure 3.15: Manage an Interview

UC14. Provide Information for an ongoing Internship

Actor	Party (Student or Company)
Entry Conditions	The Party is logged into the S&C platform, is actively involved in the selected ongoing internship and has identified new information about it to be provided.
Flow of Events	<ol style="list-style-type: none"> 1. On the ongoing internship's page, the Party clicks the "Post information" button. 2. The Party provides new information about the selected ongoing internship in the given text field. 3. The Party confirms the information provided by clicking the confirmation button. 4. The system records the new information inserted and posts it on the ongoing internship's page.
Exit Conditions	The ongoing internship's page is updated and those changes are recorded in the system.
Exceptions	<ul style="list-style-type: none"> • The Party closes the page without completing the process: the system doesn't record any information and displays the Party's dashboard.

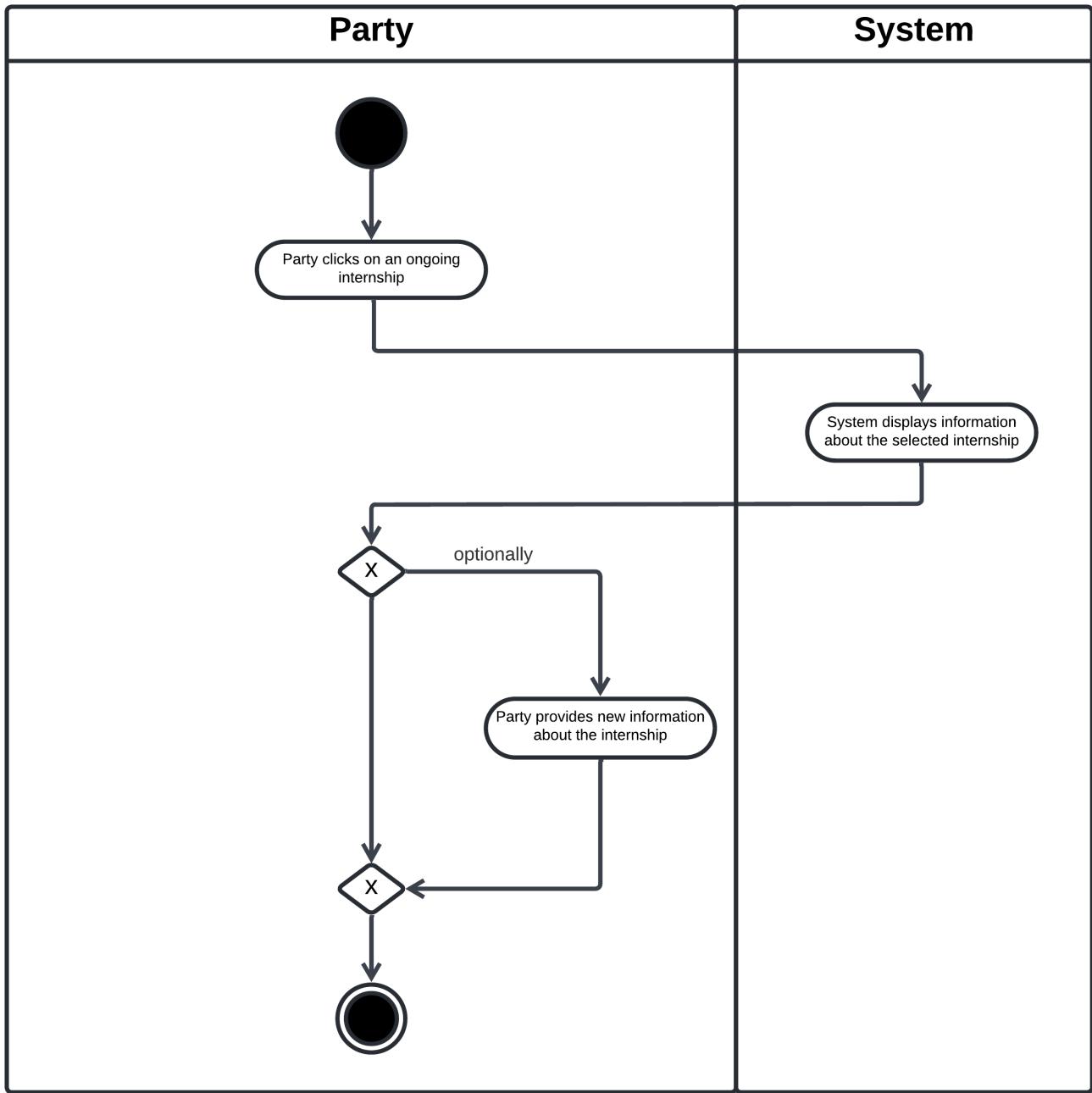


Figure 3.16: Provide Information for an ongoing Internship

UC15. Monitor an ongoing Internship

Actor	Party
Entry Conditions	The Party is logged into the S&C platform and is actively involved in at least an ongoing internship.
Flow of Events	<ol style="list-style-type: none">1. In the dashboard, the Party clicks on an ongoing internship offer in which it is actively participating, entering that internship's page.2. The system displays all the information about the selected internship.3. Optionally, the Party provides new information about the internship offer via the <u>UC. Provide Information for an ongoing Internship</u> functionality.
Exit Conditions	The system displays all the information about the selected ongoing internship.
Exceptions	None

