POLITECNICO

MILANO 1863

Requirement Analysis and Specification Document

Students&Companies project Andrea Bellani Alessandro Capellino

Andrea Bellani, Alessandro Capellino

December 20, 2024

Politecnico di Milano

Contents

List of Figures

List of Tables

Introduction 1

1.1 Purpose

During their university studies, in order to start entering the workforce, a student might decide to apply for an internship related to their field of study. Similarly, companies offering internships may be interested in finding students that are adequate for them. To facilitate the matching between students and companies, a new platform called *Students and Companies* (S&C) is to be developed. S&C allows companies to look for suitable students by publish internship advice on the platform, while students can look for internships that interest them. Moreover, the platform implements recommendation mechanism to help student and companies to find each other. Once the contact is established, S&C can provide support to the students selection process and in monitoring the status of outgoing internships.

1.1.1 Goals

The main goals of the system are:

[G1] students and companies establish contacts for doing internships;

[G2] internships selections can be monitored and supported by the system;

[G3] ongoing internships can be monitored from the system.

1.2 Scope

In this section, we are identifying the S&C domain. In particular, there are two main users categories that interact with the system: *Companies* and *Students*. The companies publish announcements about the internships they want to offer where they specify *projects* that will be carried out and the *terms* of the offer. The system itself informs the companies about the availability of students who may be suitable for their internships (based on their profile).

Students, on the other hand, may use the platform to look for internships and S&C can also notify them if there are new internships that could meet their interests, but they can still independently search through all the available internships.

Once a *contact* is established and accepted by the two parties, the student selection process begins. At this point the company defines selection steps and schedules the interviews for each student. Once the selection is over, the system collects feedback and suggestions from both students and companies.

Finally, both students and companies can monitor the progress of the internships by providing information on its development and any issues that may arise. At the end of the internship, both students and companies can provide feedback about the internship; this feedback are collected by the system and, through some statistical analysis on them, used to improve the recommendation mechanism.

1.2.1 Phenomena

World Phenomena

[WP1] Students create their own CV, including their studies, work experiences, skills, and attitudes

[WP2] Company decides to offer a new internship for students who want to gain experience

[WP3] During the selection process, the company conducts an interview with the student through an oral exam

[WP4] Students selected for a particular internship begin working on the related projects

[WP5] Users of the platform (students and companies) encounter issues with the internship projects they are working on

Shared Phenomena

World-controlled Shared Phenomena

[SP1] Companies publish new internship advice [SP2] Students search for all the available internships on the platform

[SP3] Students search for all the companies registered on the platform

[SP4] Students search for specific internships using the search bar

[SP5] Students apply for an internship

[SP6] Companies accept/decline the applications of some students for their internships

[SP7] Companies offer internship proposals to specific students

[SP8] Students accept/decline the internship application offer

[SP9] Companies configure the selection process for their internships

[SP10] Companies enter the evaluations of the students' interview answers

[SP11] Students are selected/rejected by the selection process's company

[SP12] Users (students and companies) provide feedback and suggestions about the internship and the selection processes

[SP13] Users (students and companies) monitor the status of ongoing internships, providing complaints, problems and information about them

Machine-controlled Shared Phenomena

[SP14] The systems notifies students when an internship that might interest them become available

[SP15] The system notifies companies about the availability of interesting students regarding their internships

[SP16] The system notifies students if the companies accept their application for the internships

[SP17] The system notifies companies if the students accept their application proposal for the internships

[SP18] The system notifies students about the interview dates

[SP19] The system notifies students their selection final result

[SP20] The system perform recommendation analysis

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

- **internship advice** : a call for application related to an internship that will be offered by a company;
- **recommendation** : the mechanism related to the fact that the system both informs students whether new internship advice that might interest them are published and notifies companies of the presence of students that might be suitable for their internships;
- **contact**: a student and a company establish a contact when the company accepts the application of the student or the student accepts the proposal of application of the company
- **project** (of an internship advice) : the definition of the application domain, the set of tasks to be performed and the set of the most relevant adopted technologies (if any) for an internship;
- **terms** (of an internship advice) : the set of benefits offered by an internship (e.g. paid/not paid, training, lunch voucher...);
- **selection process** : each internship advice is followed by a sequence of selection steps.
- **platform**: synonym of system

1.3.2 Acronyms

- S&C: Students&Companies, the name of the platform;
- UML: Unified Modeling Language;
- CV: Curriculum Vitae.

1.3.3 Abbreviations

- Gn: Goal number n;
- Rn: Requirement number n;
- Dn: Domain assumption number n;
- WPn: World Phenomena number n;
- SPn: Shared Phenomena number n;
- UC: Use Case.

1.4 Revision history

First version	
22/12/2024	

1.5 Reference documents

The Documents used to deliver the RASD document are the following:

- the Specification of RASD and DD assignment of Software Engineering 2;
- the class slides on WeBeep, in particular slides on RE (requirement engineering), scenarios and Use Cases and UML diagrams;

1.6 Document structure

1. **Introduction**: this section provides a brief introduction to the purpose of the platform to be developed, S&C in this case, focusing in particular on the most important goals which the system has to achieve and on the various phenomena identified;
2. **Overall Description** : an high-level (conceptual) description of the system functionalities explained through scenarios, high-level class diagram, product functions and domain assumption;
3. **Specified Requirements** : the detailed requirements analysis. In this section is detailed the entire requirement set (functional and non-functional), the most relevant use-cases (including sequence diagrams that formalize them) and the design constraints that must be stated also at the requirement level;
4. **Formal Analysis** : formal modeling and simulation of a simplified model of the system, in order to formally prove the correctness of the (possibly) foremost requirements (using Alloy 6);
5. **Effort Spent**: report of the time spent by any group member in any document section;
6. **References**: list of software and documents used to develop the document.

Overall description 2

2.1 Product perspective

2.1.1 Scenarios

Student signs up to S&C

Student Bob enters in the system for the first time. On the homepage, he first clicks the *Registration button* and then the *Student Registration button*. To register, Bob fills out a form providing its institutional e-mail (bob.johnson@mail.polimi.it) and password (which will be used for future logins), a brief description of his academic background and specifies whether he would like to take part to the recommendation analysis. Finally, Bob uploads his CV by clicking the *Upload CV button*. Now Bob is registered and can search for internships that interest him.

Company signs up to S&C

The company FinestraMI enters the system for the first time. On the homepage, it first clicks the *Registration button* and then the *Company Registration button*. To register, the company fills out a form providing its name, a brief description of its area of expertise and its business area (the market where it operates) and finally its corporate e-mail (info@finestrami.it) and password (which will be used for future logins). FinestraMI also specifies, by selecting the appropriate option, whether it wants take part into the recommendation analysis. Now, FinestraMI is registered and can publish its internships advice.

Company publishes an internship offer

The company FinestraMI enters in the system; on the homepage, it clicks the *Login button*. Once logged in, FinestraMI accesses the *Publish New Internship section*. A new internship advice is added by filling out a form where the following information is provided:

- ”Window restore” (the intership title);
- ”The aim of this internship is to give to student to opportunity to repair office windows and...” (a brief description);
- ”third year bachelor students...” (experience required);
- ”not suffering from dizziness” (desired skills);
- ”1. coordination of glass disposal; 2. ...” (main activities the internship involves);
- ”no paid, canteen tickets available” (terms of the internship);
- ”22/11/2024” (advice deadline).
- max 42 applications (max number of applications)

Now the internship advice is visible to students registered on the platform (and also to FinestraMI).

Student proactively searches for an internship

Students Bob, Alice and Micheal access to the system by clicking ”Login”. Each one of them wants to find an internship to apply but each one of them has a different idea of what and where he/she would like to do/be:

- Bob is really interested on doing practice on an handwork but he neither knows a name of a company nor knows which kind of handwork apply for so, he goes to the *View Internships section*, where he can see all the published internships, listed from the most recent to the least recent. The most recent one is ”Window restore” by FinestraMI, then he selects it;

- Alice has not already decided the kind of internship she wants to apply for but knows many names of companies that operate near her home and so she prefers to go to the *View Companies section*, where she can see all the registered companies and all the internships published by each company. Then she recognized FinestraMI and since she knows that it is expanding, she decides to select it. ”Window restore” is the only available advice of FinestraMI but she select it anyways;
- Micheal is looking forward to do an internship related to windows restoration, so he uses the search bar to insert ”windows restoration” and selects the option ”only paid internships”, but no internship are found. Then he removes the option and find the internship of FinestraMI. Since it is the only left, he selects it.

Student receive a notification about a new internship

The company CancellaMI (previously registered to the platform) publishes a new internship related to railings maintenance then, Student Bob, who has chosen to be notified by the system when new internships that might be of interest are published, receives an email informing it that a new intership related to his studies is available, since it stated in his CV that after the internship at FinestraMI he became passionate of railings. Bob then logs into the platform and, by going to the *Notification section*, can view the internships offer in more detail.

Company receives a notification about new possibly interested students

Company FinestraMI, which has chosen to be notified by the system, receives an email informing it that new students are appealing for its intership ”Window Restore” (based on their CVs). FinestraMI then logs into the platform, goes into the *Internship section*, clicks on *Windows restore internship* and by going to the *Notification section* can view the students’profiles and their CVs in more detail.

