



POLITECNICO
MILANO 1863

**Requirement Analysis
and
Specification Document**

Software Engineering 2

Emanuele Cimino - Gabriele Lorenzetti

Prof. Di Nitto Elisabetta



Contents

Table of Contents	2
List of Figures	4
List of Tables	4
1 Introduction	5
1.1 Purpose	5
1.1.1 Goals	5
1.2 Scope	5
1.2.1 Phenomena	7
1.3 Definitions, Acronyms, Abbreviations	8
1.3.1 Definitions	8
1.3.2 Acronyms	8
1.3.3 Abbreviations	8
1.4 Revision history	8
1.5 Reference Documents	9
1.6 Document Structure	9
2 Overall Description	10
2.1 Product perspective	10
2.1.1 Scenarios	10
2.1.2 Domain-level Diagram	12
2.1.3 State Diagrams	13
2.2 Product functions	16
2.3 User characteristics	17
2.4 Assumptions, dependencies, and constraints	17
3 Specific Requirements	18
3.1 External Interface Requirements	18
3.1.1 User Interfaces	18

3.1.2	Hardware Interfaces	18
3.1.3	Software Interfaces	18
3.1.4	Communication Interfaces	18
3.2	Functional Requirements	19
3.2.1	Use case diagrams	19
3.2.2	Requirements	36
3.2.3	Mapping on requirements	37
3.3	Performance Requirements	37
3.4	Design Constraint	37
3.4.1	Standard Compliance	37
3.4.2	Easy to use	38
3.4.3	Hardware limitations	38
3.4.4	Any other constraint	38
3.5	Software system attributes	38
3.5.1	Availability	38
3.5.2	Modularity	38
3.5.3	Maintainability	38
3.5.4	Security	38
3.5.5	Portability	38
4	Formal Analysis Using Alloy	39
4.1	Main Objectives	39
4.2	Static Model	39
4.2.1	Signatures	39
4.2.2	Facts	40
4.2.3	Predicates	41
4.3	Dynamic Model	43
4.3.1	Signatures	43
4.3.2	Facts	44
4.3.3	Predicates	46
5	Effort Spent	50

List of Figures

1	Class diagram	12
2	Proactive selection	13
3	Recommendation process	13
4	Selection process	14
5	Collection process	15
6	Use case diagram	19
7	AD1 - Register user	21
8	AD2 - Login user	23
9	AD3 - User data management	25
10	AD4 - Internship insertion	27
11	AD5 - Student proactive search	29
12	AD6 - Student notification of an available internship	31
13	AD7 - Company notification of an available student for an internship	33
14	AD8-9-10 - User leaves...	35
15	Static Alloy Model	42
16	DAM - 0	47
17	DAM - 1	47
18	DAM - 2	47
19	DAM - 3	48
20	DAM - 4	48
21	DAM - 5	49

List of Tables

1	Phenomena	7
2	UC1 - Register user	20
3	UC2 - Login user	22
4	UC3 - User data management	24
5	UC4 - Internship insertion	26
6	UC5 - Student proactive search	28
7	UC6 - Student notification of an available internship	30
8	UC7 - Company notification of an available student for an internship	32
9	UC8 - User leaves a feedback	34
10	UC9 - User leaves review	34
11	UC10 - User leaves a complaint	34
12	Requirements	36
13	Mapping on requirements	37
14	Effort Recap	50

1 Introduction

1.1 Purpose

If you are looking for an internship or want to offer one, you are in the right place, Student&Companies will help you! S&C is a platform that helps match university students looking for internships and companies offering them.

The primary objectives of the platform are as follows:

- Students can search for an internship on their own.
- The system will help both students and companies to find each other through a system called "Recommendation", which helps to find the correspondence between the characteristics offered by the student and those requested by the company.
- The management of the selection process.
- The monitoring of the execution of the internships, including statistics, feedback, complaints and reviews.

1.1.1 Goals

[G1]: All unregistered students and companies must be able to subscribe and login to the S&C platform.

[G2]: Students and companies must be able to write their descriptions and preferences.

[G3]: Students must be able to complete their CV.

[G4]: Companies must be able to create internship offers.

[G5]: Students must be able to search for an internship.

[G6]: Student and Company are informed when there is a match between them.

[G7]: Monitoring the execution of the selection process.

[G8]: Monitoring the execution of the internship.

[G9]: Statistics collection.

[G10]: Feedback collection.

[G11]: Complaints collection.

[G12]: Reviews collection.

[G13]: Companies rank based on reviews.

1.2 Scope

S&C allows student and companies to communicate easily in a guided environment.

Students can upload their CVs and profile. Companies upload their internship offers through the platform.

Students need to be able to actively search for an internship: with the search-bar through a keyword or scrolling the main list, he/she can also use some filters.

The system implements a process called "recommendation": it is a mechanism that applies a research

through the internship preferences and students' characteristics; it informs, with a notification, students when an interesting internship is available and informs also companies that a student matches with their preferences.

The platform also proposes to help companies in the selection process. When there is a match between the two parts and both of them accept it, or the company is interested in the student that asked for an opportunity, the process starts. Companies feed, with the help of the system, questionnaire to the students and collect their answers, so that companies can assess their fit with the student and finalize the selection.

The platform also stores statistics based on the choices made by companies and students, it stores feedback from both at mid-term and at the end of the internship, also a review must be left after the end of the internship. It also helps with the collection of complaints from both parts of the interest. All these stored data are needed to sharpen the recommendation system, throughout the updating of the statistical analysis which is at the base of its functioning.

1.2.1 Phenomena

Phenomenon	Who controls it?	Is shared?
Users decides to use S&C	W	N
User registration	W	Y
User login	W	Y
Check username and password	M	N
Student compile CV	W	Y
Company create offer	W	Y
User profile manage	W	Y
Recommendation process	M	N
Student search offer	W	Y
System shows offers list and results of a search	M	Y
Student request interview	W	Y
Notify the company of the interview request	M	Y
Acceptation of the requested interview notification	W	Y
Match notification	M	Y
Match acceptance	W	Y
Interview process start	M	N
System send form	M	Y
Student send form compiled	W	Y
System send notification of compiled form to company	M	Y
Company decides which students to accept and which not	W	Y
System send notification of result to students	M	Y
User can compile complaint	W	Y
User interested is notified of the complaint	M	Y
User's feedback on the midterm notification	M	Y
User compile feedback on the midterm	W	Y
User's feedback on ending internship notification	M	Y
User's review on ending internship notification	M	Y
User compile feedback on ending internship	W	Y
User compile review on ending internship	W	Y
Statistics collection and analyses	M	N
Companies rank computation	M	N
Companies rank publication	M	Y

Table 1: Phenomena

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **User:** It is a generic user of the platform, who wants to offer a job (Company), or try to find an offer (Student).
- **Complaint:** It is a form that can be compiled from users to inform their counterparts about problems.
- **Feedback:** It is a form that must be compiled from users to evaluate their fit during and after the internship with their counterparts.
- **Review:** It is a form that must be compiled from users at the end of the internship to leave a comment and evaluate with a score their counterpart.
- **Statistical analysis:** Analysis based on the previous general choices of students, companies and feedback.
- **Filters:** Used by the student to make an advanced search for the apprenticeship, applying the so-called filters, such as minimum salary or company name.

1.3.2 Acronyms

- **S&C:** Student and Companies
- **CV:** Curriculum vitae
- **GDPR:** General Data Protection Regulation is EU regulation for personal data and privacy.
- **TCP/IP:** Transmission Control Protocol and Internet Protocol.
- **TLS:** Transport Layer Security is a cryptographic protocol designed to guarantee communications security, providing authentication, data integrity and -confidentiality.
- **UTC:** Universal Time Coordinated is the most used time standard.

1.3.3 Abbreviations

- **[Gn]:** Used to list all the goals, it stands for the n-th one.
- **[An]:** Used to list all the domain assumptions.
- **[UCn]:** It refers to the n-th use case provided.
- **[ADn]:** It refers to the n-th activity diagram.
- **[DAMn]:** It refers to the n-th state of the dynamic alloy model.

1.4 Revision history

- Version 1: 22/12/2024

1.5 Reference Documents

- Slides of the course “Software Engineering 2”.
- Michael Jackson, The World and the Machine.
- Specification document "01. Assignment RDD AY 2024-2025”
- Alloy documentation - <https://alloytools.org/documentation.html>

1.6 Document Structure

1. **Introduction:** a description of the problem showing the purpose and the scope of the application. In order to precisely delineate the scope, phenomena and goals related to the problem are identified. In this section information about terms used in this document is also present, along with references and revision history.
2. **Overall Descriptions:** a high-level view of the project. The perspective of the product is developed with scenarios and descriptions about requirements of the service interfaces. Product functions describe the required functions of the system in order to fulfill the goals as specified by the stakeholders. Furthermore, possible actors are also identified in the user characteristics section. In the end there is a list of the taken domain assumptions.
3. **Specific Requirements:** detailed description of the user interface, function and non-functional service requirements specification. Functional Requirements are supported from specific use cases and mappings that permit to acknowledge how the goals are satisfied.
4. **Formal Analysis Using Alloy:** Alloy model useful in checking necessary properties of the system, and generating possible world in which the same will operate.
5. **Effort Spent:** hours spent by each group member on the various activities related to the document developing.

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

Here are presented possible scenarios for the users of the S&C platform.

1. The Nexus corp. company wants to have access to the S&C services:

Nexus corp. would like to offer one or more internships to students, but does not know how to find/contact them and how to choose the most suitable one. Therefore, the company registers on the S&C platform, enters its name, an email address, a password, its VAT number, a description of what they are interested in within relative fields. For subsequent times, to access it only has its e-mail and password.

2. Francesco wants to have access to the S&C services:

Francesco wants to have the opportunity to gain more experience during his studies or, also, to earn some money, but he does not know which company to choose, so he subscribes to the platform S&C. When subscribing, Francesco enters his/her first name, last name, e-mail, a password, a description of itself, compile a form with his CVs' information within relative fields. For subsequent times, to access he/she only has to enter his e-mail and password.

3. Nexus corp. insert internship offers

Now that it is inside the platform, Nexus corp. can add its new internship offer, inserting the title, an adequate description, the start and the end date of the internship, the deadline to accept application, the characteristics and the questions to be included in the relative form, including the deadline to collect responses, that will be filled out by the students at the beginning of the selection process.

4. Francesco/Nexus corp. manage personal data and description

Now that it is inside the platform Francesco can modify his/her CV and Nexus corp. can insert offers. Both can also modify/update their personal data and description.

5. Francesco/Nexus corp. look at the ranked companies list

Francesco/Nexus corp. from the home page can click the button "Best places to work" and the system will display the ranked list of the companies.

6. Francesco search for an internship

To search for an internship, Francesco can scroll or enter a keyword ,i.e. the name of a company, in the search bar and adding some filters if he want. Once he has chosen the offer that is right for him, he can request to start the selection process and S&C will take care of notifying the company in question or he can decide to close the search.

7. Recommendation system notification

Based on the CV inserted by Francesco, the offers by companies and the statistical analysis from the data collected by the platform, both Francesco and Nexus corp. get notified when a match is available, throughout the research made by the recommendation system.

8. Acceptance phase

When Francesco or Nexus corp. receive a notification, they can see details of the notified user, they have to accept or refuse it through a button.

9. Selection process

After the establishment of a contact, the selection process start, and Francesco has to compile a

form with specific questions to check if he/she really fit with the position. If it is not compiled before the deadline Francesco is rejected. After receiving responses Nexus Corp. (if it's chosen) decide if Francesco is suitable by accepting him, else it can refuse him before the deadline, and Francesco can accept or not the internship before the end of selection process deadline.

10. Complaint collection

Francesco/Nexus corp. go to the active internship and it select the action inserting complaint and send it.

11. Midterm feedback system notification

Reached the midterm of the internship, the system will automatically send notification of the availability of the feedback form to both Francesco and Nexus corp. , if involved, that can be compiled in a week.

12. Ending internship

At the end of the internship, the system will automatically send notification of the end of the internship with the availability of feedback and review forms to both Francesco and Nexus corp. , if involved.