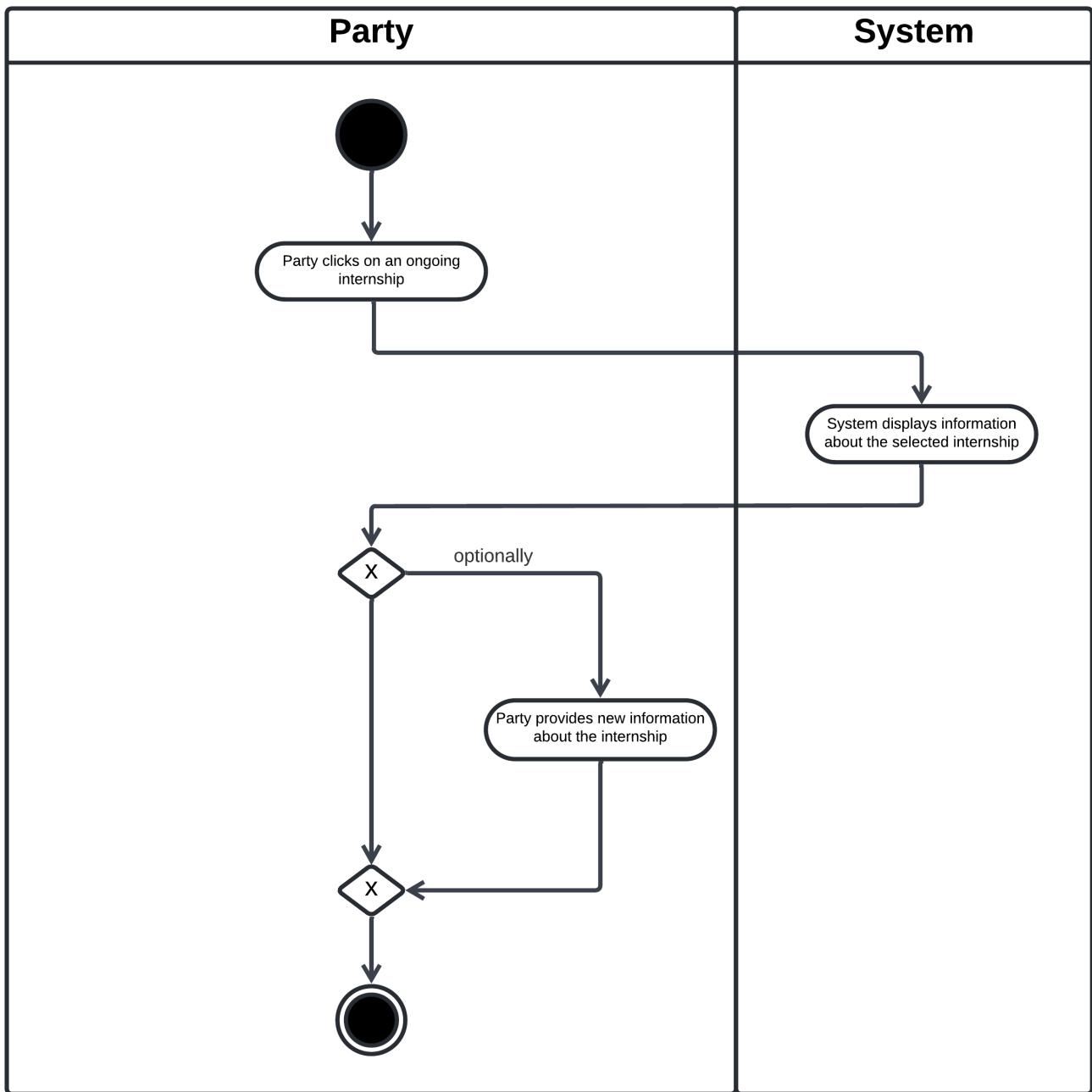


Figure 3.17: Monitor an ongoing Internship

UC16. Report Problems during an Internship

Actor	Party (Student or Company)
Entry Conditions	The Party is logged into the S&C platform, is actively involved in an internship and has identified an issue requiring intervention.
Flow of Events	<ol style="list-style-type: none"> 1. In the dashboard, the Party navigates to the "Report Problems" section. 2. The Party provides a detailed description of the issue, providing: <ul style="list-style-type: none"> • Nature of the problem. • Relevant details about when and how the issue occurred. • Optionally, attachments as images or documents to further describe the problem. 3. The Party submits the problem report by clicking the "Report" button. 4. The system forwards the Party's report to the Student's University.
Exit Conditions	The issue reported by the Party is registered in the system and made available to the University.
Exceptions	<ul style="list-style-type: none"> • Some mandatory fields are missing: the system doesn't allow the Party to complete the procedure until all the mandatory fields are filled out.

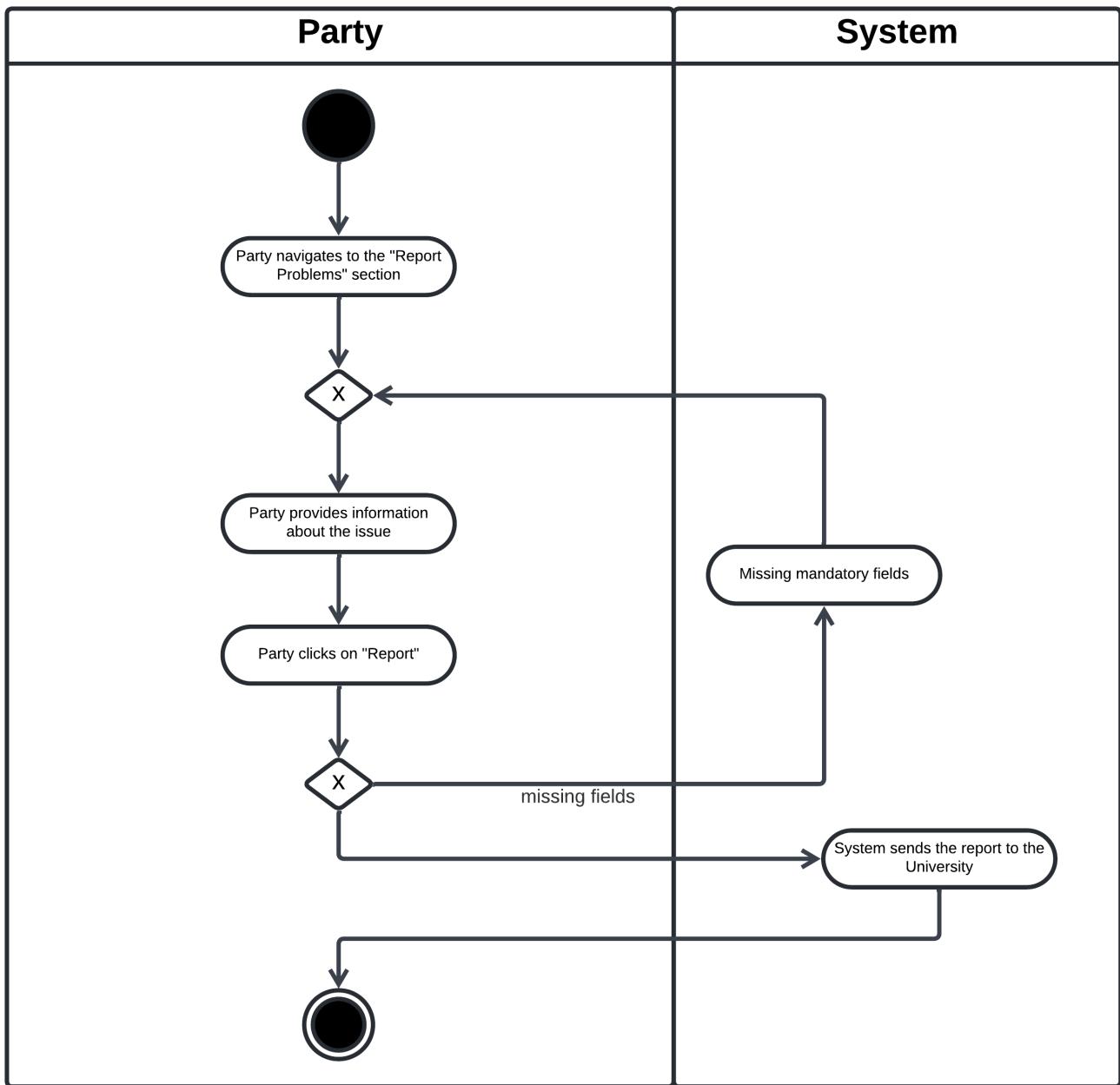


Figure 3.18: Report Problems during an Internship

UC17. Handle Problems during an Internship

Actor	University
Entry Conditions	The University is logged into the S&C platform and one of the Parties involved in an internship concerning one of their students has reported a problem.
Flow of Events	<ol style="list-style-type: none"> 1. In the dashboard, the University navigates to the "Handle Problems" section. 2. The University selects a specific problem report and reviews the details to understand the problem. 3. The University communicates outside the platform with the involved Parties to gather additional information and work collaboratively to resolve the issue. 4. Based on the outcome, the University updates the status of the reported problem in the system.
Exit Conditions	The issue previously reported by the Party is formally addressed by the University and the report is updated accordingly in the system.
Exceptions	None