Student applies for an internship

Student Bob wants to apply for the internship ”Windows restore”. To do so, they log into the system, access the page for ”Windows restore” internship and click the *Apply button*. Automatically, the system will send a notification to FinestraMI (the company offering the internship) to inform it that Bob has applied

The company accepts the application of a student

Company FinestraMI receives the email regarding student Bob’s application for the internship ”Window Restore”. FinestraMI then logs into the platform, navigates to the *Internships section*, select the *Window Restore Internship*, goes to the *Notification section* and clicks the *Accept Application button* to approve Bob’s application.

The company proposes to a student to apply for one of its internships

The company FinestraMI consults its list of recommended students for Window Restore and send a proposal to Bob Jones (by clicking on the dedicated button). Soon after the system sends a notification of the proposal to Bob.

Student accepts an internship proposal

Bob receives an email regarding Window Restore proposal (of the company FinestraMI), then Bob logs into the platform, navigates to the notification section, open the notification regarding the proposal and clicks on the accept button.

The application deadline expires and the selection process is configured

The administrator of the company FinestraMI notices that the application deadline for the internship advice ”Window Restore” (which was previously published on the platform) is now expired and selection process for that internship has not configured yet, so he goes to the designated page and configures:

- two steps (the selection process will be made up of two steps);
- a set of metrics to evaluate students (”manual skills” and ”knowledge of materials” in this case);
- each step is configured as a questionnaire with a series of questions for the students, in this case in particular:

1. first step is test of both open and closed questions regarding knowledge of materials. For closed questions, the platform is also able to automatically check if they are corrected or not (and so, for each closed question, also the scores to assign to each possible answer are inserted into the system). Open questions will be evaluated manually by the company;
2. second step is an oral exam. Since there are no predefined questions for this step, the company only inserts into the system one open question called ”oral exam”, scores will be inserted by the company at the end of the exam.

- for each step and for each candidate, the company chooses also the date in which it provides the questionnaire to the candidate.

The selection process runs

For the internship advice ”Window restore”, the company FinestraMI received three applications: Bob, Alice and Micheal. FinestraMI is planning to accept only one student at time, therefore it chooses to first call Micheal for the first step, since his curriculum impressed more the company. On Micheal is called and the questionnaire is given to him. His answers are evaluated (automatically for the closed ones and manually for the opened ones) and gets an overall score of 99 out of 100: the company decides to select him, discards Bob’s application and leaves suspended the call for Alice. The company sets for Bob and Micheal the right message and the platform notifies them.

User provides a feedback at the end of the selection process

Micheal has just received the selection results for his application for the ”Window restore” internship of FinestraMI. Attached to it, the system provides him an optional questionnaire where it asks to Micheal to evaluate his experience of the selection (questions are quite standard, such as ”was the company on time with the interview appointments?”, ”did the questions related to the required skills” e.c.c.). Since it is not compulsory, Micheal does not compile it. On the other side, once the entire selection process of FinestraMI is closed, FinestraMI receives from the system a questionnaire to evaluate its experience (questions mainly concern the preparation level of the candidates, such as ”was the number of students with the required skills below average?”). Then the company compiles it from the system.

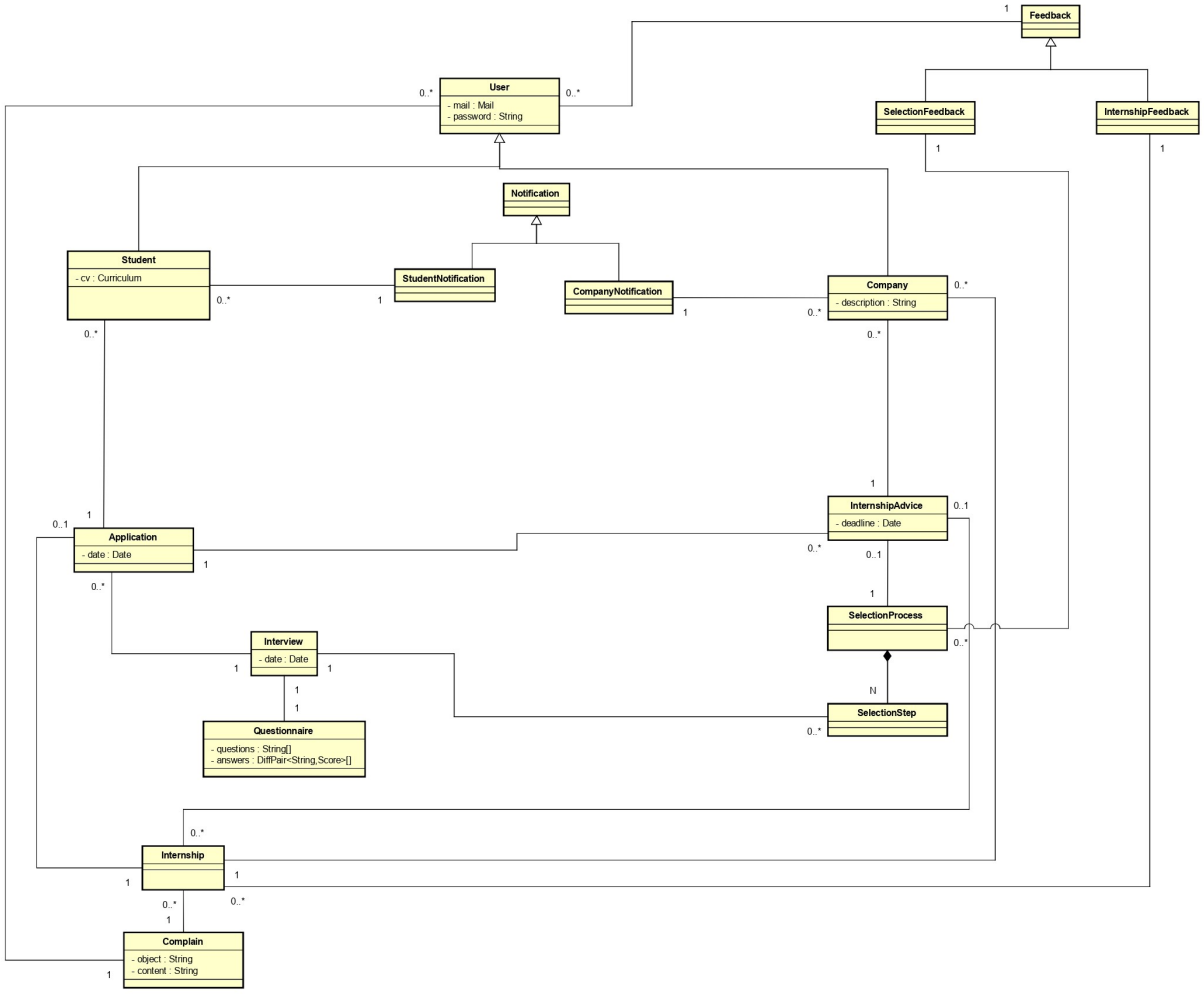
User reports a complaint on one of the internship is currently doing

Today, Alice who is currently enrolled in the internships at the company WeWorkGreat had a problem with the task that was given to her, she asks the helpdesk of the company where she is performing the internship and they ask her to upload a video on the company file sharing platform to show the situation. Alice notices that she can’t upload the video because the maximum uploading size for students is to 10 MB, then she opens Students&Companies and writes a compliant that states that the file sharing system of WeWorkGreat is only of 10 MB.

User provides a feedback at the conclusion of an internship

Alice has just finished the internship at WeWorkGreat, an non-compulsory questionnaire is given to her with some general questions related to her experience at the company (e.g. ”did the company respect the terms listed in the advice?” e.c.c.). Since it is not compulsory, Alice decides to not compile it.

2.1.2 High-level class diagram



This figure represent the domain class diagram of the system. It represents the fact that the system connects students and companies, facilitating the management of applications, selection processes, interviews, internships, and communication between the parties. It provides tools for sending notifications, collecting feedback, and managing questionnaire and complaints.

The system's users are modeled through the *User* class, which represents both Students (class *Student*) and Companies (class *Company*). Each user has basic attributes (email and password) and can receive personalized notifications (class *Notification*, with subtypes for students and companies).

Companies can publish internships advice (class *InternshipAdvice*); students can apply for an internship but also companies can send proposal of application to the students. This mechanism of applications is managed by the *Application* class. Each application born with one and only one of the boolean flags to true, corresponding to the kind of user that unleashed that application. Once the other user accepts the proposal of the counterpart, its flag is put to *true*; at the end, students that will enroll the selection process are the only related to applications with the 2 flags true.

Each application can go through a *SelectionProcess*, divided into several *SelectionStep*. During the selection process, the student may undergo an interview (*Interview* class) and his answers are collected by the system through questionnaires (*Questionnaire* class).