2.1.2 Domain-level Diagram

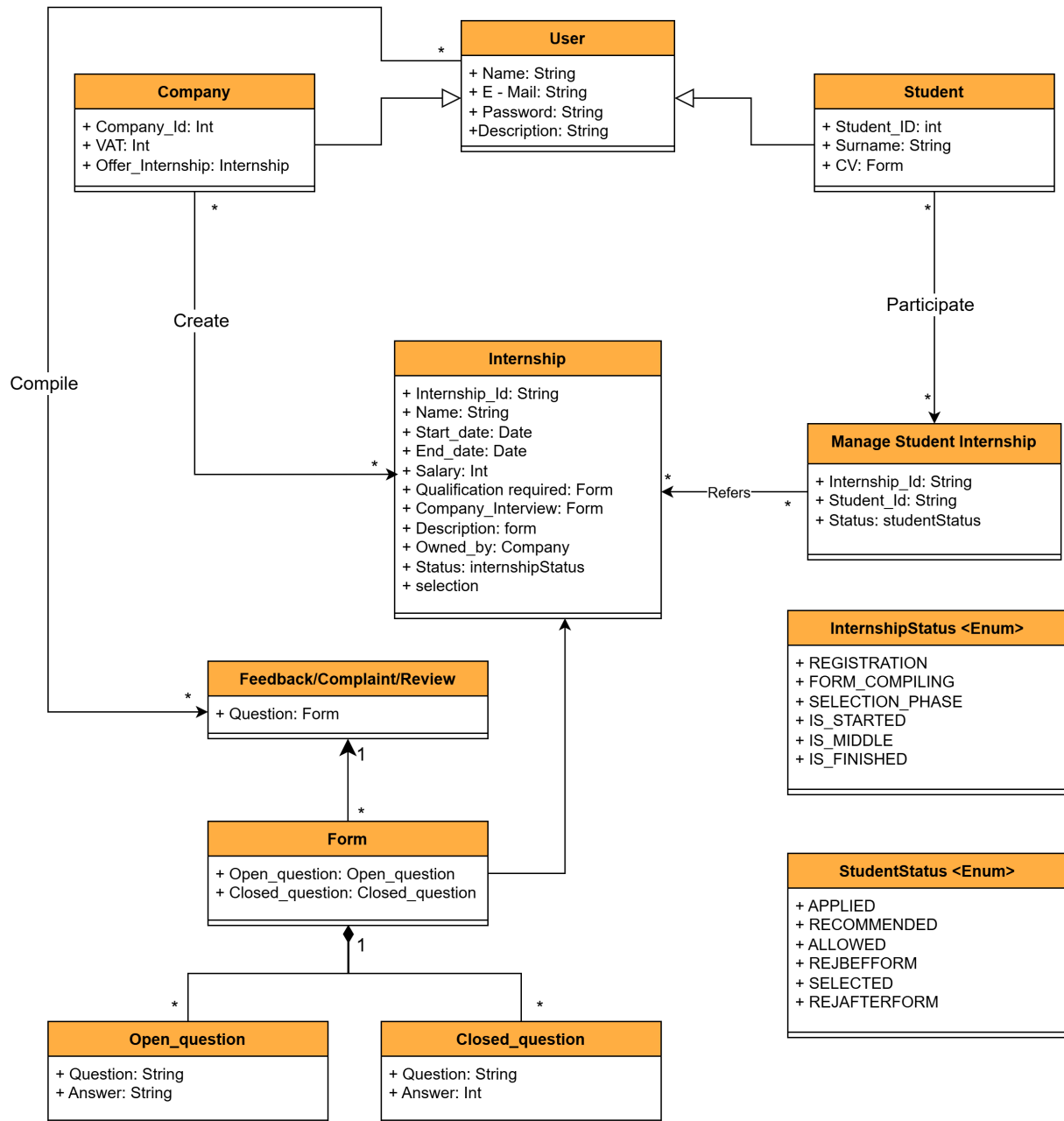


Figure 1: Class diagram

2.1.3 State Diagrams

State diagrams describe the behavior of the system while considering all possible states the system can deal with when an event occurs. This analysis helps to clarify the most critical aspects of the system.

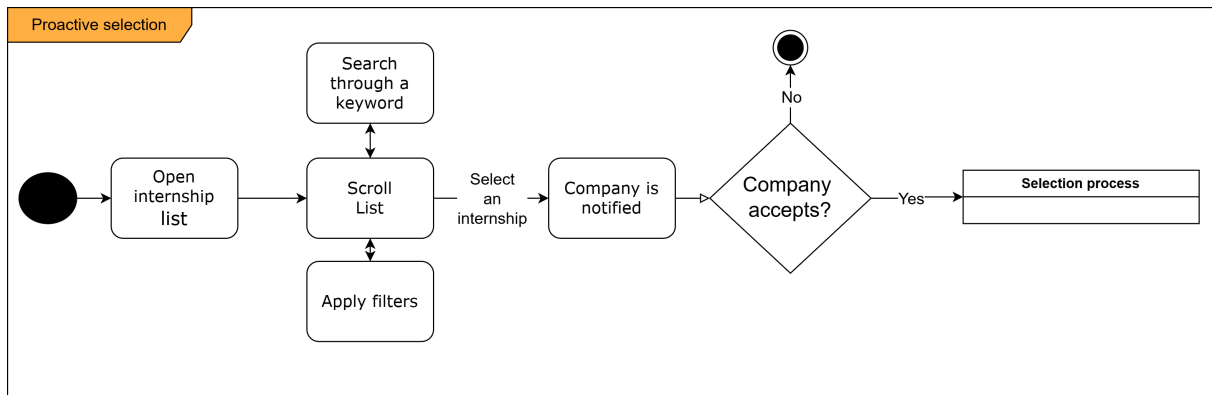


Figure 2: Proactive selection

This state diagram describes the process of manual selection of the internship by the student. In the initial state, the "Open internship list" the system is displaying the last inserted internships, and the student can select one of them, he/she can also use filters or/and the search bar to filter the main list, when is selected the system goes into the "Company is notified" state and notify the company interested, that can accept or not the request, in case of acceptance, the selection process start.

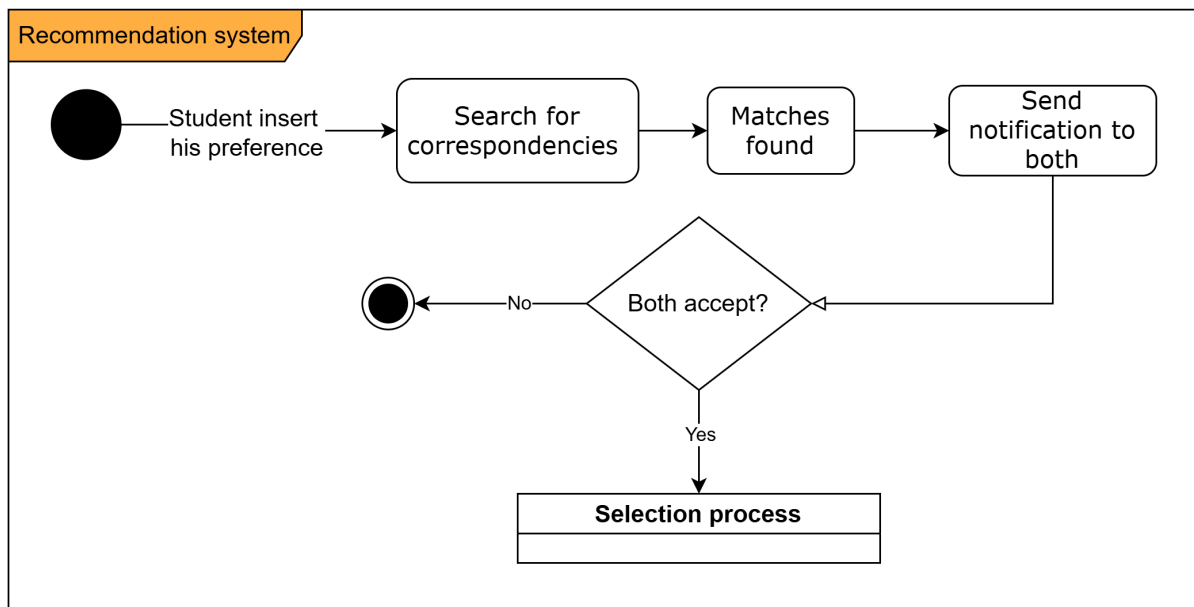


Figure 3: Recommendation process

After the insertion of the preferences by the student, the system goes to the state of "computing correspondences" where it actually searches for an internship that can match with the preferences of a company, through a keyword search of the preferences, through the CVs' contents or a statistical analysis of the internships. The system always found the best match, then proceed to the state of "notification" where it notify both the counterparts, if both of them accept, then the system proceed with the selection process; otherwise, the selection process never starts.

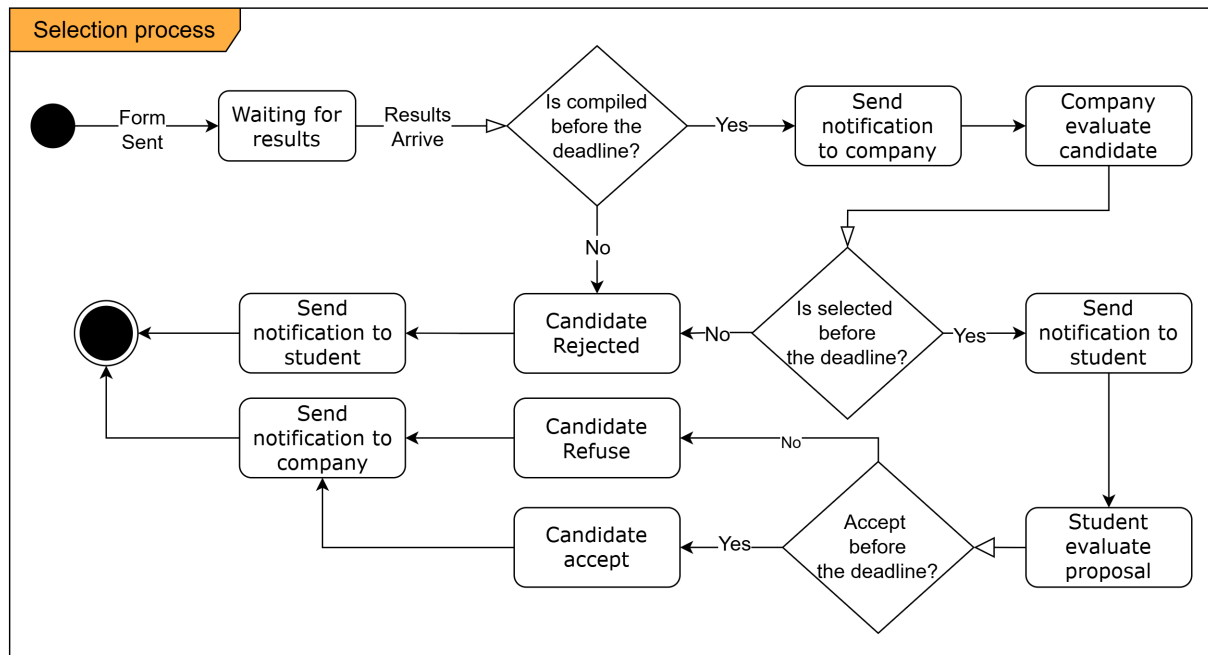


Figure 4: Selection process

This state diagram describes the selection process: once it starts, the system sends a form and goes to the "Waiting for results" state, where it waits until the student compiles the form and sends back the results. Then when they arrive, a notification is sent to the company and it goes to the "Company evaluate candidate" state, where the company can analyze the form results and decide to choose some students. If a student is selected the system proceeds with the notification and goes to the "Student evaluate proposal", where the student can accept or refuse the internship, the system will send a notification to the company with the respond. Every part of the selection process follows some deadline indications, given by the company when an internship is inserted, if its are not respected the student is rejected.