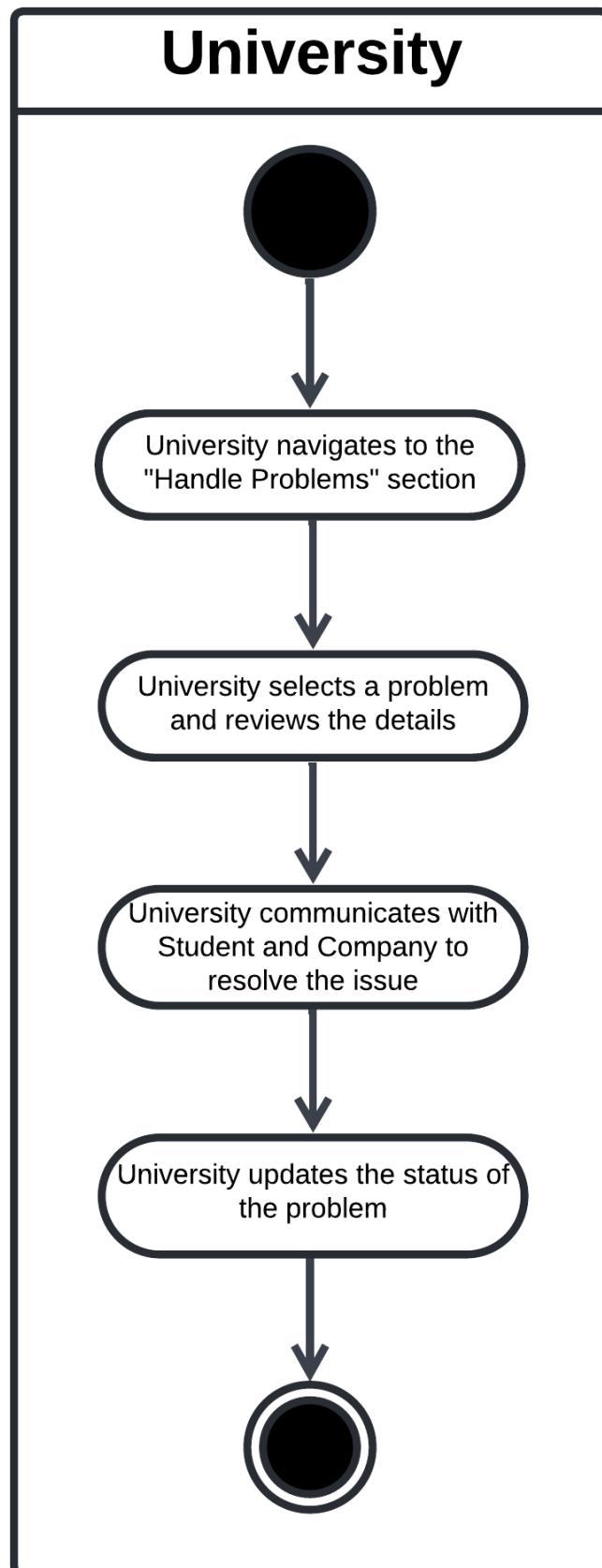


Figure 3.19: Handle Problems during an Internship

UC18. Report Feedback after an Internship

Actor	Party (Student or Company)
Entry Conditions	The Party is logged into the S&C platform and they have been actively involved in a finished internship.
Flow of Events	<ol style="list-style-type: none"> 1. A non-mandatory survey appears in the Party's dashboard, asking to provide detailed information about the internship to feed the statistical analysis and improve the recommendation algorithm. 2. The Party fills out the survey replying to each question. 3. The Party submits the report by clicking the "Submit" button.
Exit Conditions	The feedback provided by the Party is processed for improving the recommendation algorithm and recorded in the system.
Exceptions	<ul style="list-style-type: none"> • The Party closes the survey without completing it: the system doesn't record any information and displays the Party's dashboard.

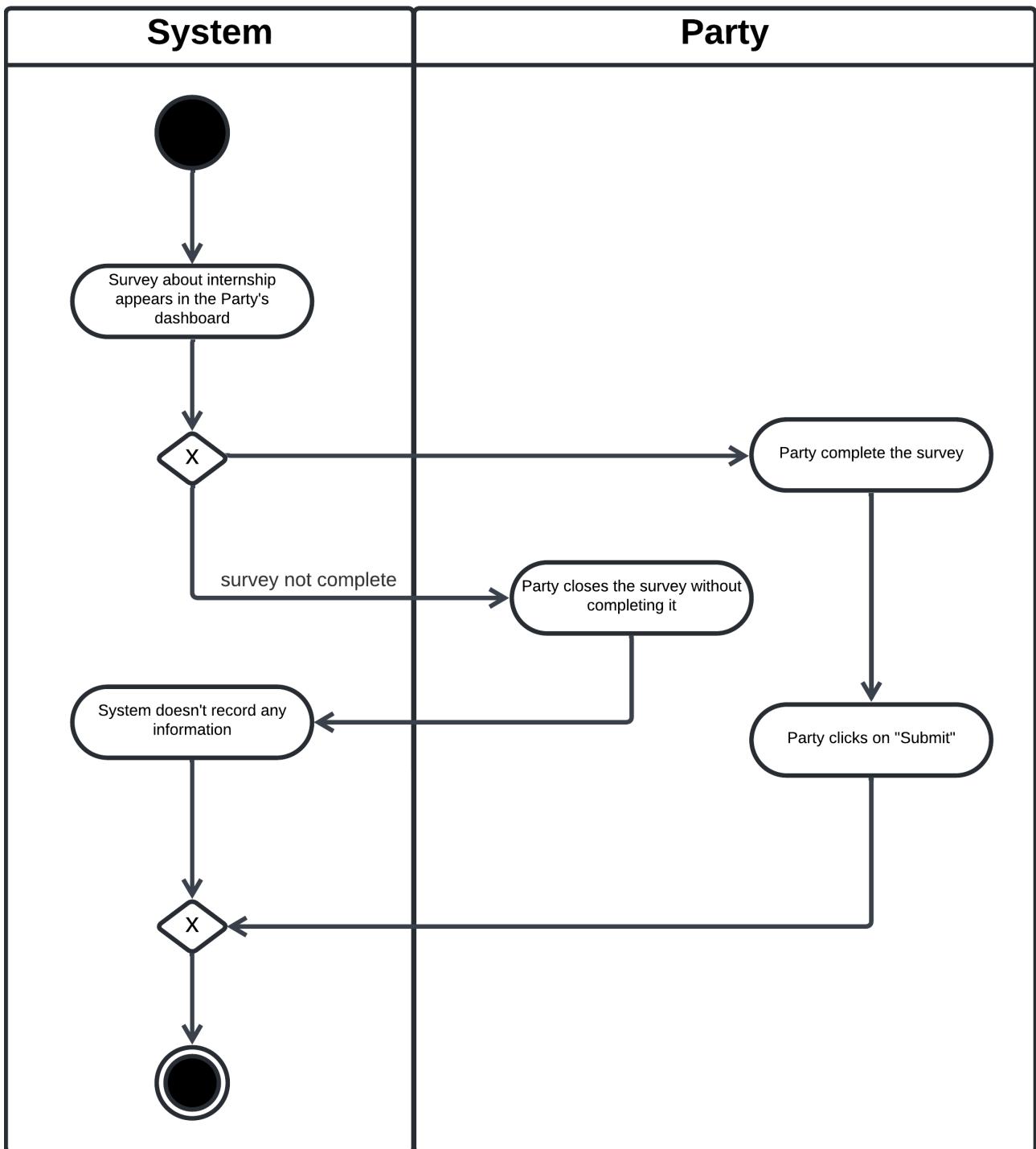


Figure 3.20: Report Feedback after an Internship

UC19. Suggest Optimizations for a Student Profile

Actor	Student
Entry Conditions	The Student is logged into the S&C platform.
Flow of Events	<p>1. In their "Profile" section, the Student clicks the "Improve Profile" button.</p> <p>2. The system automatically analyzes the student's profile, reviewing the following details:</p> <ul style="list-style-type: none"> • Academic background • Skills • Uploaded CV • Certifications and extracurricular activities <p>3. Based on the analysis, the system generates a list of personalized suggestions to improve the profile which can belong to one of the following categories:</p> <ul style="list-style-type: none"> • Add additional skills or certifications. • Update academic details or achievements. • Include or enhance descriptions of projects. • Enhance the overall style by improving clarity or content. <p>4. The Student reviews the suggestions and eventually decides which one to implement via the UC. Update User Profile functionality.</p>
Exit Conditions	The Student's profile is possibly optimized, improving its appeal and relevance for obtaining more internship offers in the future.
Exceptions	<ul style="list-style-type: none"> • No optimizations can be found: the system doesn't provide any suggestions and terminates silently.