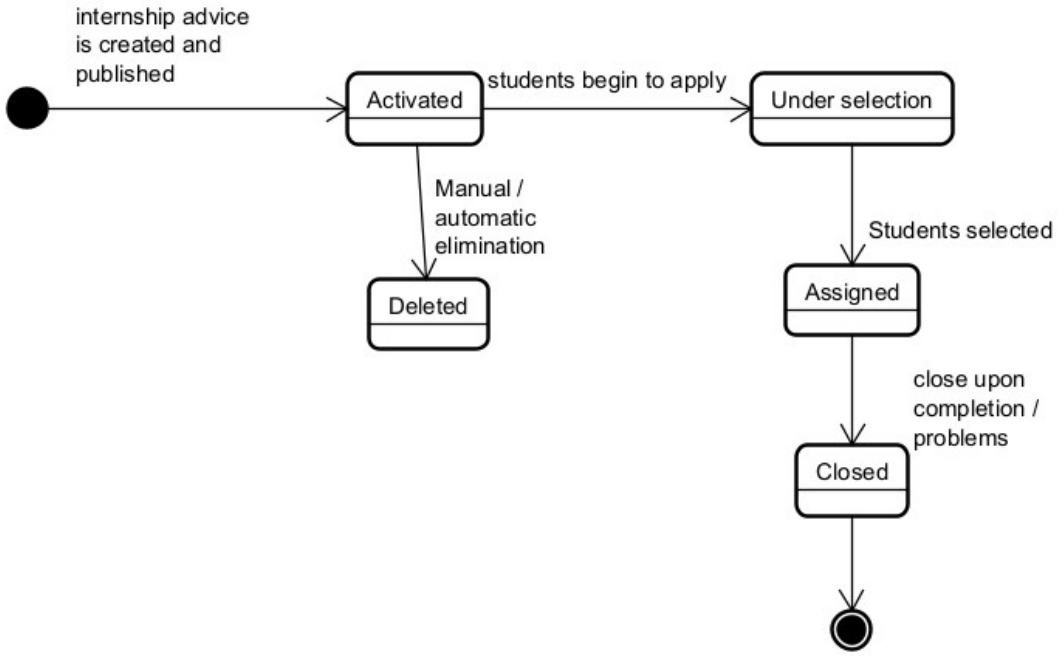
Internship advice is associated with a concrete Internsihp (class *Internship*) offered by companies; Internships are linked to feedback (*InternshipFeedback*) and complaints (*Complain*). The system supports the collection of feedback in two forms:

- *SelectionFeedback*: feedback on the selection processes.
- InternshipFeedback: feedback related to internships

2.1.3 State Diagrams

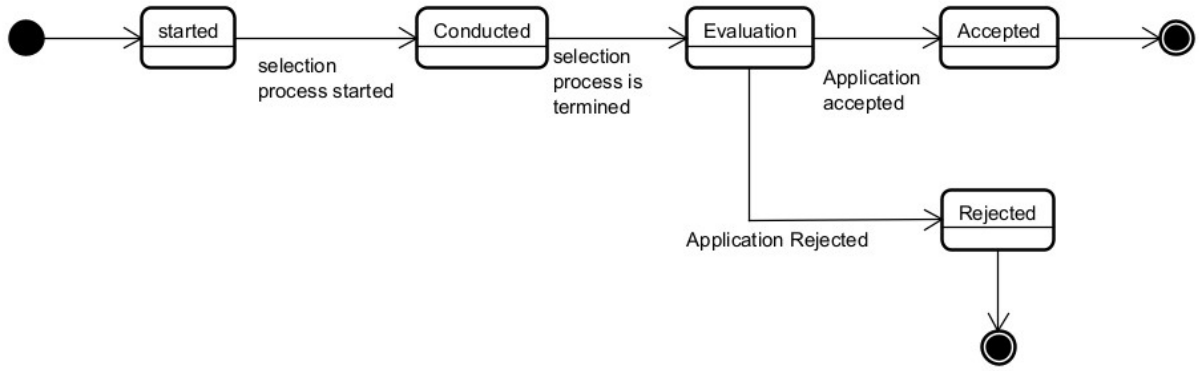
In this subsection, the most relevant state diagrams are presented in order to better understand the system's evolution through its different phases. We focus in particular on the representation of an internship's state and the evolution of a student's application.

State of an internship



In this state diagram, each state represents the status an internship can have. *Activated* is the state where the internship is published and visible on the platform (in the form of an internship advice); *Under Selection* is the state where the company is conducting the selection process;*Assigned* is the state where the internship has been assigned to specific students; *Closed* is the state where the internship has ended (either because it finished or was interrupted); *Deleted* is the state where the internship is removed from the system (the company deletes the internship advice or the selection process is ended).

State of an application for an internship



In this state diagram, the evolution of a student's application for an internship is described. The *Started* state refers to a student's application for an internship. The *Conducted* state relates to the evolution of the selection process (open questions, closed questions, interviews, questionnaire completion, etc.). The *Evaluation* state is where the company assesses the student's application. From this state, two subsequent states are possible: *Accepted* if the application is accepted, or *Rejected* if it is declined.

2.2 Product functions

Sign-up and login

This functionality permits to both students and companies to set up their personal profile on the platform and accessing to the latter via their personal information (e-mail and password). Students profiles specify basic information regarding students interests and include their CV while companies profile include all the information that may help student in understanding companies vision and business area.

Internship proposal management

Companies can post advice for internship they are going to offer and students can proactively search for them. In this case, a student that wants to apply for an internship sends a request to the company and it decides or not to enroll the student into the selection process. Moreover, thanks to the recommendation feature, companies receives profiles of possibly interested students and students receive profiles of companies that offer internship possibly related to their interests. Therefore, a companies can also suggest to a student to apply for an internship selection and students can accept or decline this proposal.

Recommendation mechanism

This functionality aims to facilitate the establishment of a contact between students and companies. By means of recommendation analysis, the system is able to propose to students internships advice that may interest them and to companies students that may be interested to their internships. This analysis are supported by companies and students profiles (CVs in particular, for students) and not compulsory feedback collected at the end of selections processes or internships.

Selection process management

A selection process can be supported by the system from the advice deadline to its finalization. In particular, companies can use the system to define process steps, schedule interviews, correct closed questions and compare students results each other (by defining their metrics).

Internship monitoring

Students that are currently involved in active internships can use the system to post complains and monitor the current status of them. On the other side, also companies can also post complains on internships they are currently providing (and of course they can also monitor their status).

Notifications management

Each message that is sent to companies or students (e.g. selection results, internship proposal, interview dates) are also sent to profile e-mail addresses (a short version of them). From the system, each message can be read entirely.

2.3 User characteristics

there are mainly two types of user that interact with the platform: Student and Companies.

2.3.1 Student

The student, once logged into the platform, is able to proactively search for internship advice that may interest them. Once the internship of interest has been selected, they can apply for an interview directly through the platform. Additionally, the student receives a list of possible internships that may suit them, based on their studies/CV, via the aforementioned recommendation mechanism. Furthermore, the student may also receive a proposal to apply directly from the company offering the internship. Once they have completed a selection process, the student will receive the outcome directly through the platform.

2.3.2 Company

The company, once logged into the platform, can publish internship announcements it wants to offer. The system will periodically send the company a list of potential students who may be suitable for their internships, based on their CVs, to whom it can directly send a proposal to apply for one of its internships. Conversely, the company can also accept or reject a student's application. For a given internship, once the application period has ended, the company can schedule the selection process for each candidate directly through the system. During the selection phase, the company also collects all candidate responses using the system.Finally, the company will provide the outcome of the selection to each student directly through the platform.

2.3.3 common characteristics

Both users can also provide feedback on the selection process as well as at the end of the internship. This feedback is collected by the system to improve the recommendation process. Additionally, both users can monitor the status of an outgoing internship and provide information or complaints about it.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain Assumption

the following assumption are made for domain. They are properties or condition that the system will take for granted. They must be checked to ensure a correct platform behavior.

[D1] Users must have a reliable internet connection

[D2] Information contained in the CVs are truthful

[D3] Company fill the form about internship advice accordingly to their business decisions

[D4] Company correctly enters the answers provided by the student during the interview into the system

[D5] When users registers, they register with an active email address that belongs to them

[D6] Uploaded CVs are written in EuroPass format

2.4.2 Dependencies

- The system will integrate with email system to send notifications to Users

2.4.3 Constraints

- The system shall be compliant to local laws and regulations, in particular users data should be treated according to the GDPR. This means that users should be always able to request their data
- The data collected for matching (feedback, ...) are sufficiently detailed to support effective statistical analysis.
- The number of students and companies using the platform will be manageable by the system without compromising its performance
- To better protect the users' sensitive information their data should be encrypted
- the mail used by the student to register is able to digitally sign the CV

Specific requirements 3

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

There are no specific hardware interfaces required, except for the devices used to access the platform. Both students and companies need a hardware device (such as a computer) with an internet connection to access S&C.

3.1.3 Software Interfaces

To manage the sending of all notifications (recommendation notifications, notifications for selection results, etc.) between two or more users, the *mail service* must be used. In fact, both students and companies are required to provide their email address (university/company email) during registration. Every time the system sends a notification to a user, they will receive an email informing them of the received notification. To view the content of the notification in more detail, the user must log into the system and go to the appropriate section of the platform.

To do this, the system will send a request to the mail service, which will then forward the desired email to the specific user.