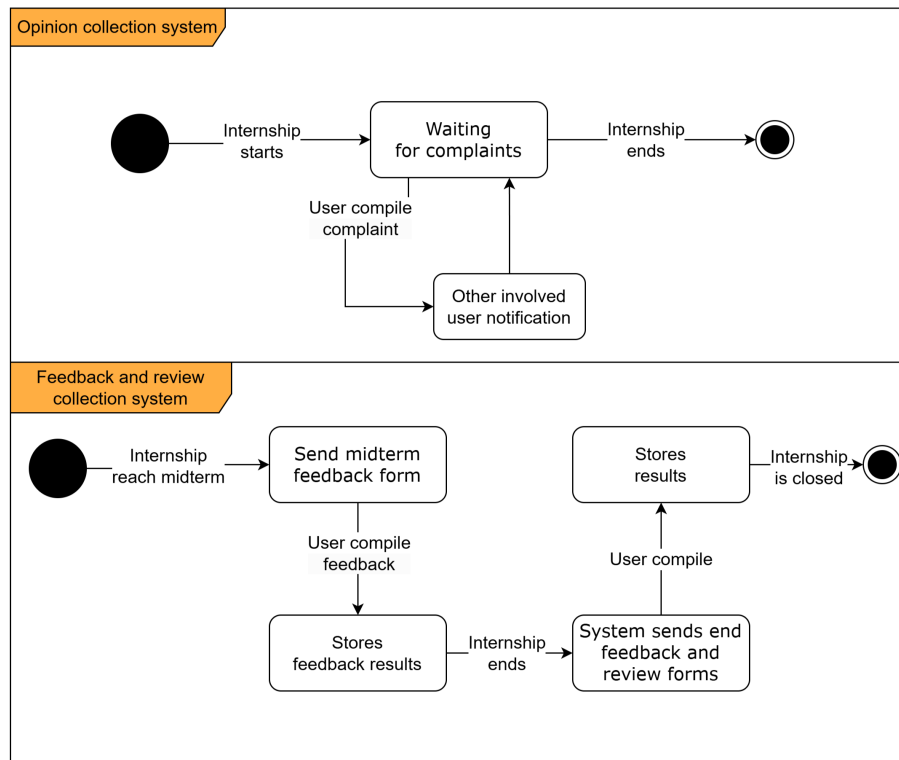


Figure 5: Collection process

This state diagram describe the collections process, there are two different process but they are described together because of their similarity.

1. Let's start from the Opinion collection system, once it starts, the system goes to the "Waiting for complaints" state, until either the internship ends or a user (student or company) compile a complaint, when this happen the system goes to "Other involved user notification" state where the system notify the other user (company or student) and goes back to the previous state.
2. The Feedback and review collection system is different in form to the previous one because the information that are inserted are in this case stored by the platform. The process starts at the midterm of an internship, when the system sends the midterm feedback form, wait for the user to compile and stores the feedback, it happens the same at the end of the internship when are sent to the user a feedback and a review form to compile, when compiled the system stores the results, the internship get closed and the process ends.

2.2 Product functions

The following section contains the main product functions of S&C.

- **Sign up and Login**

These functions will be available to all users. The sign-up functionality allows users to create an account to register on the platform. Each user will be asked to select if he/she is a student or a company and provide own data such as name, email and password; for the student will also be asked surname, his/her description/preferences and CV, for the company the VAT number and a description of itself. The login functionality allows users to access an existing account using the credentials (email and password) chosen at the time of registration.

- **Managing internships and data**

This function allows companies to insert their offers into the platform, it requires a name, a start and an end date, a salary, the required qualification and a description with other details. Also in this phase the interview form is inserted to the platform.

- **Managing user data**

This function allows users to insert updated data.

- **Autonomous research**

This function allows a student to search through all available internships and find what suits him/her better and he/she ask the company to start the selection process. He/She can simply search by: scrolling the list, using some keywords and through the application of some filters.

- **Recommendation system**

This function search for a correspondence between students and companies, taking in consideration CVs and preferences of students, the requests and requirements of the projects offered by companies. If this search fails, it takes into account a statistical analysis approach that produce the most suitable match from the list of internships. Fit is mainly checked through feedback.

- **Selection process**

This function handle the selection process, once all the parts have accepted to collaborate, S&C sends the form, made by the company, to the student to interview him/her and when the student send it back, the system check if it has done in time, if it is a yes, the system sends it back to the company so it can evaluate the student, if it's a positive evaluation and before the deadline, he/she receive the notification and can decide to accept the collaboration and the company get notified. If the deadlines are not respected the student is automatically rejected.

- **Feedback process**

This function help users to insert their feedback on the company or on the student, using an automatic notification to alert users. Each user must fill in feedback halfway through the internship and one at the end. Feedback remains open for one week. Feedback is collected through a form made by S&C.

- **Review**

After the end of a internship the users involved leave a review, which will be visible on the corresponding profile to all platform members.

- **Complaints** The user can, at any time during the internship, send a complaint that will be visible only to the interested user.

- **Rank companies list**

This function extract the score of the companies from the review. Creates a rank of the "best place to work" in an internship and place them on a list.

2.3 User characteristics

Here it is provided a more detailed characterization of the two different types of users of the platform, also specifying the roles they can assume in different contexts.

- **Student**

A student can be anyone who wants to find an internship and his/her is a real university student. After his/her registration, he/she gets the functionalities which are reserved for student such as managing data and autonomously searching for an internship.

- **Company**

A company can be anyone who wants to offer an internship and has a VAT number. After its registration, gain access to the functionalities which are reserved for companies, such as offers new internships and its managing.

2.4 Assumptions, dependencies, and constraints

- [A1]: The user always selects the correct option about his/her/its position (student/company)
- [A2]: All users have an active email address.
- [A3]: All users have access to an internet connection.
- [A4]: All students are a real student, subscribed to a University.
- [A5]: All users insert real and plausible data.
- [A6]: All companies select the number of students based on their needs for the internship.
- [A7]: The feedback and reviews are always compiled.
- [A8]: A student participates in only one internship at the same time.
- [A9]: An internship is proposed by only one company.
- [A10]: Companies always enter correct data in internships, so that they do not need to modify them later.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The platform is featured with several Graphic User Interfaces such to allow all types of users to interact with all its functionalities. The most important ones are presented below:

- **Registration/Login Interface**
This interface provides for two different forms to fill in personal data, either to register on the platform or to access the Home Page.
- **Home Page**
In this page the student visualizes key information about the upcoming internship, and also inserts filters or/and keyword to search on them; the company visualizes its internship. Both can visualize their notifications. All user collaborations are visible in this page, from this list it can insert feedback, complaints and review.
- **Private profile page**
With this interface, the user can check and edit its personal data set during registration. All the received complaints are visible in this page and also own reviews are visible.
- **Public company profile page**
With this interface, when students are searching for an internship or just looking at the ranked list, they can click on a button to view details of the company, its offered internships and reviews from other students.
- **Public student profile page**
With this interface, when a company is notified of a potential new student, they can click a button to view details about him/her, such as CV and reviews from other internships.
- **Ranked companies list page**
With this interface users can view the ranked companies list.
- **Internships management**
With this interface, a company can add new internship. Can also open the form responses page from here.
- **Company form responses page**
With this interface, the companies can view all responses from the students accepted and can select or reject them.

3.1.2 Hardware Interfaces

The platform does not provide any hardware interface since it is primarily a platform to find an internship: it does not require any external component or device other than the one it runs on.

3.1.3 Software Interfaces

Software Interface are not needed.

3.1.4 Communication Interfaces

The platform exploits the internet connection for communication to the main server, whose role is to manage all back-end functions such as storing data, responding to deadlines, and so on.

3.2 Functional Requirements

3.2.1 Use case diagrams

In this section, some of the most significant Use Cases for S&C platform have been represented, dividing them into two groups, one for each category of users.

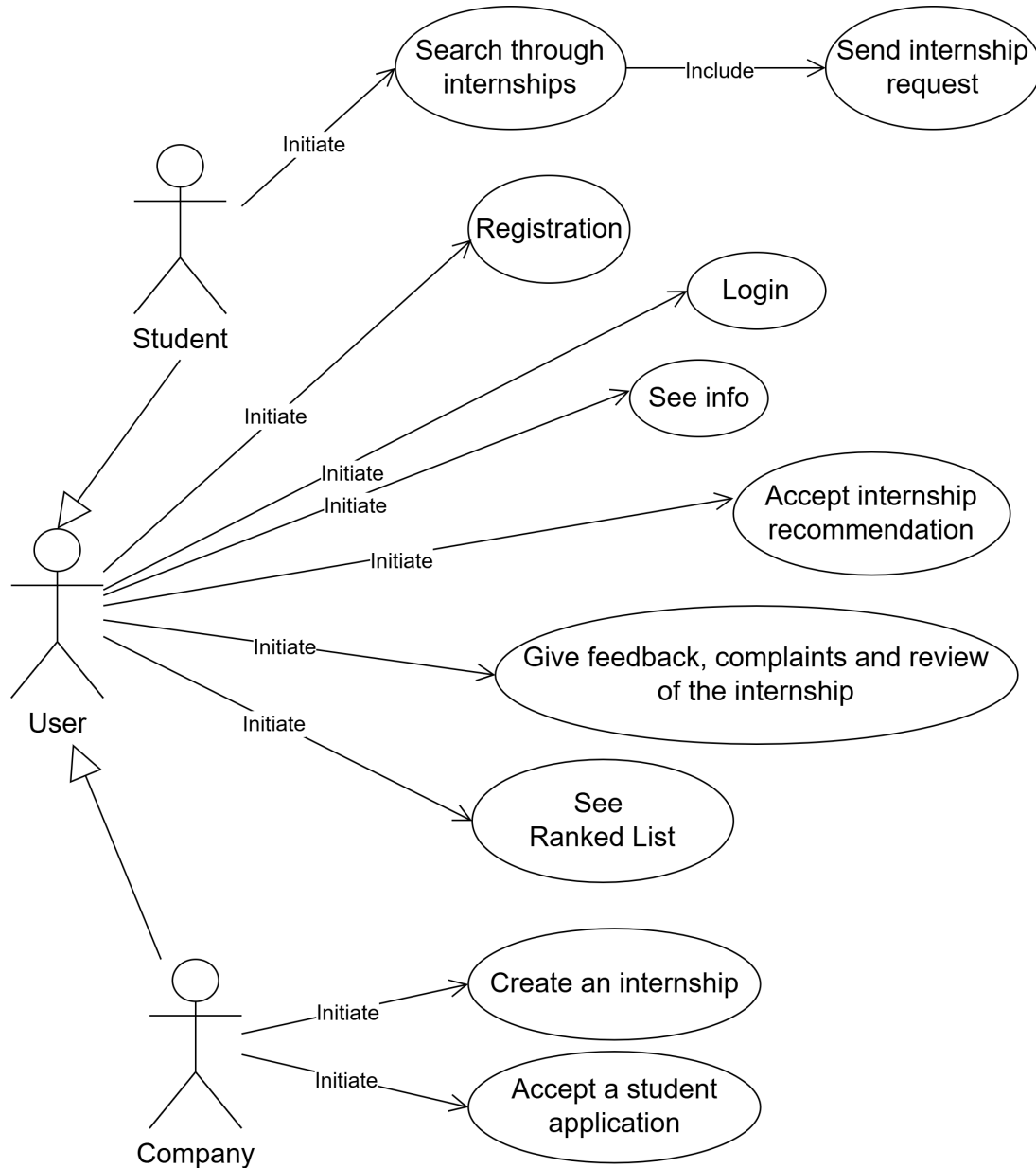


Figure 6: Use case diagram

Name	Register user
Actors	User
Entry conditions	User has opened the platform
Event flow	<ol style="list-style-type: none"> 1. The User requests to register. 2. The system asks the user to choose if it is a Student or a Company and to provide relative data. 3. User submits all necessary information. 4. The system checks if email has been already used by another user to register. 5. System updates the database with the User's information and displays a message of confirmed registration.
Exit conditions	User has successfully registered
Exception	User provides an email already registered in the database. The system displays an error.