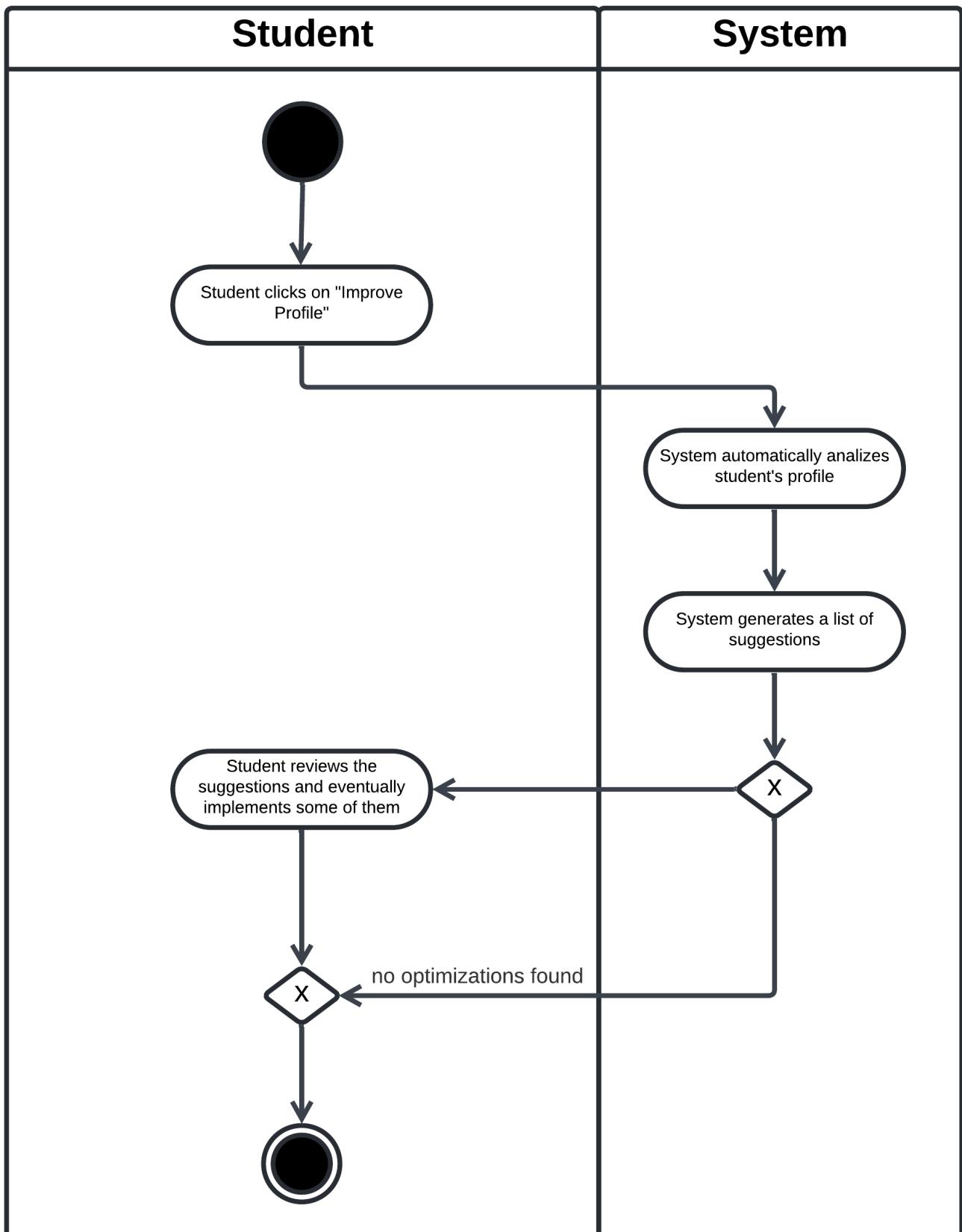


Figure 3.21: Suggest Optimizations for a Student Profile

UC20. Suggest Optimizations for an Internship Offer

Actor	Company
Entry Conditions	The Company is logged into the S&C platform.
Flow of Events	<p>1. In an internship offer's page, the Company clicks the "Improve Offer" button.</p> <p>2. The system automatically analyzes the selected internship offer, reviewing the following details:</p> <ul style="list-style-type: none"> • Description of the internship • Application domain • Required skills <p>3. Based on the analysis, the system generates a list of suggestions to improve the offer:</p> <ul style="list-style-type: none"> • Refine the description to better specify tasks and requirements. • Improve the application domain by incorporating additional information or further elaborating on the details already provided." • Introduce or modify the set of required skills. • Enhance the overall style by improving clarity or content. <p>4. The Company reviews the suggestions and eventually decides which one to implement via the <u>UC. Update Internship Offer</u> functionality.</p>
Exit Conditions	The Company's selected internship offer is possibly optimized, making it more attractive to students and improving internship visibility for the future.
Exceptions	<ul style="list-style-type: none"> • No optimizations can be found: the system doesn't provide any suggestions and terminates silently.