3.1.4 Communication Interfaces

Both students and companies use the internet to access the platform; in particular, companies require an internet connection for all the functionalities they can perform through the platform (posting internship advice, configuring the selection process, etc.). Similarly, students also need an internet connection for their functionalities (searching for internships, applying, etc.).

due to the confidentiality of exchanged data, a secure communication mean is required, such as HTTPS or VPN

3.2 Functional Requirements

Here are described all the functional requirements of the S&C platform:

[R00000] when a notification of a user is generated, the user receives it on its mailbox (in a more concise version) and can consult it on its notification section

[R10101] the system allows students to sign up to the platform with their institutional mails

[R10102] the system allows a student to set up whether he/she wants to take part into the recommendation

[R10103] the system allows students upload their CV to the platform

[R10104] the system allows students to publish on their profile a brief description of themselves

[R10105] when a CV is uploaded,the system verifies if it is digitally signed by the profile mail

[R10106] the system allows students to log in into the system by providing the registration mail and the chosen password

[R10107] the system allows students to change their profile information (including the CV) and their access information

[R10108] when a student registers, the system extracts from the CV name and surname to create the profile

[R10201] the system allows companies to sign up to the platform with their company address

[R10202] the system allows companies to insert the main information regarding their business area and area of expertise

[R10203] the system allows a company to set up if it wants to take part into the recommendation analysis

[R10204] the system allows companies to log in into the system by providing the registration mail and the chosen password

[R10205] the system allows companies to change their profile information and their access information

[R10301] the system allows companies to publish internship advice where they specify the main information regarding the internship (brief description, experience required, desired skills, main activities involved and the terms) and the submission deadline

[R10302] the system allows companies to delete internship advice which deadline is not expired yet

[R10401] the system allows students to search internships advice by name The system shall act as a search engine to present also the names of the advice that are similar to the searched one

[R10402] the system allows students to search companies by name (and also to see the complete list of registered companies) and then access to their profile

[R10403] the system allows students to filter the results they searched (e.g. ”only paid internships”, ”only companies located in Lombardy”)

[R10404] the system allows students to consult the list of all published internship advice, listed from the most recent to the last

[R10501] when the system recognizes that a new internship advice that might interest a student (that allowed the recommendation option) published, it notifies that student by sending him an e-mail (to its registration address)

[R10601] when the system recognizes that a student has a profile that would fit an internship advice, the company that published the advice is notified (for students and companies that both take part into the recommendation analysis)

[R10701] the system allows students to apply for any internship advice which deadline has not expired

[R10702] when a student applies for an internship, the related company is notified by the system

[R10801] the system allows companies to approve, discard or ignore each application they may receive for one of their published advice

[R10901] when a company opens a student profile, it can propose to him to apply for one of its internships. Then, the students receives a notification

[R11001] the system allows student that received an internship proposal from a company can decide to accept it, discard it or ignore it

[R11002] when a student accept an internship proposal, it is implicitly accepted by the company

[R11101] when a student gets his/her result of the selection, the system provides to him a non-compulsory questionnaire regarding his/her experience (in the context of that selection process)

[R11102] when a company ends a selection process, the system provides to it a non-compulsory questionnaire regarding its experience (in the context of that selection process)

[R11201] when a student concludes an internship, the system provides to him/her a non-compulsory questionnaire regarding him/her experience (in the context of that internship) and in the meantime it provides to the internship company an analogue questionnaire also non-compulsory

[R20101] when the deadline for an internship advice is expired, the system allows the company to set up the selection process by specifying for each step, the relative questionnaire (with metrics for each question) and the date in which provide it to a student (dates may differ between different students)

[R20102] the system includes into a selection process only student that had an accepted application for the relative internship advice

[R20201] the system notifies students for any interview date

[R20202] the system automatically calculates the scores of questionnaire closed answers

[R20203] the system allows companies to manually insert scores for questionnaire open answers

[R20204] the system allows companies to visualize and compare selections scores

[R20205] in any selection phase, the system allows companies to discard a student currently involved in the selection process (discarded students are removed by the selection process)

[R20206] in any selection phase, the system allows companies to accept a student currently involved in the selection process (accepted students are removed by the selection process)

[R20207] the system allows companies to write a personalized message to communicate the result of a selection

[R20208] when a selection result is prepared for a student (with the relative message), it is notified to the student

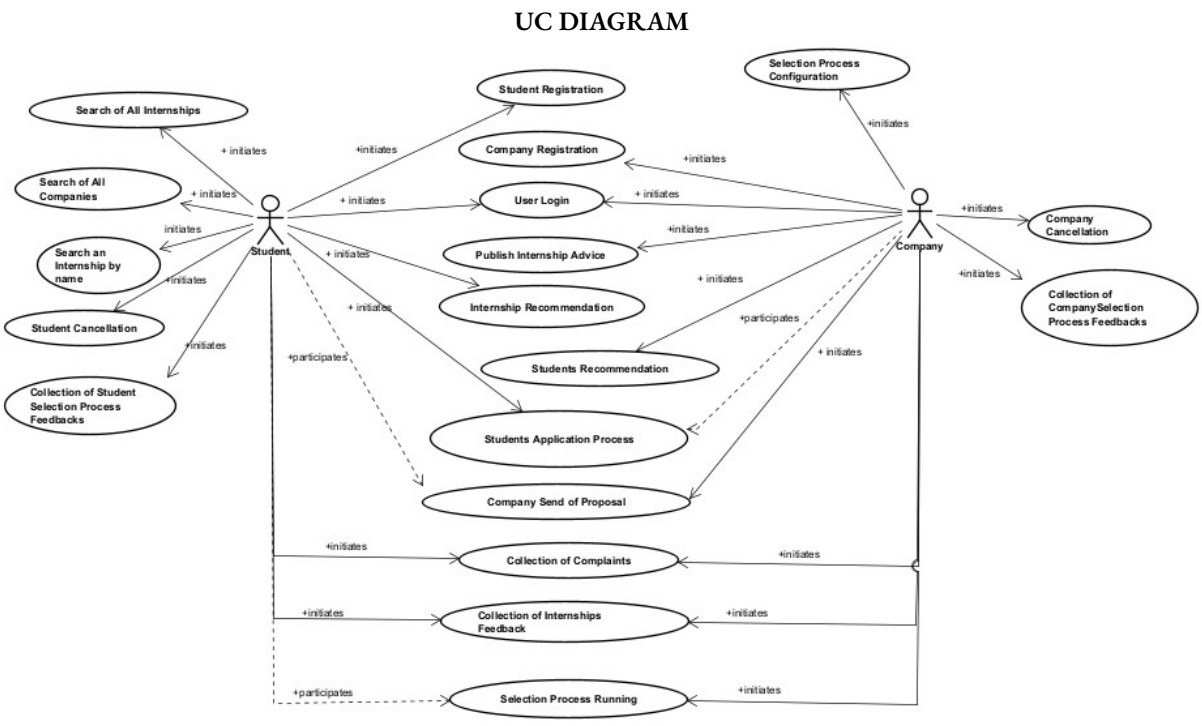
[R20209] when a selected process related to an internship advice has ended, the system deletes the advice

[R30101] the system allows students and companies to consult the internships (ongoing or finished)

[R30102] the system allows students and companies to report complaints (at most 50 words)on the internships they are involved in

[R30103] the system does not allow users different from their creator to consult complains

3.2.1 Use-case diagram



The Use Case (UC) diagram shows, for each defined UC, which actors are involved and, in general, which actor initiates the UC. For UCs that can be performed by both actors (students and companies), the common name *User* is used.

3.2.2 Use-cases

UC1 - Student Registration

Name	Student Registration
Actors	▶ Student
Entry condition	The student has opened the S&C platform for the first time
Event Flow	◻ The student press the button *Registration* ◻ The student press the button *Student Registration* ◻ The system shows a form to compile ◻ The Student compile the form providing his institutional e-mail and password and a brief description on his academic background ◻ The Student flags the *receive notification* box (part of the form) if he wants to take part into the recommendation process ◻ The student press the button *Upload CV* to upload the curriculum (the button is part of the form) ◻ The student press the button *Register* to complete the registration ◻ the system checks the digital signature on the CV
Exit condition	The student has successfully registered to the platform and the system shows the homepage
Exception	◻ There arleady exist a student with those e-mail; the system return to the entry condition showing an error message *Student arleady registered* ◻ the CV is not digitally signed; the system shows an error message informing the student about that

UC2 - Company Registration

Name	Company Registration
Actors	▶ Company
Entry condition	The company has opened the S&C platform for the first time
Event Flow	◻ The company press the button *Registration* ◻ The company press the button *Company Registration* ◻ The system shows a form to compile ◻ The Company compile the form providing its name, a brief description of its area of expertise and business area, its e-mail and password ◻ The Company flags the "receive notification" box (part of the form) if it wants to take part into the recommendation process ◻ The company press the button *Register* to complete the registration
Exit condition	The company has successfully registered to the platform and the system shows the homepage
Exception	There arleady exist a company with those e-mail; the system return to the entry condition showing an error message *Company arleady registered*

UC3 - User Login

Name	User Login
Actors	▶ User
Entry condition	The User has opened the S&C platform
Event Flow	◻ The user press the button *Login* ◻ The system shows the Login page ◻ The user logged to the platform inserting e-mail and password ◻ The user press the button *Log*
Exit condition	the user has successfully logged to the platform and the homepage is shown
Exception	1. The user has not yet registered to the platform; the system will return to the registration page 2. the credentials are not correct; the system shows an error message and the system return to the login page