Table 2: UC1 - Register user

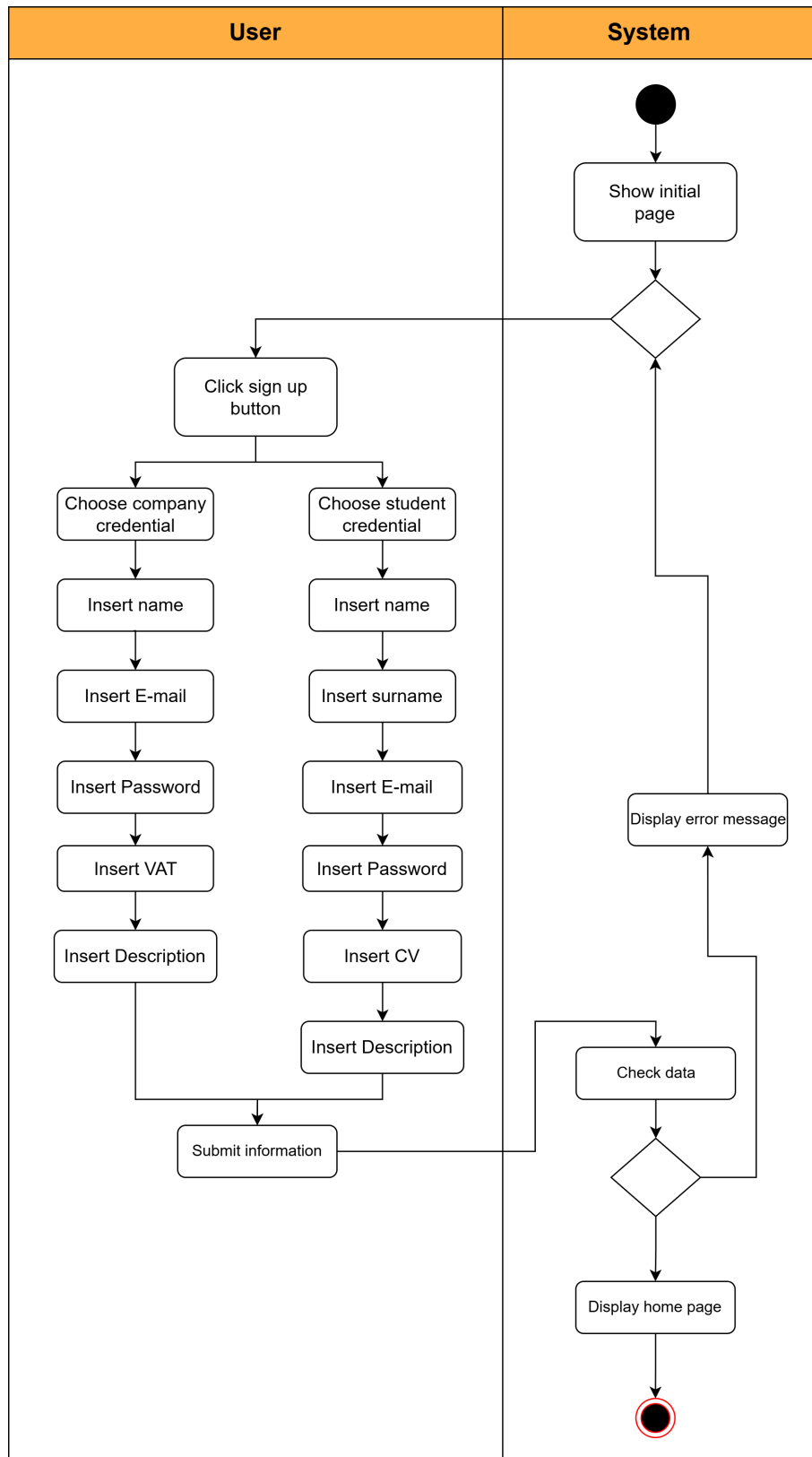


Figure 7: AD1 - Register user

Name	Login user
Actors	User
Entry conditions	User has registered to the platform
Events flow	<ul style="list-style-type: none"> • User inserts in apposite fields the credentials for logging in (username and password) and presses the “Login” button. • The system checks the correctness of the credentials inserted • The system displays the home page
Exit conditions	User is logged in
Exception	User inserts wrong combination of credentials, presses Login button and the application displays the Login page with an error.

Table 3: UC2 - Login user

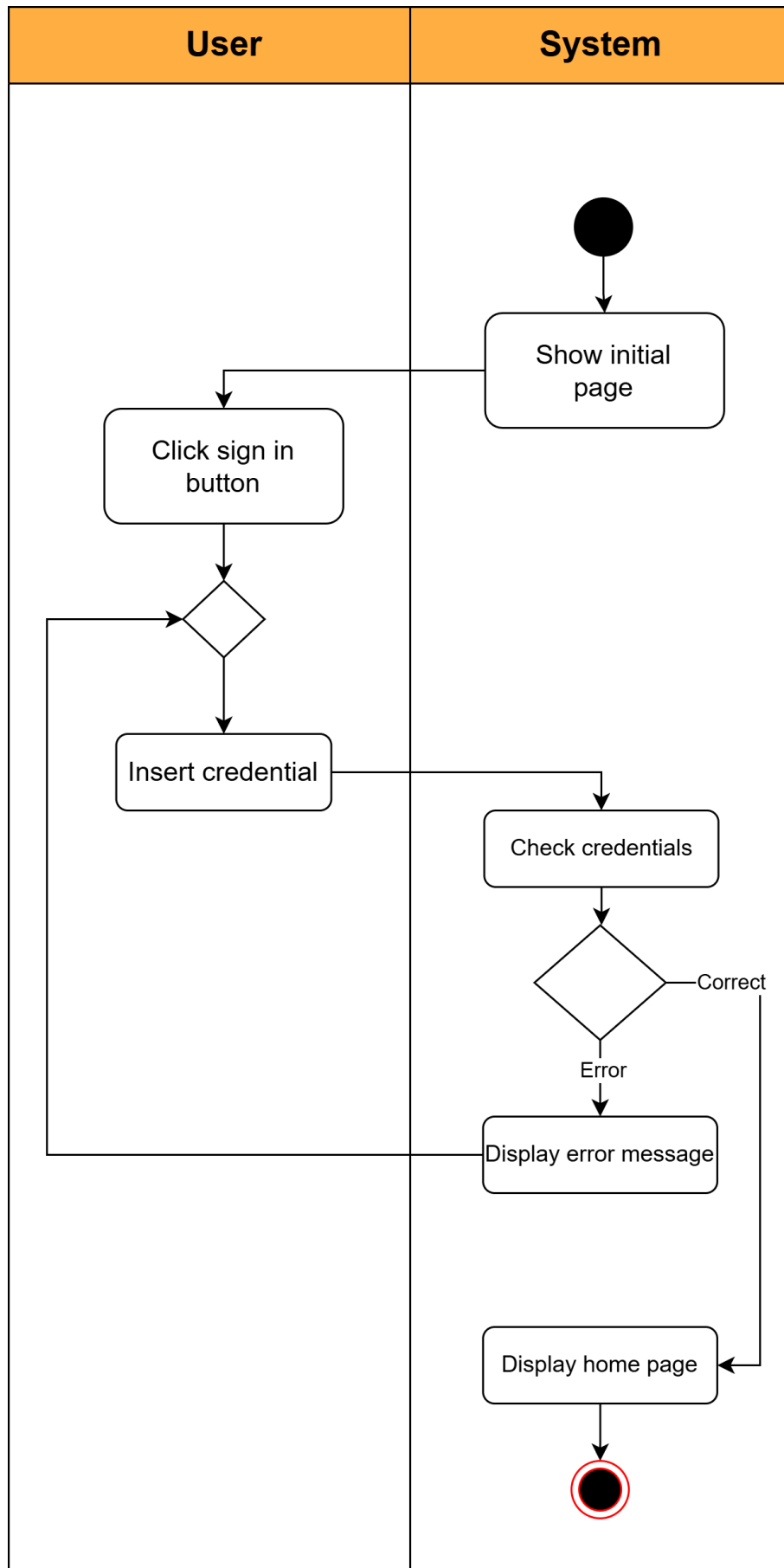


Figure 8: AD2 - Login user

Name	User data management
Actors	User
Entry conditions	User has logged in to the platform
Events flow	<ol style="list-style-type: none"> 1. System shows home page. 2. User clicks on button "My data". 3. System shows user data. 4. User clicks on button "Modify data". 5. User can modify its information. 6. User clicks on button "insert". 7. System checks validity. 8. System shows the data updated.
Exit conditions	User clicks the button "Save"
Exception	User changes e-mail, inserts a not valid one and presses "insert button" and the application displays an error message.

Table 4: UC3 - User data management

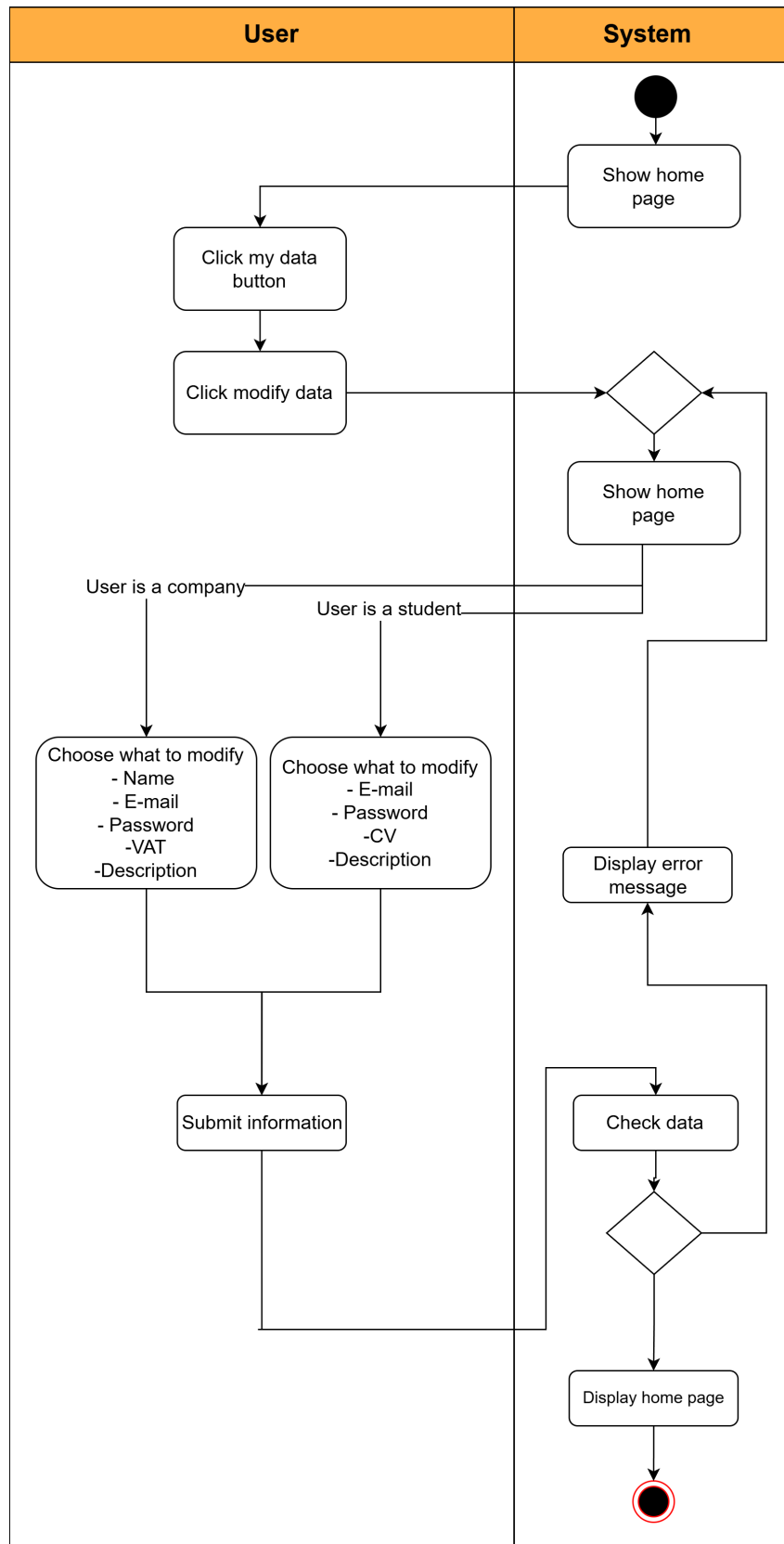


Figure 9: AD3 - User data management

Name	Internship insertion
Actors	Company
Entry conditions	Company has registered/logged in to the platform
Events flow	<ol style="list-style-type: none"> 1. System shows the home page. 2. Company clicks on button insert "New internship". 3. Company inserts valid information of an "internship", such as name, start date, end date, salary, qualification required, a description and the interview form. 4. Company clicks on button "Insert". 5. System checks validity. 6. System shows the inserted internship list updated.
Exit conditions	Company insert a valid internship
Exception	Company inserts a not valid date, presses "insert button" and the application displays an error message