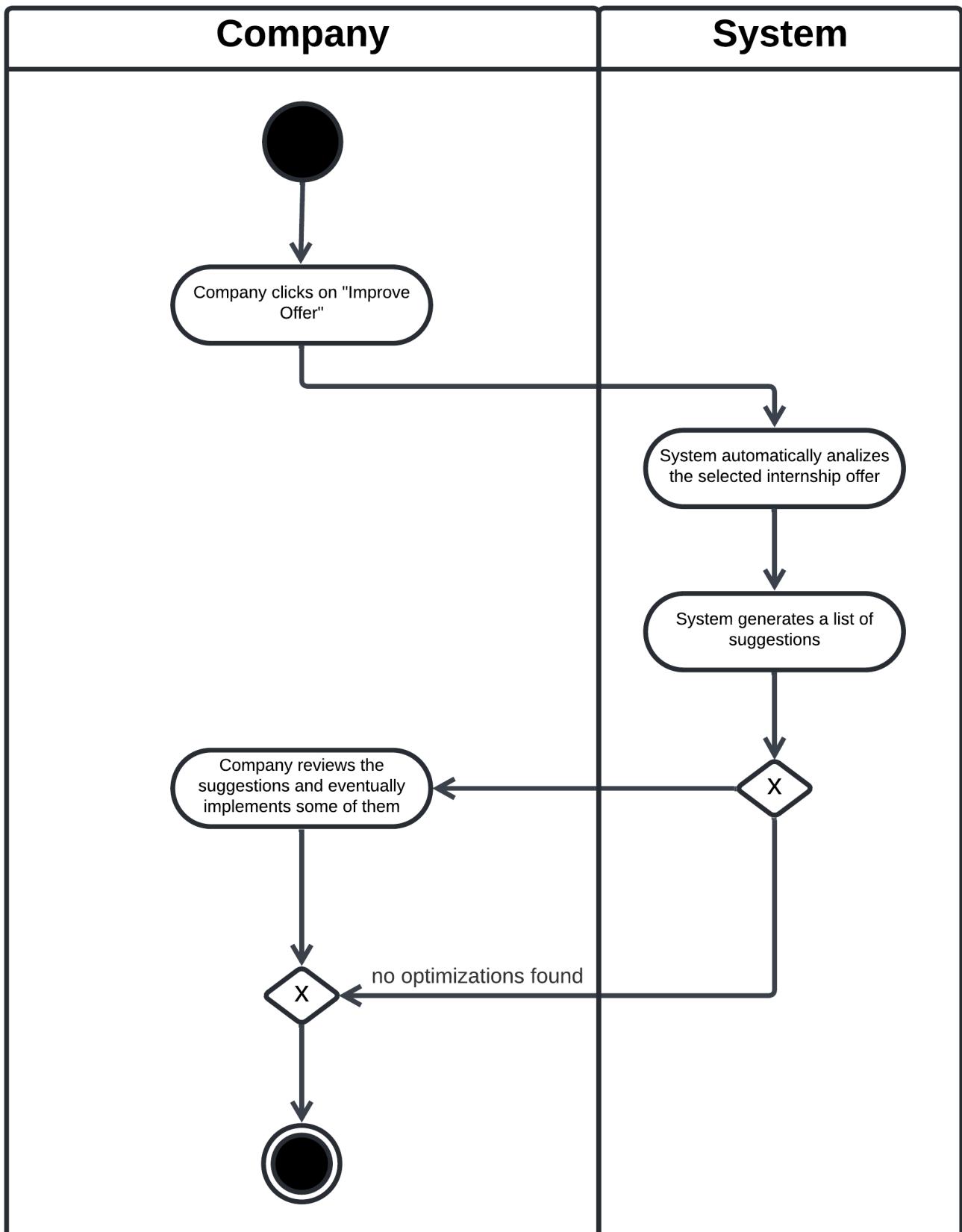


Figure 3.22: Suggest Optimizations for an Internship Offer

3.2.2. Mapping On Requirements

Row ID	Goal ID	Req ID	DA ID	UC ID
1	G1	R1, R2, R17	DA1 to DA6	UC1
2	G1	R1, R2	DA1 to DA6	UC2
3	G1	R1, R2	DA1 to DA6	UC3
4	G1 to G12	R2	DA1, DA7	UC4
5	G1	R3, R18	DA1 to DA6	UC5
6	G2	R4, R5, R16	DA1, DA7	UC6
7	G2	R2, R6, R7, R14, R15, R20, R21	DA1, DA7	UC7
8	G3	R2, R8, R9	DA1, DA8	UC8
9	G3	R2, R10, R11, R12, R13	DA1, DA8	UC9
10	G4, G5	R2, R16, R17, R18, R19, R20, R24, R38	DA1	UC10
11	G4	R2, R10, R11, R12, R13, R22, R23	DA1	UC11
12	G5	R2, R16, R22, R24	DA1	UC12
13	G6	R2, R25, R26, R27, R28, R29, R30, R31	DA1, DA9, DA10	UC13
14	G7	R2, R32	DA1, DA11	UC14
15	G7	R2, R33	DA1, DA11	UC15
16	G8	R2, R34, R35	DA1, DA12	UC16
17	G9	R2, R36	DA1, DA13	UC17
18	G10	R2, R37, R38	DA1, DA14	UC18
19	G11	R2, R39	DA1	UC19
20	G12	R2, R40	DA1	UC20

3.3. Performance Requirements

As the system does not supply any particular critical functionality to its users, it is acceptable if there are relatively small delays in the system's response times: therefore, constraints about the system's performance are in principle very lax/loose. On the other hand, requirements for the quality of the system are instead tight:

- Anytime, the system shall be able to handle a load of up to 10000 concurrent Users without significant performance degradation: however, it shall also be scalable enough

for this amount to be increased as a consequence of business decisions which cause the system's scope to widen.

- When generating recommendations, both accuracy and F1 score of the mechanisms employed for identifying those recommendations shall be at least 0.8.
- Whenever receiving a request, the system shall respond to the requesting User in less than 5 seconds. This doesn't include recommendations generation, which, however, should be able to be carried out for up to 1.000.000.000 different users.
- Whenever any kind of information (including new recommendations, interview invites, or complaints) needs to be available to a User, the system shall "append" it to that User's profile in the appropriate section within the following 0.01 seconds, so that it can immediately be shown to that User if they are logged in.

3.4. Design Constraints

3.4.1. Standards Compliance

This system is designed to adhere to a range of standards and regulations to ensure compliance with legal, technical, and usability requirements. In the following, we outline the primary standards considered:

1. **General Data Protection Regulation (GDPR)** The platform ensures compliance with GDPR to protect users' personal data. Key measures include:
 - Data minimization and anonymization techniques for data storage.
 - Mechanisms for users to access, modify, and delete their data.
 - Transparent consent collection processes and data usage policies, ensuring that the user is aware of which operations and processes are applied to the shared data.

By integrating these security measures, the S&C platform aims to foster user trust and compliance with current regulations.

2. **World Wide Web Consortium (W3C)** The system adheres to W3C standards to ensure interoperability and usability across modern browsers. Some useful target objectives derived from the guidelines are:
 - Use meaningful HTML elements to structure content properly, enabling assistive technologies to interpret the page effectively.

- Follow WAI-ARIA (Web Accessibility Initiative–Accessible Rich Internet Applications) standard, a framework for adding attributes to identify features for user interaction, how they relate to each other, and their current state
- Perform decisions in order to reduce load times and enhance responsiveness.

3. **Web Content Accessibility Guidelines (WCAG)** To ensure accessibility, the system follows the latest WCAG standards, a subset of W3C standard that outlines principles for making web content more accessible for people with disabilities:

- Support for assistive technologies: screen readers and magnifiers, speech recognition software, keyboard-only navigation. The majority of the accessibility tools are already provided by the OS; the main focus should be in ensuring that such technologies are fully supported and working while interacting with the system.
- Accessible navigation structures and compatibility across various devices and input methods.

3.4.2. Hardware Limitations

Considering the nature of the system, which is a WebApp, hardware limitations are minimal, as the system is designed to be lightweight and operates on standard devices and browsers, requiring only basic hardware specifications.

3.5. Software System Attributes

The main attributes that the system to be developed shall present are outlined here, listed in order of importance for its correct functioning.

3.5.1. Reliability

The system shall ensure high reliability: namely, it shall handle failures effectively and promptly recover from disruptions, minimizing total downtime, frequency of failures and number of functionalities involved. This shall be achieved by employing standard mechanisms for fault tolerance, such as replication of data storage or back-end computation nodes, ensuring no single point of failure compromises the system.

3.5.2. Availability

The system shall guarantee at least a 2-nines availability (99%, or a total downtime of at most 3.65 days/year), ensuring it is available except for planned maintenance, which should occur outside peak hours and be announced in advance anyway. The decision is based on the fact that it does not offer any critical service to users that cannot be postponed in time. A robust monitoring system shall track application health and trigger alerts for any performance issues or downtime.

3.5.3. Security

The system shall implement secure authentication and authorization mechanisms to verify user identities and restrict access to features based on user roles. Such guarantees may be implemented through strong password policies, 2FA, encrypted communication protocols (e.g., HTTPS), etc. Additionally, the platform must ensure data integrity by preventing unauthorized modifications and preserving the accuracy of stored information, both physically and virtually. To uphold confidentiality, the system shall encrypt sensitive data both in transit and at rest, ensuring that data remains unreadable even if intercepted or exposed. Secure data storage practices must be observed. Moreover, the system must be resilient to common attack vectors, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). The platform should also incorporate regular security patches to address newly discovered vulnerabilities.

3.5.4. Maintainability

System maintenance shall be planned for hardware or software upgrades, or for the deployment of production code containing additional functionalities, bug fixes or security updates. It should last 8 hours on average, theoretically enabling it to be carried out entirely outside peak traffic periods; in any case, it shall not exceed 24 hours. Maintenance bursts should happen exceptionally, possibly no more than twice or thrice a year, and shall always be announced in advance to the public (minimum 48 hours before). Software maintenance shall instead be employed right from the start of the development process by enforcing the application of best practices, in compliance with the industry standards; code shall be clean, modular, reusable, and low-coupled, to facilitate the introduction of new functionalities as needed.

3.5.5. Portability

Since the platform is offered as a WebApp, it operates through standard web browsers and inherently supports compatibility across various operating systems. Given this architecture, there is no need for native application installations, ensuring ease of use and minimizing setup complexity. Therefore, the chosen architecture and platform's reliance on web standards already solves all aspects related to portability. The architecture of the platform shall be designed to offer portability, ensuring that it can be deployed across various environments and, eventually, hosting services. By utilizing containerization technologies, the application shall be easily migrated regardless of the underlying infrastructure.

4 | Formal Analysis using Alloy

4.1. Objectives

The formal modeling activity is carried out using Alloy 6 and aims to validate and ensure the soundness of some of the system's core processes, with the following aspects as its primary objectives:

1. Consistent System Evolution

We have modeled the rules that define the system's dynamic evolution through time; then, we have defined the intended properties (preconditions and postconditions) of the set of actions, by enforcing role-specific constraints that differentiate their effects depending on who performs them; finally, we have verified that the evolving scenarios remain sound and correct. In particular, in our model we have focused on the following subset of all the possible actions:

- Sign Up of a generic User (which can be any of the expected actors: Student, Company or University)
- Post a new Internship Offer by a Company
- Generate a Recommendation
- Accept a Recommendation
- Reject a Recommendation
- Apply to an Internship Offer

2. Recommendation Handling Logic

We have defined when Recommendations are generated and how they are linked to Offers and Parties; then, we have modeled the long-lasting effects of actions involving Recommendations; finally, we have ensured state transitions of Recommendations are consistent and irreversible. In particular, every Recommendation generated by the system evolves from an initial *Unhandled* state, eventually becoming either *Accepted* or *Rejected* permanently.