UC4 - Publish Internship Advice

Name	Publish Internship Advice
Actors	▶ Company
Entry condition	The company is logged to S&C platform and it wants to publish a new internship advice
Event Flow	◻ The company goes to the *Publish New Internship* section ◻ The company press the button *Add New Internship Advice* ◻ The system shows a form to compile ◻ The company compile the form by providing: the title, the subject, a brief description of the internship, experience required and desired skills, the main activities the internship involved, the terms of the offer, the advice deadline and the max number of applications ◻ The company press the button *Publish*
Exit condition	The company has successfully published a new internship advice and the system return to the homepage
Exception	The company doesn't fill all the fields of the form; the system will warn the company

UC5 - Search of all Internships

Name	Search of All Internships
Actors	▶ Student
Entry condition	The student is logged to S&C platform and he wants to apply for an internship
Event Flow	□ The student goes to the *View Internships* section □ The system shows the page with a list of all the published internships advice, listed from the most recent to the last re-cent □ The student select an intern-ship advice who interested them
Exit condition	the system shows the page with all the details about the internship
Exception	The student doesn't find an interesting internship; in this case he returns to the home page clicking on the button *Home* the system returns to the homepage

UC6 - Search of all Companies

Name	Search of all Companies
Actors	▶ Student
Entry condition	The student is logged to S&C platform and he wants to apply for an internship
Event Flow	□ The student goes to the *View Companies* section □ The system shows the page with a list of all the company registered to the sytem, listed in alphabetical order □ The student clicks on the name of the company who interested him □ The system shows the list of all internships advices published by the company □ The student select an intern-ship advice who interested them
Exit condition	the system shows the page with all the details about the internship
Exception	The student doesn't find an interesting internship; in this case he returns to the home page clicking on the button *Home* the system returns to the homepage

UC7 - Search an internship by name

Name	-Search an internship by name
Actors	▶ Student
Entry condition	The student is logged to S&C platform and he wants to apply for an internship
Event Flow	◻ The student search the internship by typing its subject on the search bar ◻ The student flags the appropriate options they want the internship to meet ◻ The systems shows a page with all the internships advice matching the subject and the options chosen, from the most relevant from the last ◻ The student select an internship who interested them
Exit condition	the system shows the page with all the details about the internship
Exception	◻ The student digits a subject and no internships match it. The system will warn the student that no internships are found ◻ The student doesn't find an interesting internship; in this case he returns to the home page clicking on the button *Home*; the system returns to the homepage

UC8 - Internship Recommendation

Name	Internship Reccomendation
Actors	▶ Student
Entry condition	the student, at registration, flags the option to receive notifications and the recommendation analysis is completed
Event Flow	◻ The student logs to the platform ◻ the student goes to the *Notification* section ◻ the student clicks on the notification related to the new internship published ◻ the system shows the page with the details about the internship
Exit condition	The student view the internship more in detail

UC9 - Students Recommendation

Name	Students Recommendation
Actors	▶ Company
Entry condition	the company, at registration, flags the option to receive notifications and the recommendation analysis is completed
Event Flow	◻ The comapny logs to the platform ◻ the company goes to the *Personal Internships* section ◻ the company clicks on the internship related to the notification ◻ the company goes on the *Notification* section ◻ the system shows the page with all the notification related to the internship ◻ the company select the notification it has to consider ◻ the system shows the notification with the list of the name of the students ◻ the company clicks on one of the name
Exit condition	The company view the profile of the student more in detail

UC10 - Students Application Process

Name	Students Application process
Actors	▶ Student ▶ Company
Entry condition	the student is logged to S&C platform and he finds an internship he wants to apply
Event Flow	◻ The student access the page of the internship he wants to apply (in any way) ◻ The system shows the page of the internship ◻ The student click on *Apply* button ◻ The system sends an e-mail to the company providing the internship ◻ The company logs to the system with its credentials ◻ The company goes to *Personal Internships* section ◻ The company select the corresponding internship ◻ The company goes to *Notification* section ◻ The system shows all the notification of the corresponding internship ◻ The company select the notification of the student ◻ The company click on *Accept Application* or*Reject Application* button to approve or reject the application
Exit condition	the system automatically sends an e-mail to the student to to inform whether the application has been accepted or rejected
Exception	The deadline has already passed; the system notifies the student that they cannot apply for that internship

UC11 - Company Send of Proposal

Name	Company Send of Proposal
Actors	▶ Student ▶ Company
Entry condition	the company is logged to S&C platform and it already received some student through the recommendation mechanism
Event Flow	◻ the company goes to *Personal Internships* section ◻ the company select the corresponding internship ◻ the company select a student for that internship ◻ the system shows the profile of the student ◻ the company clicks on ”Send Proposal” button ◻ the system sends an email to the student ◻ the student logs into the platform using his credentials ◻ The student goes to *Notification* section ◻ the student clicks on the notification received ◻ The student clicks on *Accept Proposal* or*Reject Proposal* button to approve or reject the proposal
Exit condition	the system return to the homepage
Exception	The student respond to the proposal when the deadline has already passed; the system notifies the student that they cannot reply to the proposal

UC12 - Selection Process Configuration

Name	Selection Process Configuration
Actors	▶ Company
Entry condition	Internship deadline is expired and the company is logged to S&C platform
Event Flow	◻ The company goes to *Personal Internships* section ◻ The company select the appropriate Internship advice ◻ The system shows the internship advice page ◻ The company goes on *Configuration* section ◻ the company configures the number of steps (up to 2), the metrics to evaluate students and the questionnaire provided at each step (it is defined the structure, deciding which questions are closed or open/oral, and also the score for each closed question) ◻ the company save the configuration clicking on *Save* button ◻ the company goes on *Students Applied* section ◻ the system shows a page with all the students applied for the internship ◻ the company decided, for each candidate, the date of each step ◻ the company click on the button *Save*
Exit condition	The selection process is now configured for each student; the system automatically sends an e-mail to the students to inform them
Exception	◻ the company has not configured every part; the system warn the company ◻ The company has not assigned the respective dates for each student

UC13 - Selection Process Running

Name	Selection Process Running
Actors	▶ Student ▶ Company
Entry condition	Internship deadline is expired, the company configure the selection process and it has ready to start
Event Flow	◻ The company gives to the student the questionnaire with closed and open questions ◻ The student answers to the questionnaire ◻ The company asks some oral questions (if they are configured in this step) ◻ The student answers to oral questions ◻ The company logs to the platform ◻ the company goes into *Personal Internship* section ◻ the system shows the page with all the personal internships of the company ◻ the company select the appropriate internship ◻ the company select, from the list of students, the appropriate student ◻ the company select the appropriate step of the selection process ◻ the company click the button *Insert Answers* ◻ the system shows a form to fill with the answers ◻ the company insert the answers ◻ the company clicks on the button *Evaluate* ◻ the system shows the page with the evaluated answers ◻ the company decide to accept/reject/postpone the student ◻ the company click on the appropriate button to send the rigth notification
Exit condition	the evaluation of this step of the selection process is completed; the system send a notification to the student to inform him about the decision of the company
Exception	the company doesn't insert all the answers; the system shows an error message and the company has to fill all the fields of the form

UC14 - Collection of Student Selection Process Feedback

Name	Collection of Student Selection Process Feedback
Actors	▶ Student
Entry condition	the selection process is complete and the student has already received the email with the result of the interview
Event Flow	◻ The student logs into the platform with his credentials ◻ The student goes to the *Notification* Section ◻ The system shows the page with all the notifications for the student ◻ The student open the notification related to the selection process result ◻ The student open the questionnaire attached to the notification ◻ The system shows a form with a series of questions to answer ◻ The student answers to the questions ◻ The student clicks the button *Submit* to submit the feedback
Exit condition	the feedback is submitted; the system collects the feedback in order to improve the recommendation mechanism
Exception	the student doesn't answer to all the questions when he tries to submit; the system shows a warn message

UC15 - Collection of Company Selection Process Feedback

Name	Collection of Company Selection Process Feedback
Actors	▶ Company
Entry condition	The company is logged to S&C platform and has just sent the selection results
Event Flow	◻ The system shows a form with a series of questions to answer ◻ the company answers to the questions ◻ the company clicks on the button *Submit*
Exit condition	the feedback is submitted;the system collects the feedback in order to improve the recommendation mechanism
Exception	the company doesn't answer to all the questions when he tries to submit; the system shows a warn message