Table 5: UC4 - Internship insertion

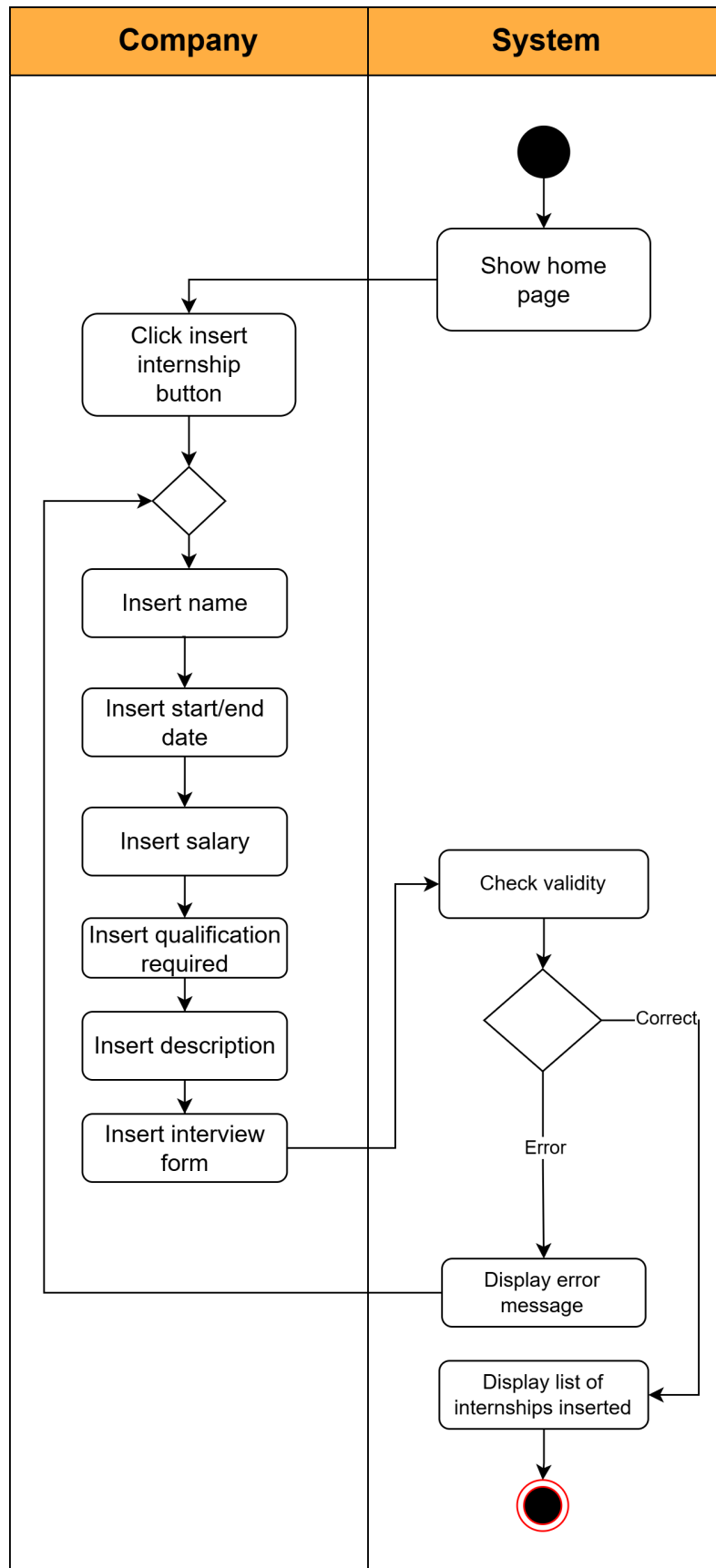


Figure 10: AD4 - Internship insertion

Name	Student proactive search
Actors	Student
Entry conditions	Student has logged in to the platform
Events flow	<ol style="list-style-type: none"> 1. System shows the list of available internship in home page, a search bar and filters button. 2. Student search through the list. <ol style="list-style-type: none"> (a) Student scrolls and clicks button "Show details". (b) Student writes a keyword on the search bar and presses the button "search". <ol style="list-style-type: none"> i. System produces some results ii. Student scrolls the list of result and clicks button "Show details". 3. System shows the selected internship information.
Exit conditions	<ol style="list-style-type: none"> (a) Student clicks on button "Ask for an internship". (b) Student clicks on button "Close the research"
Exception	Search does not produce results, the application produces an error message and goes back to the full list

Table 6: UC5 - Student proactive search

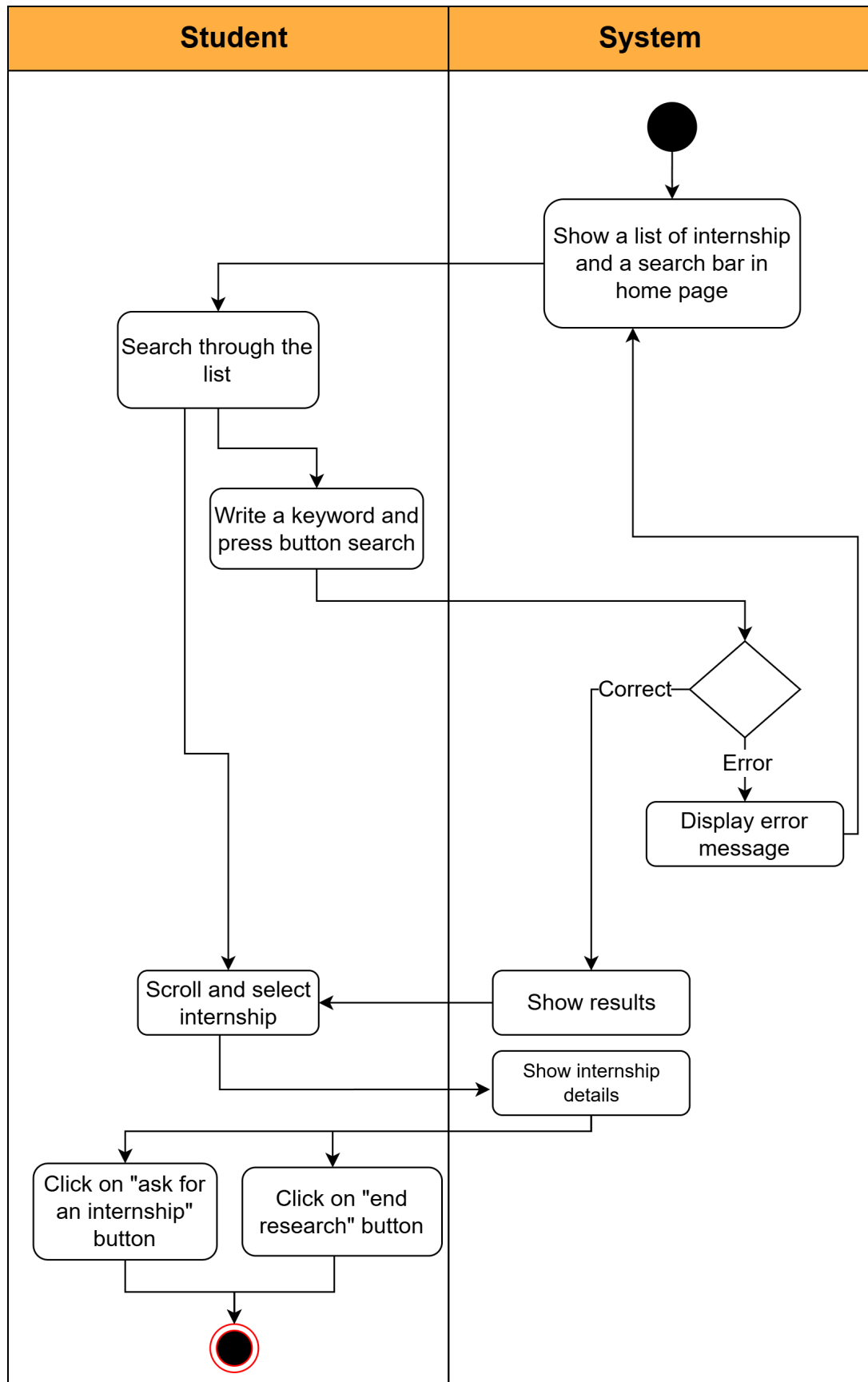


Figure 11: AD5 - Student proactive search

Name	Student notification of an available internship
Actors	Student
Entry conditions	Recommendation system find a match.
Events flow	<ol style="list-style-type: none"> 1. The notification appears on the home page of the student. 2. Student clicks on the notification.
Exit conditions	<ol style="list-style-type: none"> (a) The student decides to accept the internship. (b) The student decides to reject the internship. (c) The deadline for accepting/rejecting has passed.

Table 7: UC6 - Student notification of an available internship

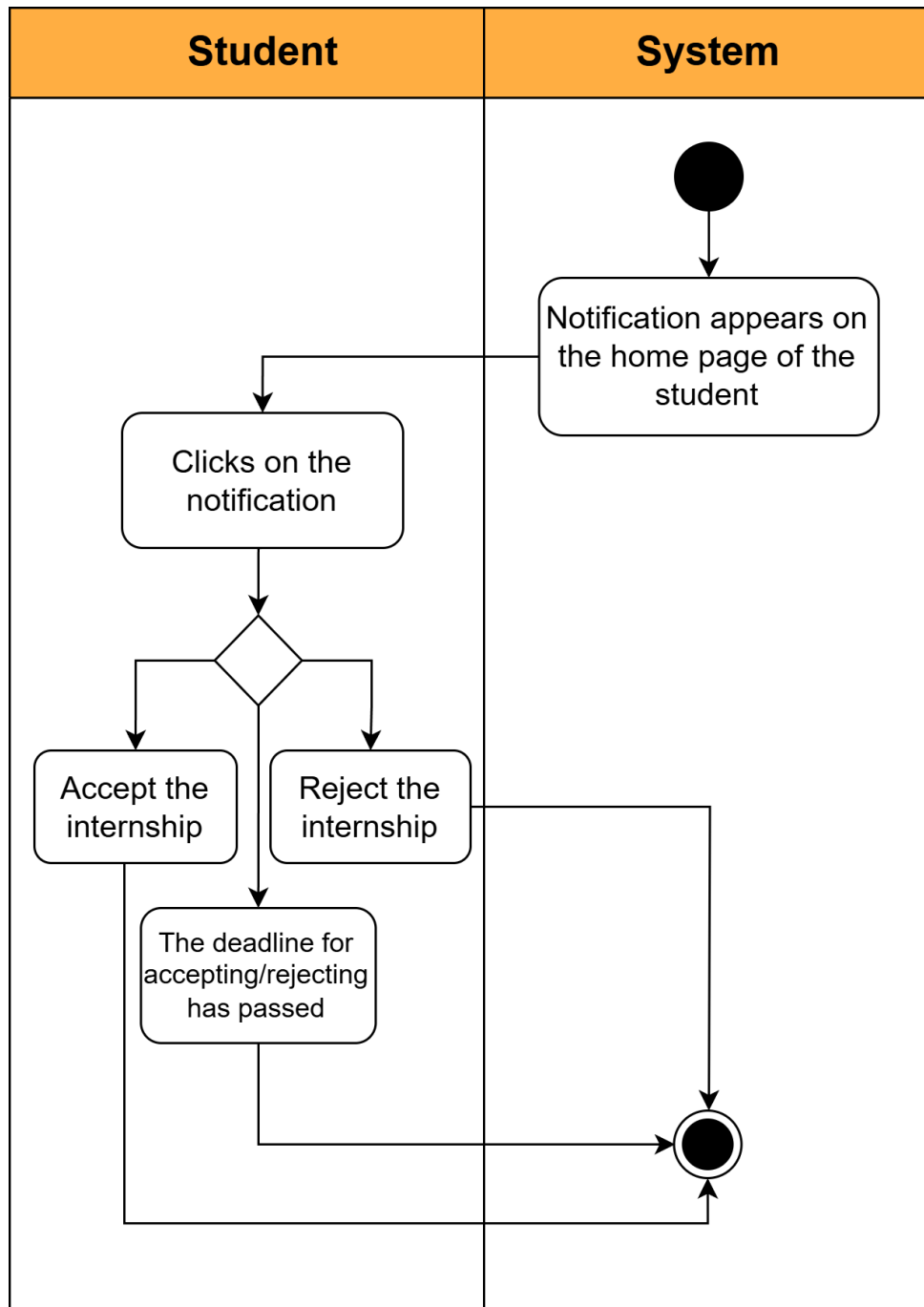


Figure 12: AD6 - Student notification of an available internship

Name	Company notification of an available student for an internship
Actors	Company
Entry conditions	<ul style="list-style-type: none"> (a) A student ask for an internship". (b) Recommendation system find a match.
Events flow	<ul style="list-style-type: none"> 1. The notification appears on the home page of the company. 2. Company clicks on the notification.
Exit conditions	<ul style="list-style-type: none"> (a) The company decides to accept the student. (b) The company decides to reject the student. (c) The deadline for accepting/rejecting has passed.