These are the parts of the system that are, in our opinion, the most critical, as the most complex interactions happen between these functionalities. Given that they have already weighed down the model a lot in terms of the high number of variables and clauses generated during validation, we have decided to avoid modeling other less relevant aspects of the systems (e.g, interview handling), or the possibility of instances being deleted over time, as it would overcomplicate the model while we were trying to achieve a simple and easily understandable representation.

4.2. The Model

The Model has been structured modularly by separating signatures, actions, facts and the actual worlds in different files.

4.2.1. Signatures

This file contains the definitions of the Signatures of the main entities: a hierarchy of Users (Universities and Parties, with the latter further splitting into Students and Companies), and then Offers and Recommendations. Their fields have been placed by prioritizing *one* as the cardinality of the relation whenever possible, in order to avoid performance overhead and complexity caused by avoidable quantifiers. As an example, *var offeringCompany: one Company* in *sig Offer* has been preferred instead of defining a *var offers: set Offer* field inside *sig Company*. For every Signature, rules are specified as constraints to ensure that all the instances are always meaningful and consistent: further details about each constraint are commented out in the code.

```

module signatures

var abstract sig User {}

var abstract sig Party extends User {
    var receivedRecommendations: set Recommendation
} {
    // If a Party has more recommendations about another Party in the context of the same
    // Offer, then all, except one of them, must be REJECTED
    all disj r1, r2 : receivedRecommendations |
        (r1.recommendedParty = r2.recommendedParty and r1.contextOffer =
         r2.contextOffer) implies
            r1.status = REJECTED or r2.status = REJECTED
    // NOTE: this also prevents that more of such recommendations are sent to a Party
    // that already has an UNHANDLED or ACCEPTED recommendation about that other
    // Party
    // for the same Offer
}

var sig Student extends Party {
    var atUniversity: one University
} {

```

```

// Its University must never change (as we aren't modeling profile update operations)
atUniversity' = atUniversity
}

var sig Company extends Party {
} {
// A Company must have recommendations only in the context of its Offers
no r : receivedRecommendations | r.contextOffer not in offeringCompany.this
}
var sig University extends Party {}

var sig Offer {
// offeringCompany, as a relation , must be variable because new Offers will be added to
// the set of Offers, along with their offering Companies
var offeringCompany: one Company,
var applicatingStudents: set Student
} {
// However, for a specific Offer, its offeringCompany must never change
offeringCompany' = offeringCompany
}

var sig Recommendation {
// contextOffer and recommendedParty, as relations , must be variable because new
// Recommendations will be added to the set of Recoomendations, along with their
// Offers and Parties to be recommended
var contextOffer: one Offer,
var recommendedParty: one Party,
var status: one RecommendationStatus
} {
// However, for a specific Recommendation:
// its contextOffer must never change
contextOffer' = contextOffer
// its recommendedParty must never change
recommendedParty' = recommendedParty
// when not UNHANDLED, its status must never change
(not status = UNHANDLED) implies status' = status
// NOTE: together with the facts that all new Recommendations must have been
//generated by using the appropriate predicate and that same predicate sets the
// status to UNHANDLED in the creation instant, this implies that every new

```

```

// Recommendation starts with UNHANDLED, and thus it is not necessary to
// specify that the status must have been UNHANDLED once

// Exactly one Party must have this as a Recommendation
one (receivedRecommendations).this

// Students must not be recommended to other Students
recommendedParty in Student iff this not in Student.receivedRecommendations
// Offers (and thus Companies) must not be recommended to other Companies
recommendedParty in Company iff this not in Company.receivedRecommendations
//           (one of these above is redundant)

// Companies recommendations must match with the contextOffer's publishing Company
recommendedParty in Company iff recommendedParty = contextOffer.offeringCompany
}

enum RecommendationStatus { ACCEPTED, REJECTED, UNHANDLED }

```

4.2.2. Actions

This file contains the modeling of the chosen actions, which have been defined as predicates that, in every time instant, are true if executed and false otherwise. For each action, preconditions that need to hold for it to be executed are conjoined with its postconditions, which are the effects of that specific action to be visible in the next state of the system. Some actions, as has been explained previously in the Use Cases, "call" other ones during their execution: this is the case, for example, for a Student accepting a Recommendation (which means that the system shall automatically make them apply to the corresponding Offer), or for the generation of Recommendations (which must be done by the system as a consequence of other actions happening).

```

module actions

open signatures

pred doNothing [] {}

// Predicate that models the preconditions and postconditions of a signUp action
pred signUp [] {
    some newUser : User' - User | newUser in User'
}
```

```

and
all newUser : User' - User | (
(
    some o : Offer |
        (generateRecommendation[o, o.offeringCompany, newUser] or
         generateRecommendation[o, newUser, o.offeringCompany])
)
implies
newUser in Student' - Student
)
}

// Predicate that models the preconditions and postconditions of a postNewOffer action
pred postNewOffer [c : Company] {
    some newOffer : Offer' - Offer | newOffer in Offer'
    and
    all newOffer : Offer' - Offer | (
        no newOffer.(applicatingStudents') and
        (
            doNothing or
            some s : Student |
                (generateRecommendation[newOffer, c, s] or
                 generateRecommendation[newOffer, s, c])
        )
    )
}

// Predicate that models the preconditions and postconditions of a generateRecommendation action
pred generateRecommendation [o : Offer, recommended : Party, to : Party] {
    some newRecommendation : Recommendation' - Recommendation | (
        newRecommendation in Recommendation'
        and
        newRecommendation in (to.receivedRecommendations),
        and
        newRecommendation.contextOffer' = o
        and
        newRecommendation.recommendedParty' = recommended
    )
}

```

```

and
all newRecommendation : Recommendation' - Recommendation | (
    newRecommendation.status' = UNHANDLED
)
}

// Predicate that models the preconditions and postconditions of an acceptRecommendation
// action
pred acceptRecommendation[r : Recommendation] {
    r.status = UNHANDLED and (r.status)' = ACCEPTED
    and
    (
        receivedRecommendations.r in Student implies (
            (
                receivedRecommendations.r in r.contextOffer.applicatingStudents
                or
                applyToOffer[receivedRecommendations.r, r.contextOffer]
            )
        )
    )
    and
    (
        receivedRecommendations.r in Company implies (
            (some r1 : Recommendation | r1.contextOffer = r.contextOffer and r1 in
             r.recommendedParty.receivedRecommendations)
            or
            generateRecommendation[r.contextOffer, r.contextOffer.offeringCompany,
             r.recommendedParty]
        )
    )
}
}

// Predicate that models the preconditions and postconditions of a rejectRecommendation
// action
pred rejectRecommendation[r : Recommendation] {
    r.status = UNHANDLED and (r.status)' = REJECTED
}

// Predicate that models the preconditions and postconditions of an applyToOffer action

```

```

pred applyToOffer [s : Student, o : Offer] {
    (s not in o.applicatingStudents)
    and
    (s in (o.applicatingStudents)’)
    and
    all r : (s.receivedRecommendations :> contextOffer.o) |
        r.status = UNHANDLED implies (r.status)’ = ACCEPTED
}

```

4.2.3. Facts

This file contains facts for enforcing the consistency of the world, which mostly ensure that the system evolution only happens throughout the action defined in the previous file, and not spontaneously. Signature-specific constraints are not specified here, as they have already been defined in the *Signatures.als* file.

```

module facts

open signatures
open actions

// Set of Users never loses elements: it either stays the same or some sign up has happened
fact consistentUsersSet {
    always (
        (User = User’)
        or
        (User in User’ and signUp[])
    )
}

// Set of Offers never loses elements: it either stays the same or some company has published a new Offer
fact consistentOffersSet {
    always (
        (Offer = Offer’)
        or
        (Offer in Offer’
        and
        all o : Offer’ - Offer | postNewOffer[o.offeringCompany’])
    )
}