UC16 - Collection of Internship Feedback

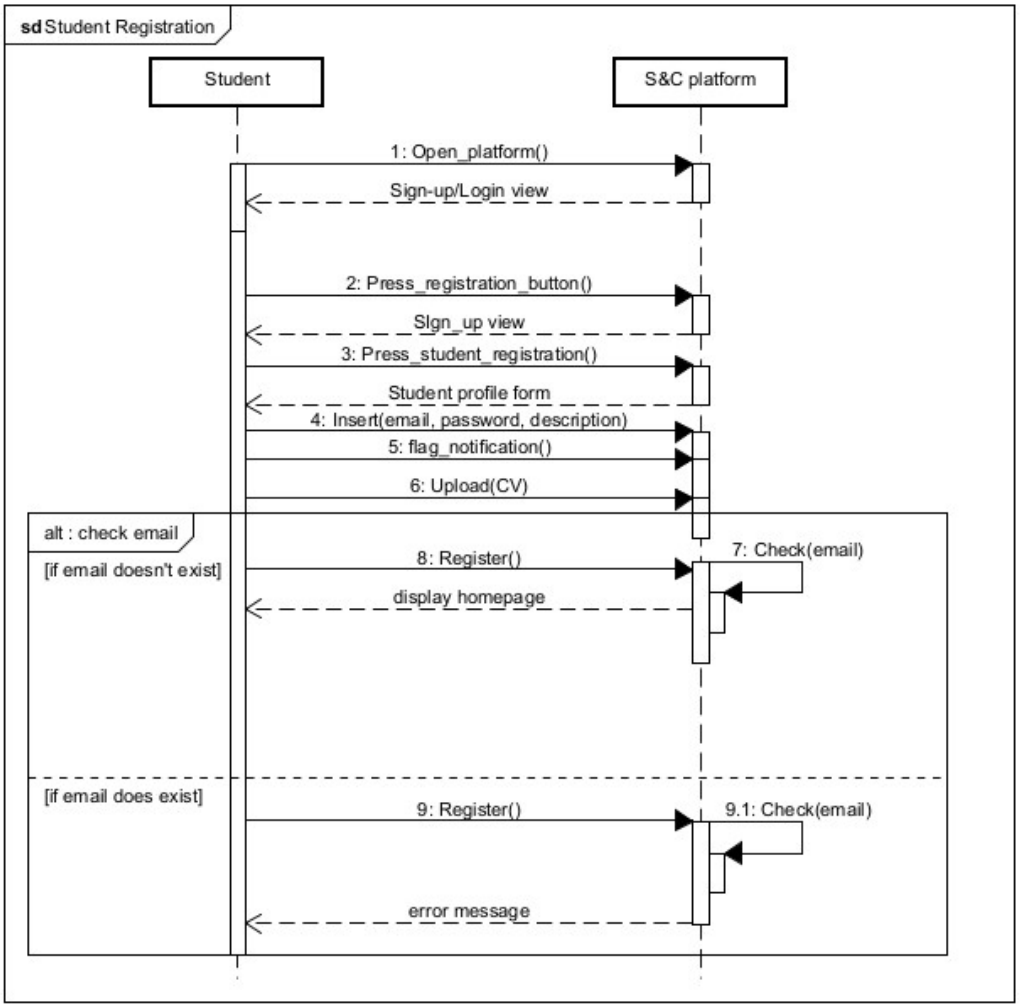
Name	Collection of Internships Feedback
Actors	▶ User
Entry condition	The User is logged to S&C platform and it completed an internship
Event Flow	◻ The user goes to *Ongoing Internships* section ◻ The system shows a page with all the ongoing internships of the user ◻ The User select the interested internship ◻ the User opens the form in the page of the internship ◻ the system shows a form with a series of questions to answer ◻ the User answers to the questions ◻ the User clicks on the button *Submit*
Exit condition	the feedback is submitted;the system collects the feedback in order to improve the recommendation mechanism
Exception	the user doesn't answer to all the questions when he tries to submit; the system shows a warn message

UC17 - Collection of Complaints

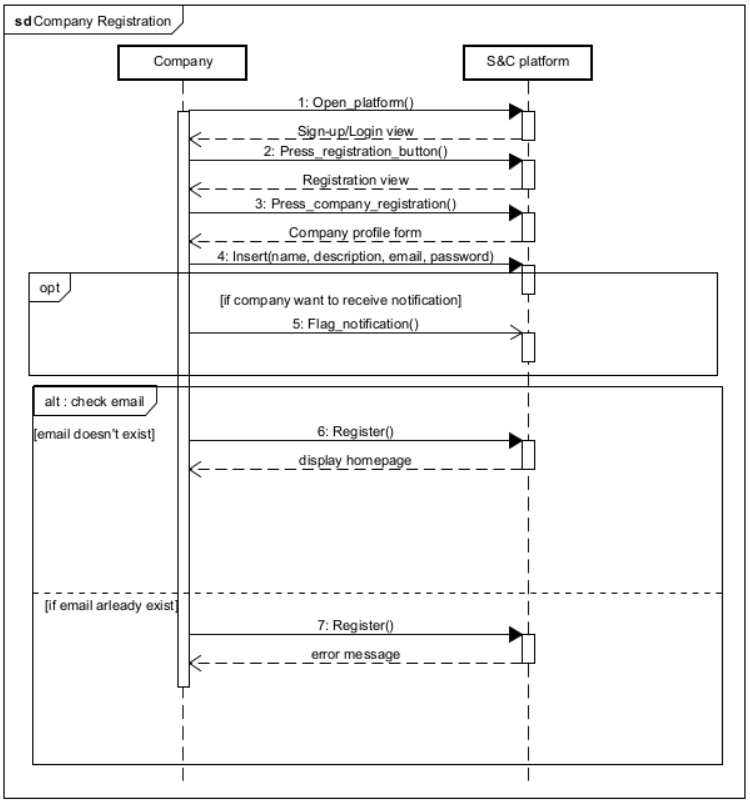
Name	Collection of Complaints
Actors	▶ User
Entry condition	The user is logged to S&C platform, and there is an ongoing internship in which it is involved.
Event Flow	□ The user goes on *Ongoing Internships* section □ The system shows a page with all the ongoing internships related to the user □ The user select the interested internship □ The system shows the page where the user can monitor the execution of the internship □ The user write a complaint using the apposite box (max 50 words) □ The user click on the *Submit* button
Exit condition	The complaint has been submitted; the system collect the complain related to the internship
Exception	The complain exceed the max number of words (50); the system warn the user that the complaint is too long