Table 8: UC7 - Company notification of an available student for an internship

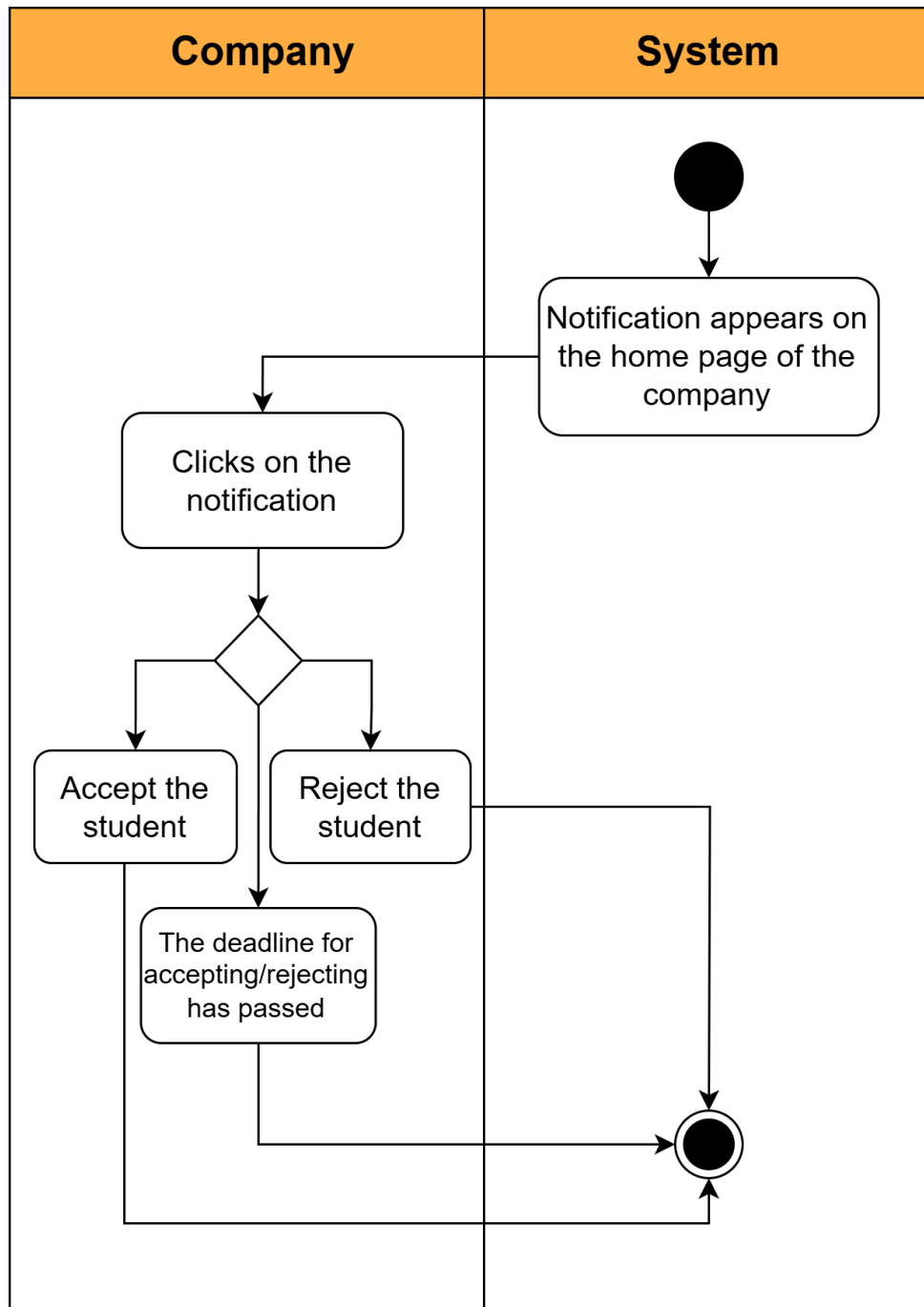


Figure 13: AD7 - Company notification of an available student for an internship

Name	User leaves a feedback
Actors	Student and company
Entry conditions	The internship is halfway through or it's over and the notification appears on the home page of the users
Events flow	<ol style="list-style-type: none"> 1. User clicks on the notification. 2. The system shows the feedback form. 3. User fills in the form.
Exit conditions	User click the button "Save"
Exception	If the user does not fill in the form correctly, an error message is displayed

Table 9: UC8 - User leaves a feedback

Name	User leaves review
Actors	Student and company
Entry conditions	The internship is over and the notification appears on the home page of the users
Events flow	<ol style="list-style-type: none"> 1. User clicks on the notification. 2. The system shows the review form. 3. User fills in the form.
Exit conditions	User click the button "Save"
Exception	If the user does not fill in the form correctly, an error message is displayed

Table 10: UC9 - User leaves review

Name	User leaves a complaint
Actors	Student and company
Entry conditions	The internship has started and the notification appears on the home page of the users
Events flow	<ol style="list-style-type: none"> 1. User clicks on the notification. 2. The system shows the complaint form. 3. User fills in the form.
Exit conditions	User click the button "Save"
Exception	If the user does not fill in the form correctly, an error message is displayed

Table 11: UC10 - User leaves a complaint

Only one activity diagram is used for the three previous UCs, as they are very similar to each other.

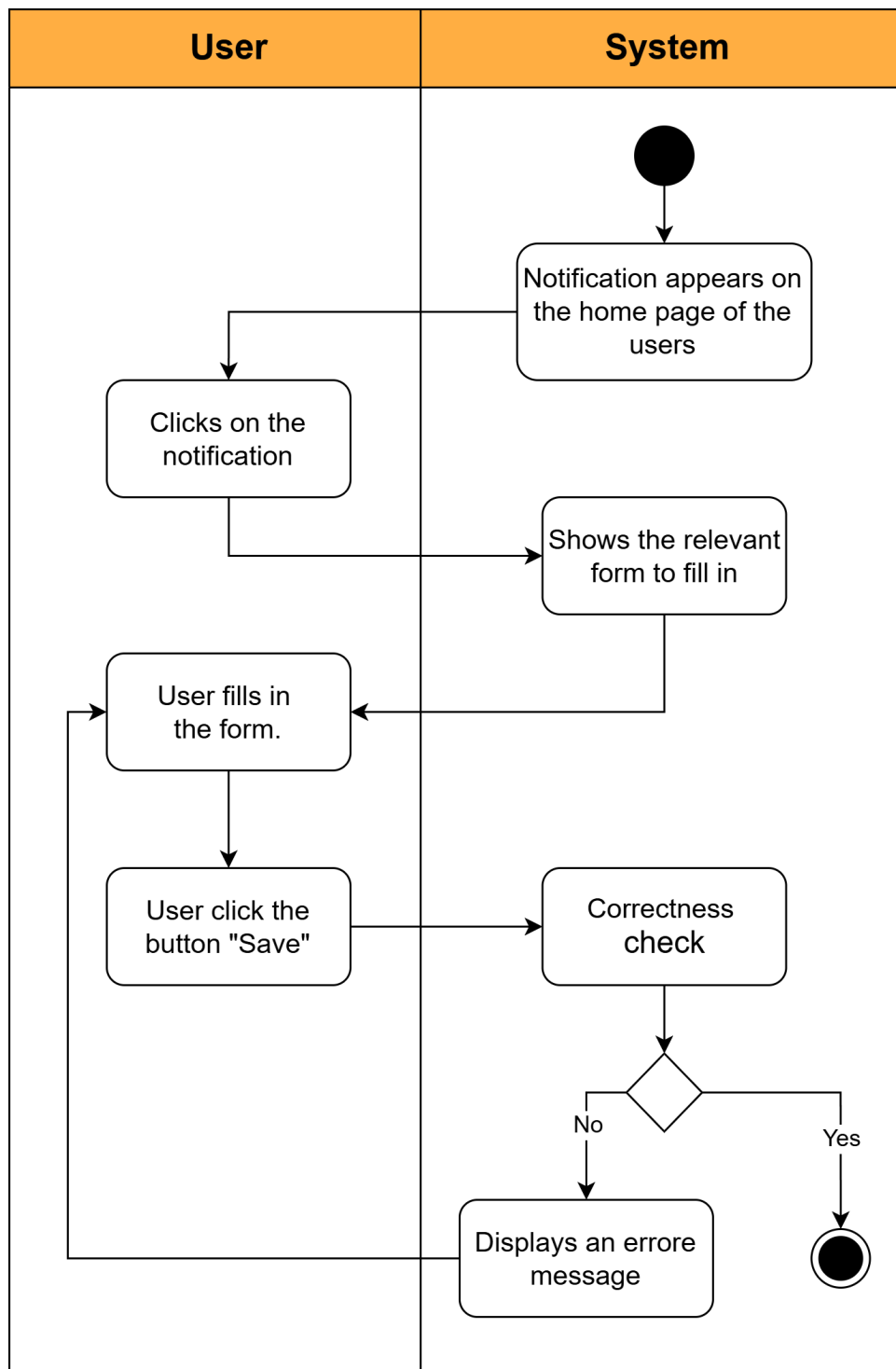


Figure 14: AD8-9-10 - User leaves...

3.2.2 Requirements

R1	The S&C platform allows users to register
R2	The S&C platform allows users to login using their credential
R3	The S&C platform allows users to manage their data and modify them
R4	The S&C platform allows companies to insert an internship and the relative interview form
R5	The S&C platform allows users to see others public profiles
R6	The S&C platform should provide the "search internship" functionality to students
R7	The S&C platform should use the recommendation system to find a match between student and companies
R8	The S&C platform should notify users when a match is found
R9	The S&C platform should allows student to send the request for a internship
R10	The S&C platform should notify companies when a student requests for a internship
R11	The S&C platform should allows users to accept or not the match made by the platform
R12	The S&C platform should allows users to terminate the selection process with a rejection
R13	The S&C platform should send to the students a form related to the chosen internship
R14	The S&C platform checks all the deadlines
R15	The S&C platform should allows users to leave feedback in the middle and at the end of internship
R16	The S&C platform should allows users to leave complaints during the internship
R17	The S&C platform should allows users to leave a review after the end of the internship

Table 12: Requirements

3.2.3 Mapping on requirements

Goal	Requirements
[G1]: All unregistered students and Companies must be able to subscribe and login to the S&C platform.	R1,R2
[G2]: Students and companies must be able to write their descriptions and preferences.	R3,R4
[G3]: Students must be able to complete their CV.	R3
[G4]: Students must be able to search for an internship.	R5, R6, R10
[G5]: Companies must be able to create internship offers.	R3
[G6]: Student and Company are informed when there is a match between them.	R7,R8,R9, R11
[G7]: Monitoring the execution of the selection process.	R12, R13, R14
[G8]: Monitoring the execution of the internships.	R14
[G9]: Statistics collection.	R14
[G10]: Feedbacks collection.	R15
[G11]: Complaints collection.	R16
[G12]: Reviews collection.	R17
[G13]: Companies rank based on reviews.	R17

Table 13: Mapping on requirements

3.3 Performance Requirements

The main performance indicator for this application should be scalability because a large number of users is expected.

3.4 Design Constraint

3.4.1 Standard Compliance

- The platform includes the full adherence to **GDPR**, which stands as one of the most significant and internationally recognized standards for the protection of personal data and the ensuring of user privacy. The system is committed to handling user data in **GDPR** compliant ways, ensuring transparency in data collection and processing, and adopting appropriate security measures to protect such data.
- To obtain data correctness and protection even at the communication level, the platform should adopt the use of **TCP/IP** together with the application of the **TLS** security protocol.
- The purpose of **S&C** is to gather students and companies from all over the world. So, it is very important to use a time standard, such as **UTC**, to achieve the synchronization of all the users, and the handling of deadlines.

3.4.2 Easy to use

The application should be very user-friendly to allow the vast majority of people to use it.

3.4.3 Hardware limitations

In order to enable an effective use of the platform for as many users as possible, the platform should not require high-level hardware and should work on almost all types of machine.

3.4.4 Any other constraint

S&C platform is intended to welcome students from all over the world. So, it should be necessarily designed completely in English, allowing every student to understand its pages, interfaces, commands, etc.

3.5 Software system attributes

3.5.1 Availability

The system must be available as much as possible to allow the user to benefit from the services when they need them. The system should be available with a minimum value of 99% of time. Who will be more affected by lack of availability are users. In this case, students may not be able to search the internship list and cannot participate in the selection process.

3.5.2 Modularity

The system must be designed in a modular way, both for the client-side and for the server-side. The two kinds of actors will have different interfaces that permit the execution of different functionalities. From the server side, traffic is distributed among several servers and managed through a load balancer server. This solution will also allow the user to use the application during the downtime period needed to maintain the server.