```

```

        )
}

// Set of Recommendations never loses elements: it either stays the same or some
Recommendation has been
// generated as a consequence of some other action

fact consistentRecommendationsSet {
    always (
        (Recommendation = Recommendation')
        and
        receivedRecommendations = receivedRecommendations')
        or
        (Recommendation in Recommendation'
        and
        receivedRecommendations in receivedRecommendations'
        and
all r : Recommendation' - Recommendation | (
            generateRecommendation[r.contextOffer', r.recommendedParty',
            receivedRecommendations'.r]
        )
    )
}
}

// A Recommendation can be generated only by specific actions
fact doNotGenerateRecommendationsSpontaneously {
    always (
        all r : Recommendation' |
        generateRecommendation[r.contextOffer', r.recommendedParty',
        receivedRecommendations'.r]
        implies
        (
            (receivedRecommendations)'.r in Party' - Party // a signUp[] has happened
            or
            (r.contextOffer)' in Offer' - Offer // a postNewOffer[...] has happened
            or
            one studRec : Company.receivedRecommendations | ( // an
                acceptRecommendation[...] has happened
                studRec.contextOffer = (r.contextOffer)' and

```

```

        studRec.recommendedParty = (receivedRecommendations)' .r and
        acceptRecommendation[studRec]
    )
)
)
}
}

// Relation of applications to Offers never loses elements: it either stays the same or
some Student has applied to some Offer

fact consistentApplicationsRelation {
    always (
        (applicatingStudents = applicatingStudents')
        or
        (applicatingStudents in applicatingStudents'
        and
        some s : Student, o : Offer | applyToOffer[s, o])
    )
}
}

// A change of status in a Recommendation can happen only if it has been accepted or rejected

fact doNotHandleRecommendationsSpontaneously {
    always (
        all r : Recommendation | (
            r.status = ACCEPTED implies once acceptRecommendation[r]
            and
            r.status = REJECTED implies once rejectRecommendation[r]
        )
    )
}
}

```

4.2.4. Main

This file contains two worlds: a smaller, somewhat "predetermined" dynamic world, accurately devised to show all the actions and their effects, and a larger static world, casually generated to show that the system is consistent also for a large number of instances. It also contains a predicate modeling the system's standard behavior, which is not really useful as the analyzer tends to always execute the *doNothing* predicate.

open signatures

```
open actions
open facts

pred initialize [] {
    #Student = 0
    #Company = 0
    #University = 0
    #Offer = 0
    #Recommendation = 0
}

pred System {
    initialize
    and
    always(
        (
            doNothing
            or
            signUp[]
            or
            (some c : Company | postNewOffer[c])
            or
            (some s : Student, o : Offer | applyToOffer[s, o])
            or
            (some r : Recommendation | acceptRecommendation[r])
            or
            (some r : Recommendation | rejectRecommendation[r])
        )
    )
}

pred largeStaticWorld {
    #Student = 6
    #Company = 4
    #University = 2
    #Offer = 5
    #Recommendation = 5
    #status.ACCEPTED = 2
    #status.REJECTED = 1
}
```

```

#applicatingStudents = 6
}

run largeStaticWorld for 24 but 1..1 steps

pred exampleDynamicWorld {
    // World initialization , and some new Users sign up
    initialize and signUp[];

    // Two distinct Companies post some new Offers and at least one new Recommendation is
    generated
    #Student = 1 and #Company = 2 and #University = 1 and
    some disj c1, c2 : Company | postNewOffer[c1] and postNewOffer[c2] and
    some o : Offer', s : Student, c : Company | generateRecommendation[o, s,
    c];

    // Some new Users sign up; a Company accepts a Recommendation, causing another one to be
    sent to the
    // recommended Student
    signUp[] and #Company = 3 and #Offer = 2 and #Recommendation < 4 and
    some s : Student, r : Company.receivedRecommendations |
    (no s.receivedRecommendations and no applicatingStudents.s and
    r.recommendedParty = s and acceptRecommendation[r]);

    // A Student applies to an Offer spontaneously, without accepting a Recommendation;
    another one accepts
    // a Recommendation, so that they also apply to an Offer via that operation ; a
    Recommendation gets
    // rejected ; another Company without Offers posts an Offer
    some disj s1, s2 : Student | (
        (some o : Offer | applyToOffer[s1, o]) and
        (some r : s2.receivedRecommendations | acceptRecommendation[r])
    ) and
    some r : Recommendation | rejectRecommendation[r]
    and
    some c : Company | no offeringCompany.c and postNewOffer[c];

    // Bound on the maximum # of instances (to avoid cluttered worlds, which would prevent
    the understanding
}

```

```
// of what is happening)
#User = 6 and #Offer = 3 and #Recommendation < 5
}

run exampleDynamicWorld for 10 but 5..6 steps
```

4.3. Validation through the Model

This section shows a possible instance for each of both static and dynamic scenarios according to the defined signatures, actions and facts, obtained by running the predicated in the *main.als* file.

4.3.1. Instance of the Large Static World

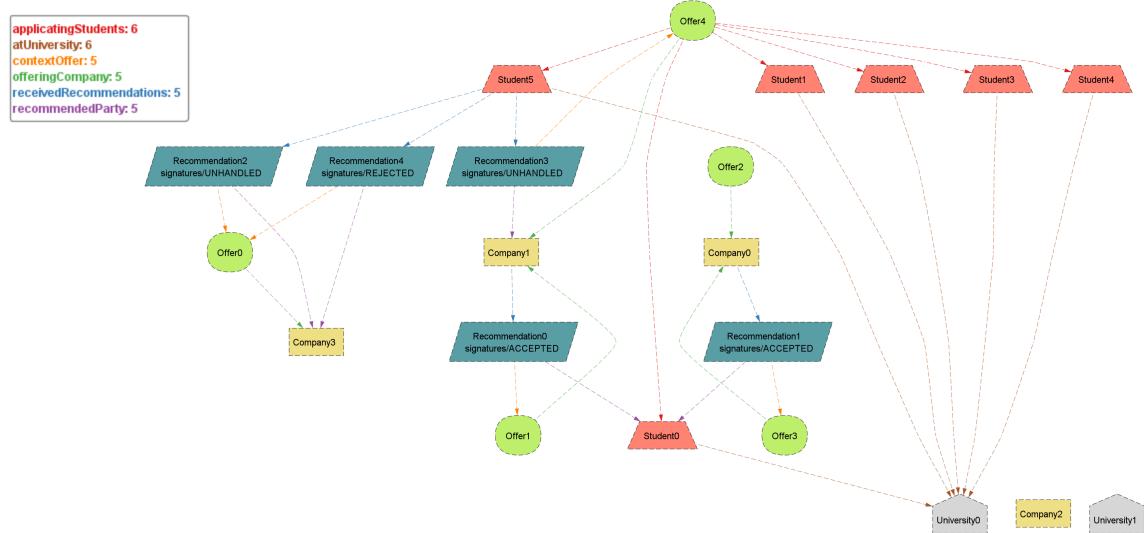


Figure 4.1: Large Static World - a meaningful static world spontaneously generated from the Alloy Analyzer, containing a lot of instances and relationships that abide by the static constraints defined.

4.3.2. Instance of the Example Dynamic World

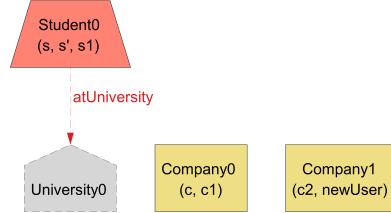


Figure 4.2: Example Dynamic World, Step 1

User instances are added to the world as a result of the `signUp[]` predicate being true.