3.2.3 Sequence diagrams

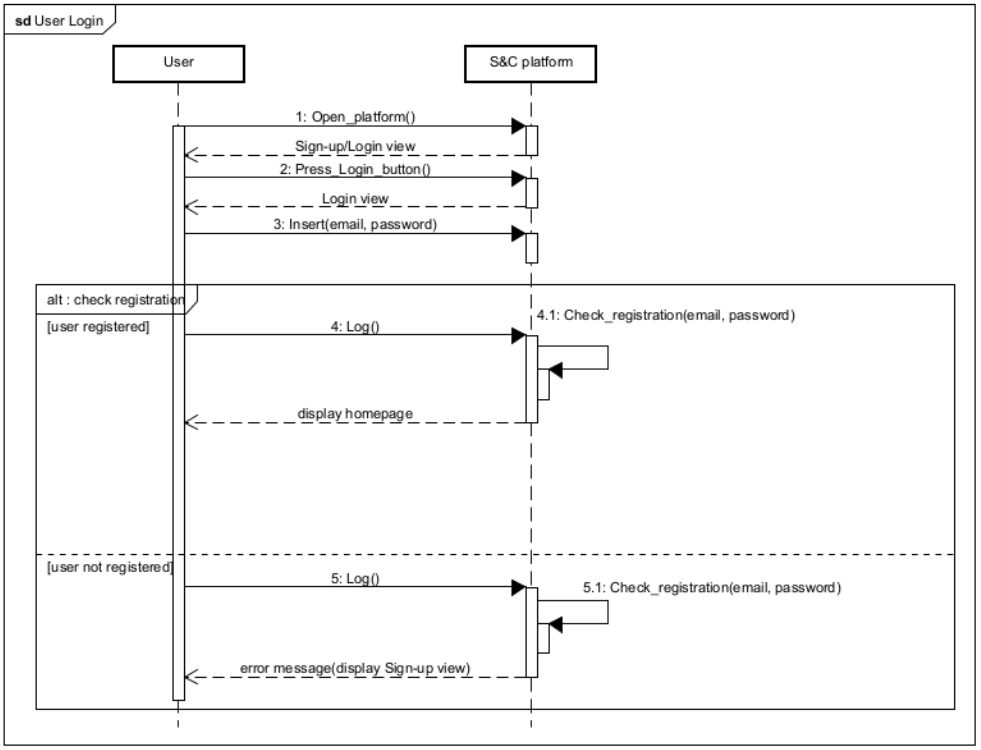
[UC1] - Student Registration



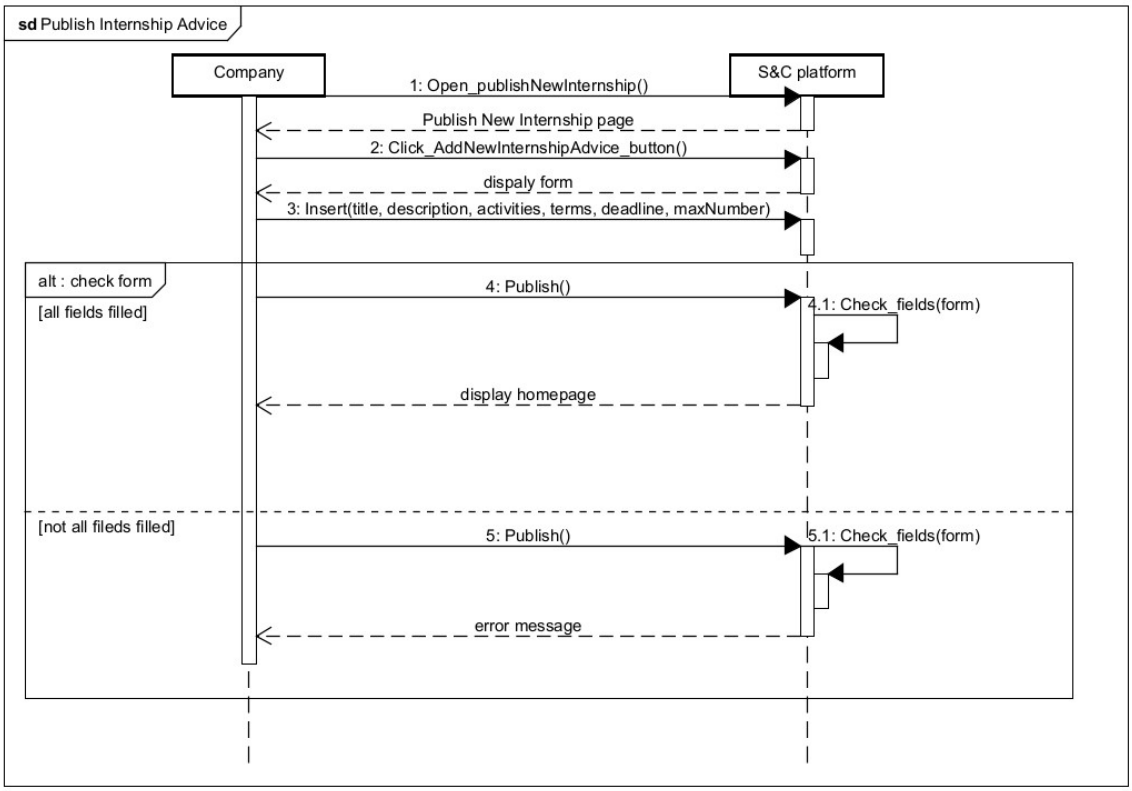
[UC2] - Company Registration



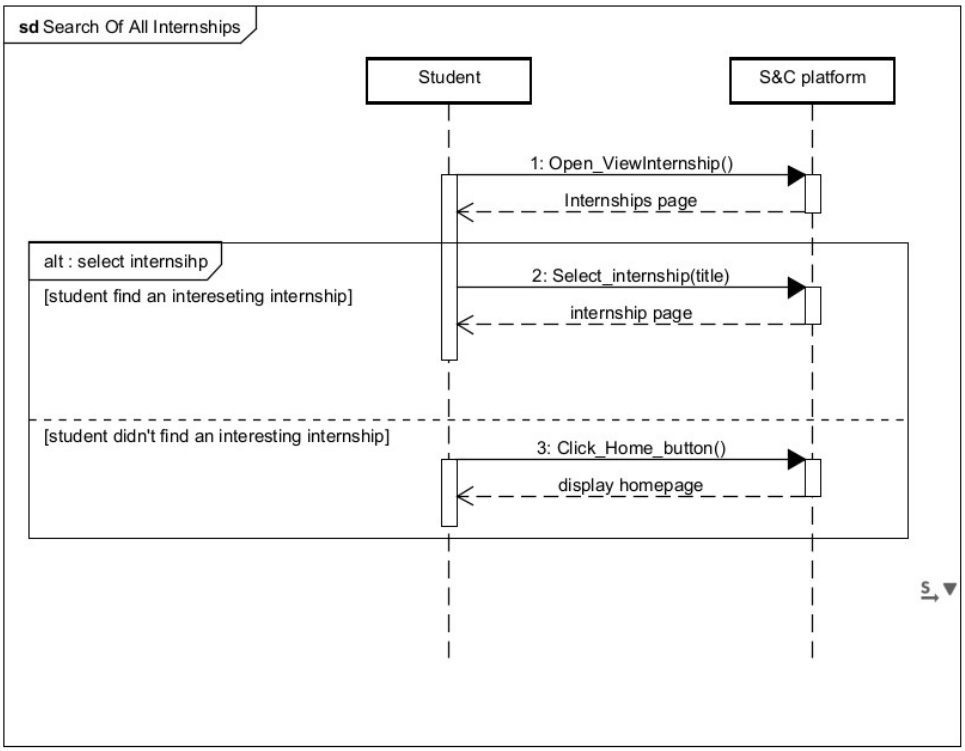
[UC3] - User Login



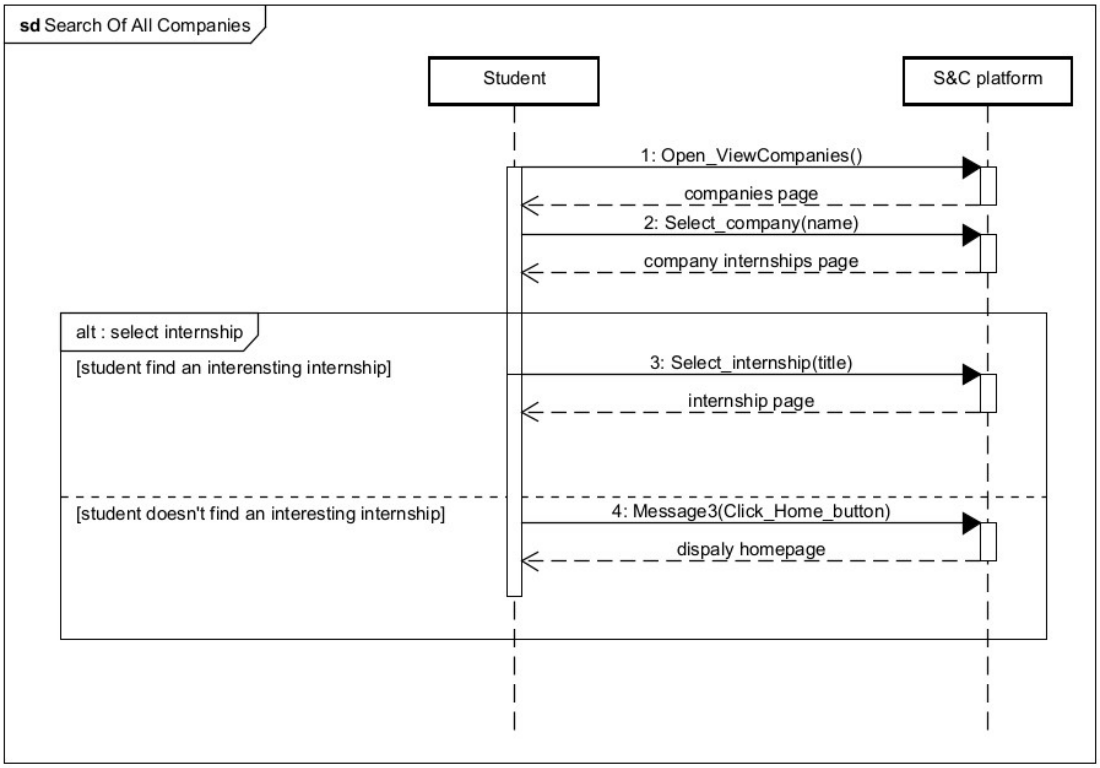
[UC4] - Publish Internship Advice