3.5.3 Maintainability

It is very important to ensure that the source code of the system can be easily understood, modified, and improved over time. To achieve this goal, the code should be clear and well documented, making it easy for developers to understand and facilitating maintenance.

3.5.4 Security

To match the **GDPR** compliance, the platform should achieve protection of personal data through an authentication system that involves unique e-mails and strong passwords.

There are some aspects of the platform that are private (updating students data, inserting internships), i.e. they are reserved only for a specific group of users. For this purpose, the platform should provide a keyword-based protection system, capable of generating and managing unique keywords: users are asked to submit the correct keyword to access private contexts.

3.5.5 Portability

Given S&C's scale and reach, it is crucial to ensure its compatibility with a wide range of operating systems, including Windows, MacOS, and Linux, for an effortless deployment.

4 Formal Analysis Using Alloy

4.1 Main Objectives

In the following section, we formally describe the domain in which our system operates and its properties using Alloy. In detail, we formally model components through Signatures and we impose constraints on their behaviors through Facts. Finally, we check that our world is consistent. In order to better and easier analyze our world, we provide some worlds that focus on a specific aspect of the entire environment.

4.2 Static Model

With this model we want to demonstrate the various relationships between the students and the internship during the selection process, starting from the eventual application of the student or the recommendation of the system.

4.2.1 Signatures

Here we have the list of all the signatures acting in the model, each one corresponding to an entity implicated in the system.

```
abstract sig User{}

sig Student extends User{
    ApplyTo: Internship,
    RecommendedTo: Internship
}

sig Company extends User{
    Offer: set Internship
}

sig Internship{
    RecommendedStudents: set Student,
    AppliedStudents: set Student,
    AllowedStudents: set Student,
    RejBefForm: set Student,
    SelectedStudents: set Student,
    RejAfterForm: set Student,
    OwnedBy: one Company,
    ToCompile: one InterviewForm,
}

sig InterviewForm{
    Compile: one Form,
    CompiledBy: set Student
}

sig Form{}
```

4.2.2 Facts

With the fact keyword, Alloy allows to define constraints and properties that must be true in all the possible instances of the model. Here is the list of all the constraints of the designed model.

```
//An interviewForm must be unique and associated to exactly one internship.
fact UniqueInterviewForm {
    all iF: InterviewForm | one iF.~ToCompile
    all f: Form | one f.~Compile
}

//A student that apply to an internship must be binded with it
fact Bivalent_application_1 {
    all s: Student | s.ApplyTo in s.~AppliedStudents
    all i: Internship | i.AppliedStudents in i.~ApplyTo
}

//A student if recommended to an internship must be binded with it
fact Bivalent_recommendation_1 {
    all s: Student | s.RecommendedTo in s.~RecommendedStudents
    all i: Internship | i.RecommendedStudents in i.~RecommendedTo
}

//One internship must be owned by only one company, that is the one who offer it
fact UniqueOwned {
    all i: Internship | one i.~Offer
}

fact OwnershipConsistency {
    all c: Company, i: Internship |
        i in c.Offer implies i.OwnedBy = c
}

//A student can not be both
fact noSelnorRec{
    all s: Student |all i: Internship |
        s !in (i.RecommendedStudents & i.AppliedStudents)
        and s !in (i.AllowedStudents & i.RejBefForm)
        and s !in (i.SelectedStudents & i.RejAfterForm)
        and s !in (i.SelectedStudents & i.RejBefForm)
        and s !in (i.RejBefForm & i.RejAfterForm)
}

//A student which is not applied or recommended to an internship can not be
//allowed or rejected by that internship
fact not_jumping_phases{
    all s: Student |all i: Internship |
        (s !in (i.RecommendedStudents + i.AppliedStudents)) implies
            s !in (i.AllowedStudents + i.RejBefForm)
}
```



```

//A student which is applied or recommended to an internship must be allowed
//to start the selection process
fact Allow_Process{
    all s: Student | all i: Internship |
        (s in i.RecommendedStudents or s in i.AppliedStudents)
        implies ((some i.AllowedStudents or some i.RejBefForm) and
            s in i.AllowedStudents or s in i.RejBefForm)
}

//An allowed student must be selected for the internship or rejected
fact Sel_Process{
    all s: Student | all i: Internship |
        s in i.AllowedStudents
        implies ((some i.SelectedStudents or some i.RejAfterForm)
            and s in i.SelectedStudents or s in i.RejAfterForm)
}

//Every student that has been selected or rejected after the form must have
//compiled the form
fact Obligated_To_compile{
    all s: Student | all i: Internship |
        s in i.(SelectedStudents + RejAfterForm)
        implies s in i.ToCompile.CompiledBy
}

```

4.2.3 Predicates

```

pred show {
    one s, s2: Student | one i, i2: Internship |
        s in i.RecommendedStudents and
        s in i.SelectedStudents and
        s !in i2.AppliedStudents and s !in i2.RecommendedStudents and
        s2 in i2.AppliedStudents and s2 in i2.RejAfterForm and
        s2 !in i.AppliedStudents and s2 !in i.RecommendedStudents
    one c: Company | no c.Offer
    #Company = 2
}
run show for 4

```

In this predicate we show the various relations that can exist between a student and an internship, we have two students that apply for two different internships, and that follows different types of relations.



4.3 Dynamic Model

This model has been created to show the correct evolution of the selection of a student for an internship. Here we focus on one internship for the sake of readability and we have created only one scenario that incorporates all different possible evolutions of the said process.

4.3.1 Signatures

```
abstract sig User{}

sig Student extends User{
  var status : one StudentStatus,
}

var abstract sig StudentStatus{}
var sig NonInterested extends StudentStatus {}
var sig Recommended extends StudentStatus {}
var sig Applied extends StudentStatus {}
var sig Allowed extends StudentStatus {}
var sig RejBefForm extends StudentStatus {}
var sig Selected extends StudentStatus {}
var sig RejAfterForm extends StudentStatus {}

sig Company extends User{
  var Offer: set Internship
}

sig Internship{
  var ProposedStudents: set Student,
  var InternshipStatus: one ISStatus,
  OwnedBy: one Company,
  var InterviewToCompile: one InterviewForm,
  var MiddleFeedToCompile: one MiddleFeedForm,
  var EndFeedToCompile: one EndFeedForm,
  var ReviewToCompile: one ReviewForm
}

var abstract sig ISStatus{}
var sig Registration extends ISStatus {}
var sig FormCompiling extends ISStatus {}
var sig SelectionPhase extends ISStatus {}
var sig ISStarted extends ISStatus {}
var sig ISMiddle extends ISStatus {}
var sig ISFinished extends ISStatus {}

sig InterviewForm{
  var InterviewCompiledBy: set Student,
  var ICBY_Company: set Company
}
```

```
sig MiddleFeedForm{
  var MiddleFeedCompiledBy: set Student,
  var MFCBY_Company: set Company
}
```

```
sig EndFeedForm{
  var EndFeedCompiledBy: set Student,
  var EFCBY_Company: set Company
}
```

```
sig ReviewForm{
  var ReviewCompiledBy: set Student,
  var RCBY_Company: set Company
}
```

4.3.2 Facts

//An internship is offered and owned by only one company

```
fact OwnershipConsistency {
  always all c: Company, i: Internship |
    i in c.Offer <=> i.OwnedBy = c
}
```

//Every internship phase has only one form to compile

```
fact VariousFormsUnicity{
  always all iF: InterviewForm | one iF.~InterviewToCompile
  always all mF: MiddleFeedForm | one mF.~MiddleFeedToCompile
  always all eF: EndFeedForm | one eF.~EndFeedToCompile
  always all rF: ReviewForm | one rF.~ReviewToCompile
}
```

//When a student apply or is recommended is proposed to the internship from the system

```
fact proposed{
  always all i: Internship, s: Student | i.InternshipStatus = Registration
    and s.status = NonInterested
    implies s !in i.ProposedStudents
  always all i: Internship, s: Student | i.InternshipStatus != Registration
    and s.status != NonInterested
    implies s in i.ProposedStudents
}
```

*//Every Student that has been at least allowed to compilation of the interview form,
//have compiled it*

```
fact Interview_Compiled{
  always all s: Student, i: Internship |
    (s.status = Allowed or s.status = Selected or s.status = RejAfterForm)
    and s in i.ProposedStudents
    implies s in i.InterviewToCompile.InterviewCompiledBy
  always all s: Student, i: Internship |
    (s.status != Allowed and s.status != Selected and s.status != RejAfterForm)
    implies s !in i.InterviewToCompile.InterviewCompiledBy
}
```

```

}
//Every Student that is in an internship, from more than half its duration,
//have compiled the first feedback
fact Half_Work_Done{
    always all s: Student, i: Internship |
        (i.InternshipStatus = ISMiddle or i.InternshipStatus = ISFinished) and
        s.status = Selected and s in i.ProposedStudents
        implies (s in i.MiddleFeedToCompile.MiddleFeedCompiledBy and
            i.OwnedBy in i.MiddleFeedToCompile.MFCBY_Company)
    always all s: Student, i: Internship |
        (i.InternshipStatus != ISMiddle and i.InternshipStatus != ISFinished)
        implies (s !in i.MiddleFeedToCompile.MiddleFeedCompiledBy and
            i.OwnedBy !in i.MiddleFeedToCompile.MFCBY_Company)
}

//This fact states that students who have finished an internship
//compiled all the forms
fact students_who_have_finished{
    always all s: Student, i: Internship | i.InternshipStatus = ISFinished and
        s.status = Selected and s in i.ProposedStudents
    implies (s in i.EndFeedToCompile.EndFeedCompiledBy and
        s in i.ReviewToCompile.ReviewCompiledBy and
        i.OwnedBy in i.EndFeedToCompile.EFCBY_Company and
        i.OwnedBy in i.ReviewToCompile.RCBY_Company)
    always all s: Student, i: Internship | i.InternshipStatus != ISFinished
    implies (s !in i.EndFeedToCompile.EndFeedCompiledBy and
        s !in i.ReviewToCompile.ReviewCompiledBy and
        i.OwnedBy !in i.EndFeedToCompile.EFCBY_Company and
        i.OwnedBy !in i.ReviewToCompile.RCBY_Company)
}

//When a student is in notinterested state and do not
//apply or get recommend to an internship
fact not_in_time_registration{
    always all i: Internship, s: Student |
        i.InternshipStatus != Registration and s.status = NonInterested
    implies always s.status = NonInterested
}
fact init_state{
    all i: Internship | i.InternshipStatus = Registration
    all s: Student | s.status = NonInterested
}

```

```

//Internship states evolution
fact INT{
    always all i: Internship | i.InternshipStatus = Registration
        implies after i.InternshipStatus = FormCompiling
    always all i: Internship | i.InternshipStatus = FormCompiling
        implies after i.InternshipStatus = SelectionPhase
    always all i: Internship | i.InternshipStatus = SelectionPhase
        implies after i.InternshipStatus = ISStarted
    always all i: Internship | i.InternshipStatus = ISStarted
        implies after i.InternshipStatus = ISMiddle
    always all i: Internship | i.InternshipStatus = ISMiddle
        implies after i.InternshipStatus = ISFinished
    always all i: Internship | i.InternshipStatus = ISFinished
        implies after i.InternshipStatus = ISFinished
}

//Students states evolution
fact STU{
    always all s: Student | s.status = NonInterested
        implies after (s.status = Applied or s.status = Recommended
            or s.status = NonInterested)
    always all s: Student | (s.status = Applied or s.status = Recommended)
        implies after (s.status = Allowed or s.status = RejBefForm)
    always all s: Student | s.status = Allowed
        implies after (s.status = Selected or s.status = RejAfterForm)
    always all s: Student | s.status = RejBefForm
        implies after s.status = RejBefForm
    always all s: Student | s.status = RejAfterForm
        implies after s.status = RejAfterForm
    always all s: Student | s.status = Selected
        implies after s.status = Selected
}

```

4.3.3 Predicates

```

pred show{
    one s, s1, s2, s3: Student |
        after s.status = Applied and after after s.status = Allowed
            and after after after s.status = Selected and
        after s1.status = Recommended and after after s1.status = Allowed
            and after after after s1.status = RejAfterForm and
        after s2.status = Recommended and after after s2.status = RejBefForm
            and always s3.status = NonInterested
    #Internship = 1
    #Student = 4
    #Company = 2
}
run show for 7

```

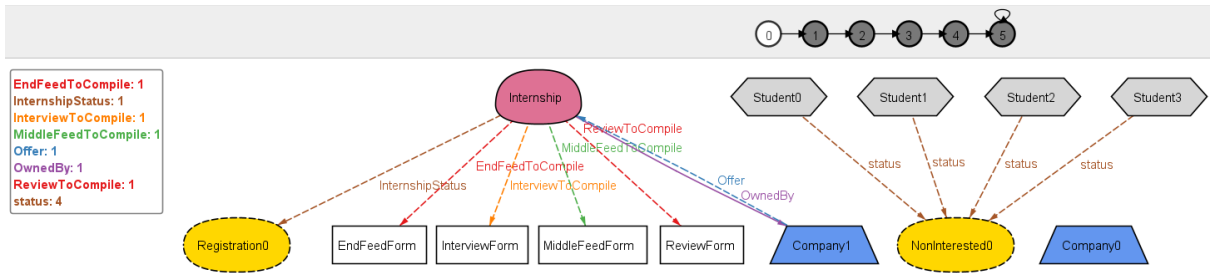


Figure 16: DAM - 0

Here, we have 2 companies, one that does not offer any internship, which is possible on our platform, the other offers an internship, all students at this time are in the state "nonInterested".