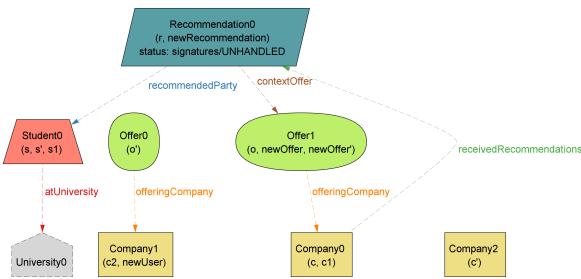


Figure 4.3: Example Dynamic World, Step 2

New Offers are added to the world, and a Recommendation is generated as a result.

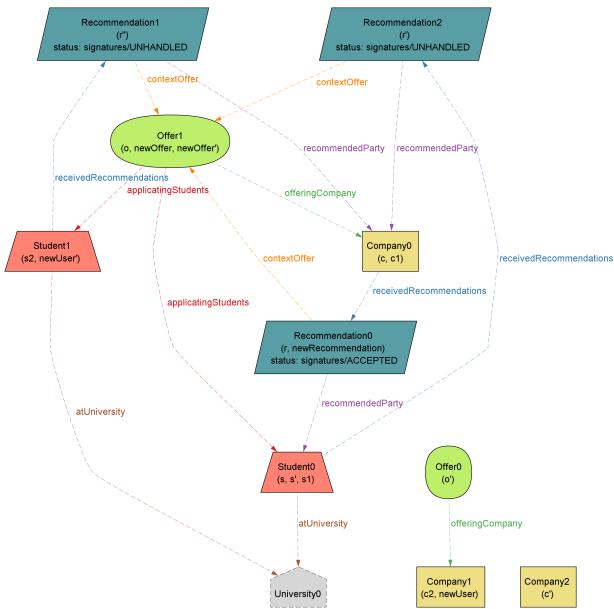


Figure 4.4: Example Dynamic World, Step 3

New Users are added to the world; a Company accepts a Recommendation about a Student who doesn't have a reciprocal Recommendation in that context, causing one to be generated.

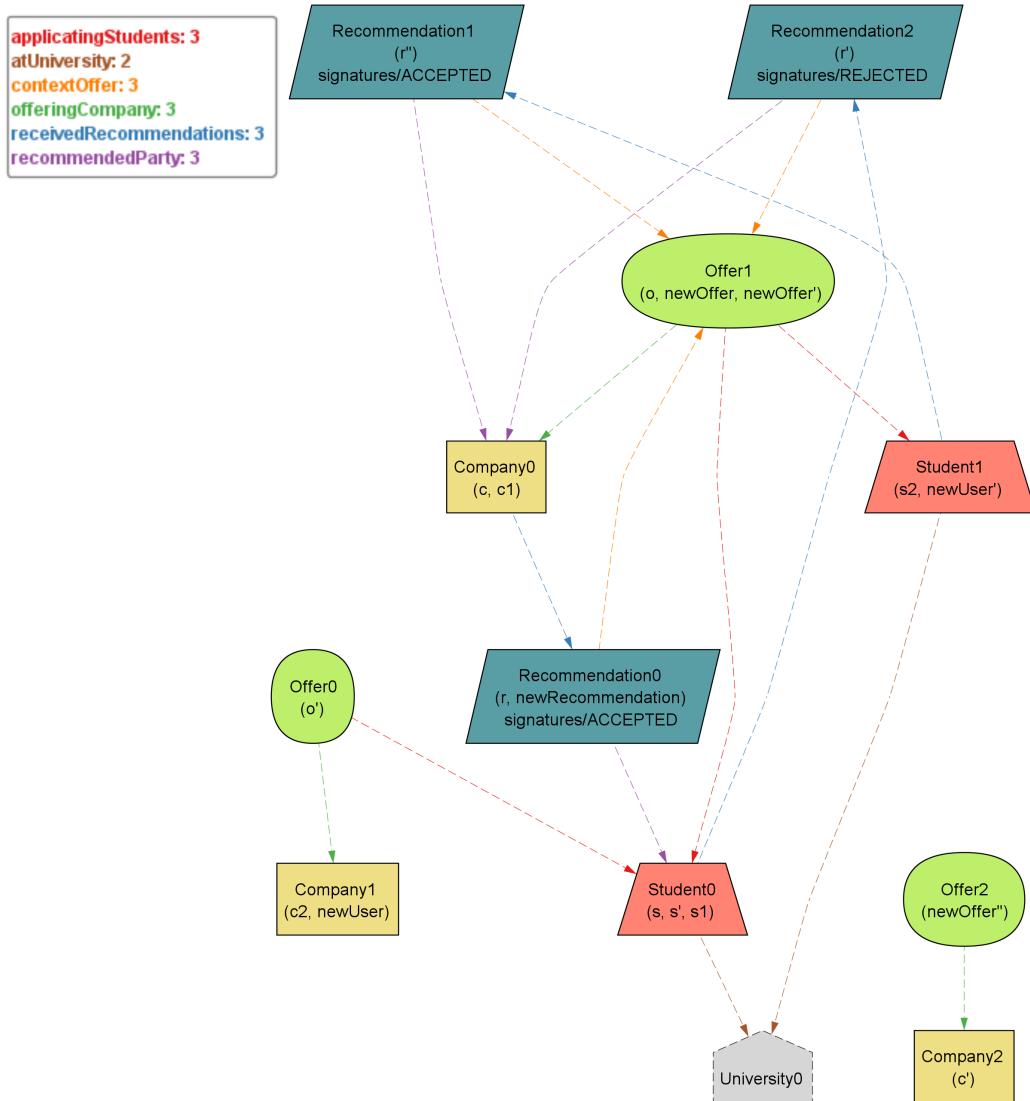


Figure 4.5: Example Dynamic World, Step 4

A Student applies spontaneously to an Offer without passing for a corresponding Recommendation; another Student accepts a Recommendation instead, forcibly applying to the Offer in the context of the Recommendation; a Company rejects a recommendation; a Company without Offers posts a new Offer.

5 | Effort Spent

Group Member	Effort Spent in each Section	
Riccardo Piantoni	Introduction	4h
	Overall Description	7h
	Specific Requirements	6h
	Formal Analysis	17h
	Reasoning	8h
Matteo Rossi	Introduction	5h
	Overall Description	8h
	Specific Requirements	16h
	Formal Analysis	3h
	Reasoning	6h
Jacopo Sacramone	Introduction	4h
	Overall Description	6h
	Specific Requirements	16h
	Formal Analysis	0h
	Reasoning	8h

Table 5.1: Effort spent by each member of the group.

6 | References

6.1. Used Tools

- **Overleaf**: LaTeX collaborative writing.
- **Lucid Chart**: diagrams.
- **Alloy**: formal analysis for the system.

List of Figures

2.1	Domain-Level Class Diagram.	12
2.2	Internship Status State Diagram.	14
3.1	WebApp Map	22
3.2	Use Case Diagram	27
3.3	Sign Up by a Student	29
3.4	Sign Up by a Company	31
3.5	Sign Up by a University	33
3.6	Login by a User	35
3.7	Update User profile	37
3.8	Publish an Internship Offer	39
3.9	Update Internship Offer	41
3.10	Search Internship offer	43
3.11	Apply to an internship offer	45
3.12	Generate Recommendations	47
3.13	Manage Internship Recommendations	49
3.14	Manage Students Recommendations	51
3.15	Manage an Interview	53
3.16	Provide Information for an ongoing Internship	55
3.17	Monitor an ongoing Internship	57
3.18	Report Problems during an Internship	59
3.19	Handle Problems during an Internship	61
3.20	Report Feedback after an Internship	63
3.21	Suggest Optimizations for a Student Profile	65
3.22	Suggest Optimizations for an Internship Offer	67
4.1	Large Static World - a meaningful static world spontaneously generated from the Alloy Analyzer, containing a lot of instances and relationships that abide by the static constraints defined.	84
4.2	Example Dynamic World, Step 1	85
4.3	Example Dynamic World, Step 2	85
4.4	Example Dynamic World, Step 3	85
4.5	Example Dynamic World, Step 4	86