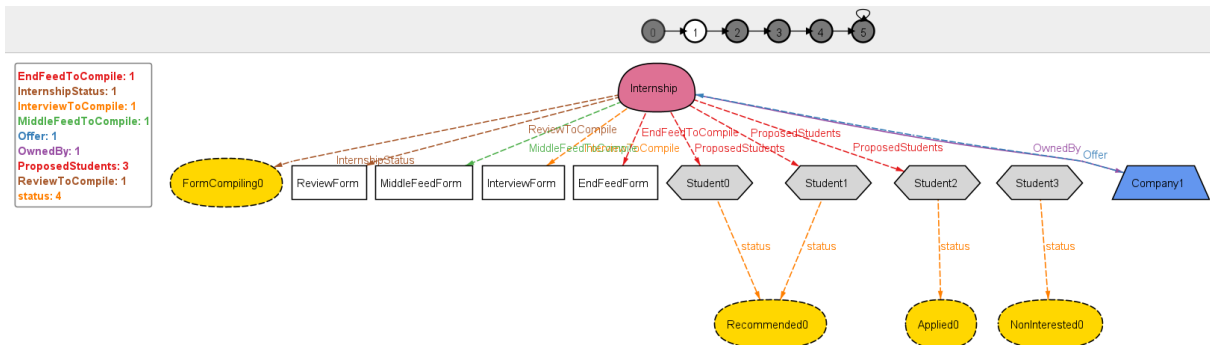


Figure 17: DAM - 1

Here, the students go into diverse states, we omitted how a student goes into the state of recommended or applied because we are not focusing on that in this model, a student can remain, in terms of relationship with one internship: "recommended" (from the system); "applied" (by itself); "nonInterested" (as before).

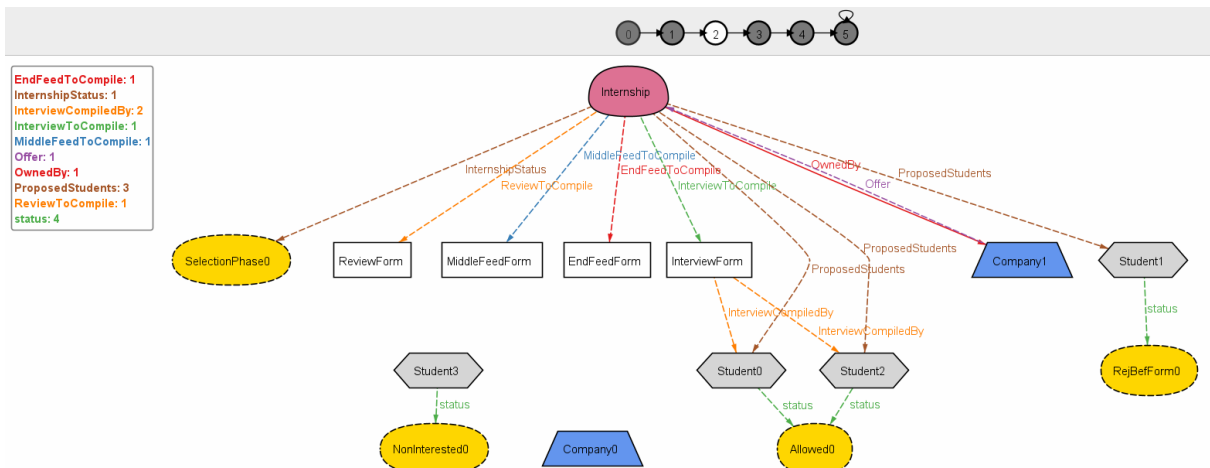


Figure 18: DAM - 2

In moment 2, the students who are bound with the internship, through the "ProposedStudent" relation, can go into two different states, which are "Allowed" or "RejBefForm", that in the first case, as said in the name, allows the student to continue in the process and in the second, the student is rejected, or do not want to pursue that internship.

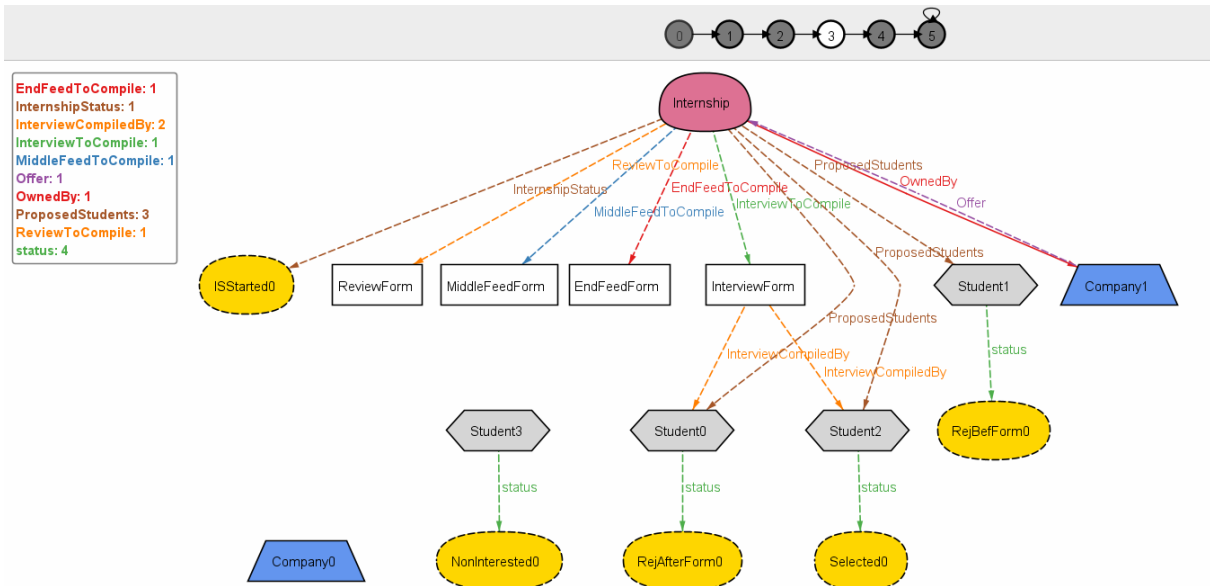


Figure 19: DAM - 3

In moment 3, only the student in "Allowed" state continue their journey through the selection process, are now bounded with the Interview form that they have to compile to be judged from the company. Students can go to the "Selected" state or to the "RejAfterForm".

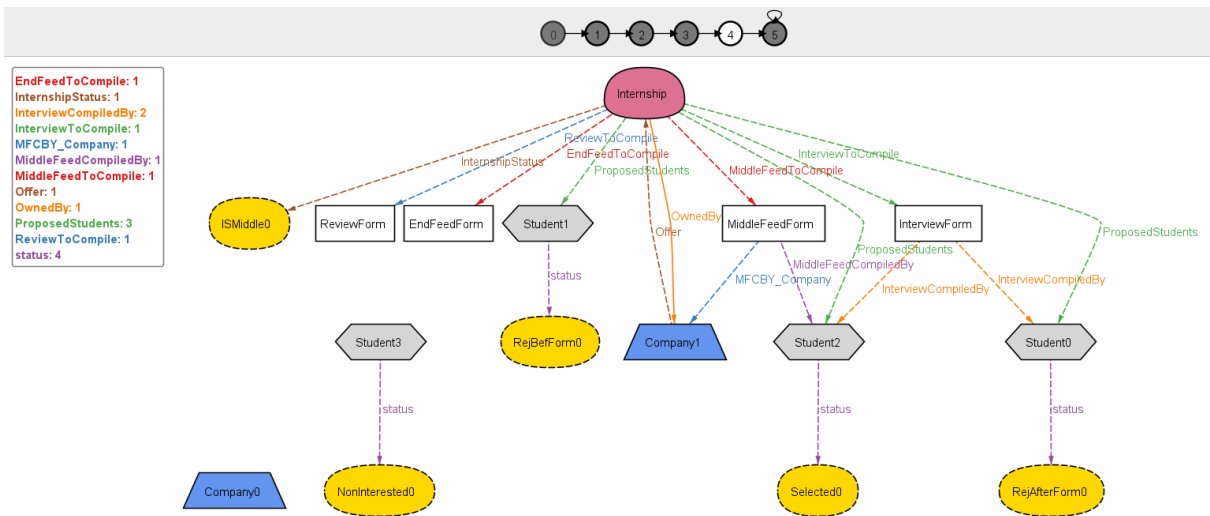


Figure 20: DAM - 4

Once in this state in time, a student and a company are binded for an internship, notice that more than one student can do the same internship at once, but here the focus is on just one for the sake of readability. Both compile at half of the internship the middleFeedForm, that represents the middle feedback collection.

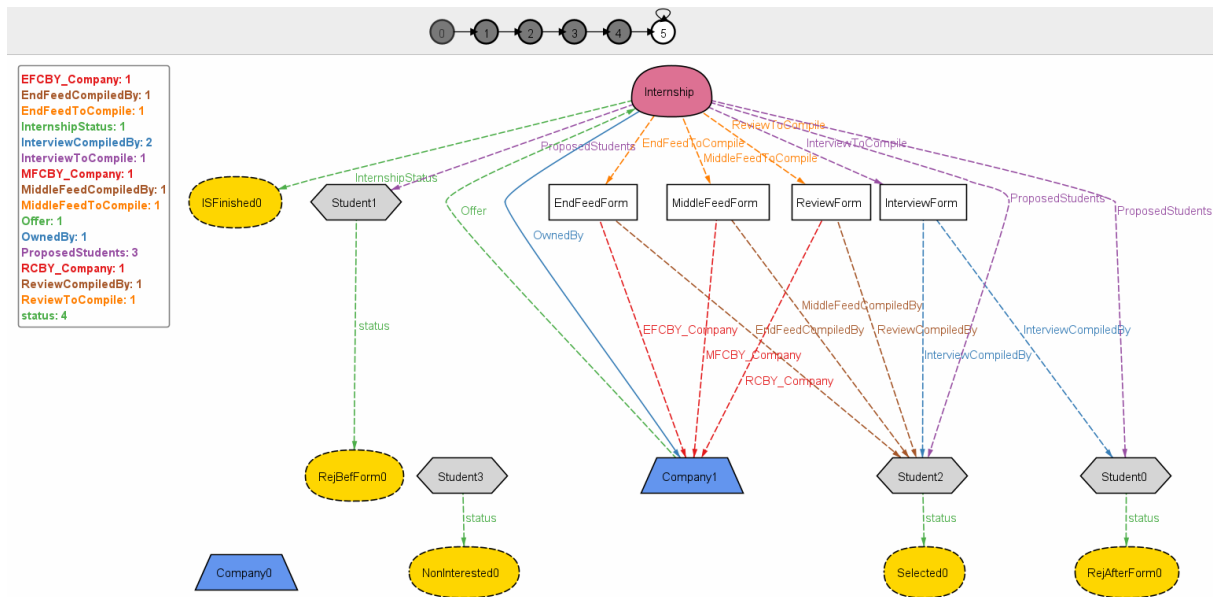


Figure 21: DAM - 5

In the last moment of this execution, the internship is finished, the student and the company compiles the last feedback and the review, and the execution of selection process and internship managements end.

5 Effort Spent

Sections	Cimino	Lorenzetti	Together	Total
Introduction	2	1	10	13
Overall Description	1	1	13	15
Specific Requirements	2,5	5	20,5	28
Formal Analysis using Alloy	2	2	19	23
Total	7,5	9	62.5	79

Table 14: Effort Recap