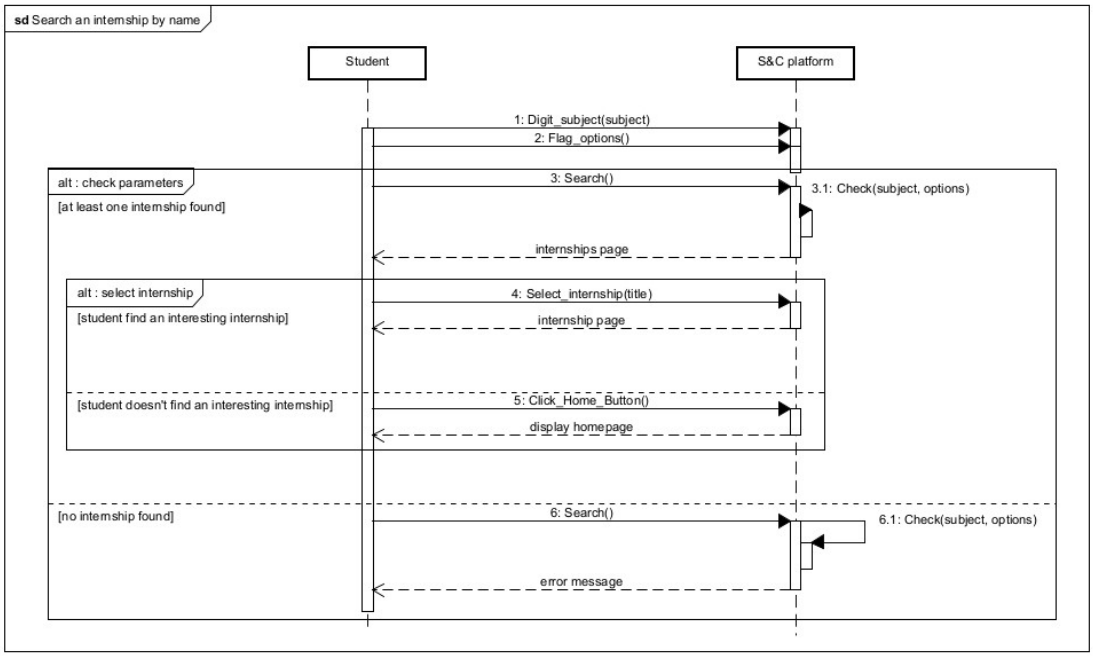
[UC5] - Search of all Internships



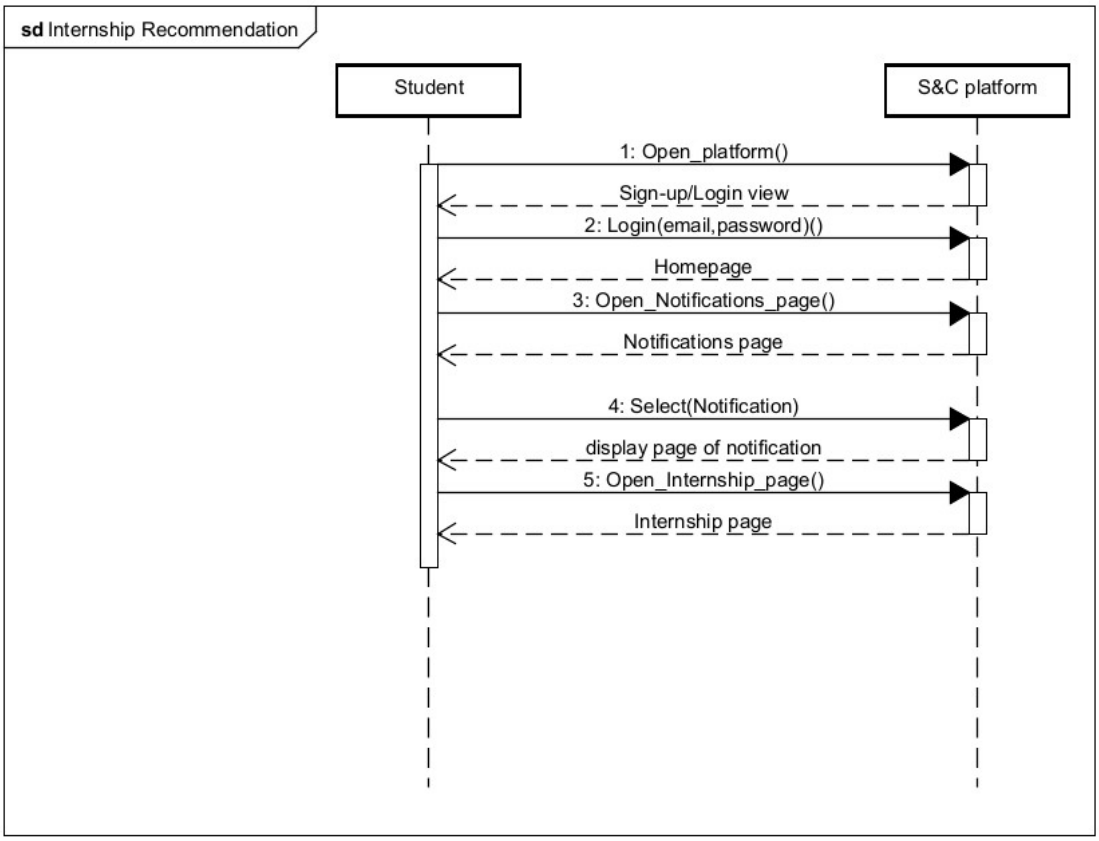
[UC6] - Search of All Companies



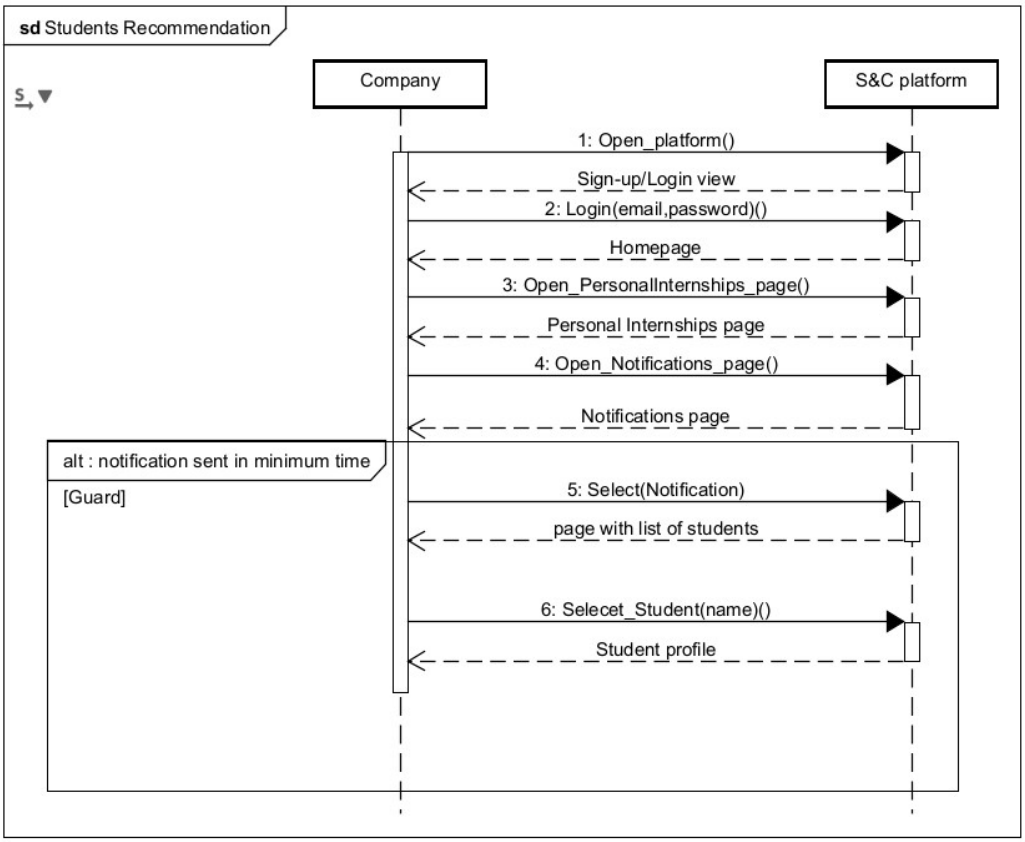
[UC7] - Search an Internship by Name



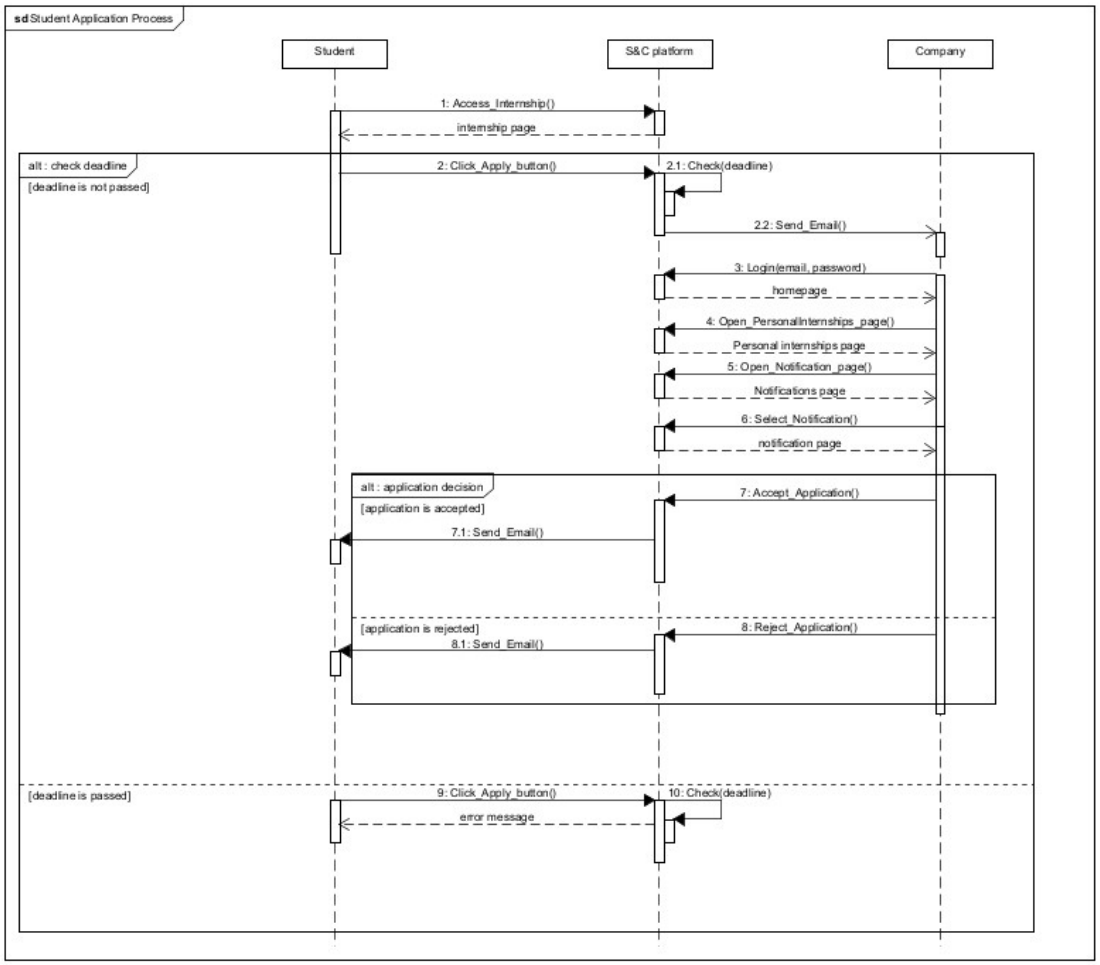
[UC8] - Internship Recommendation



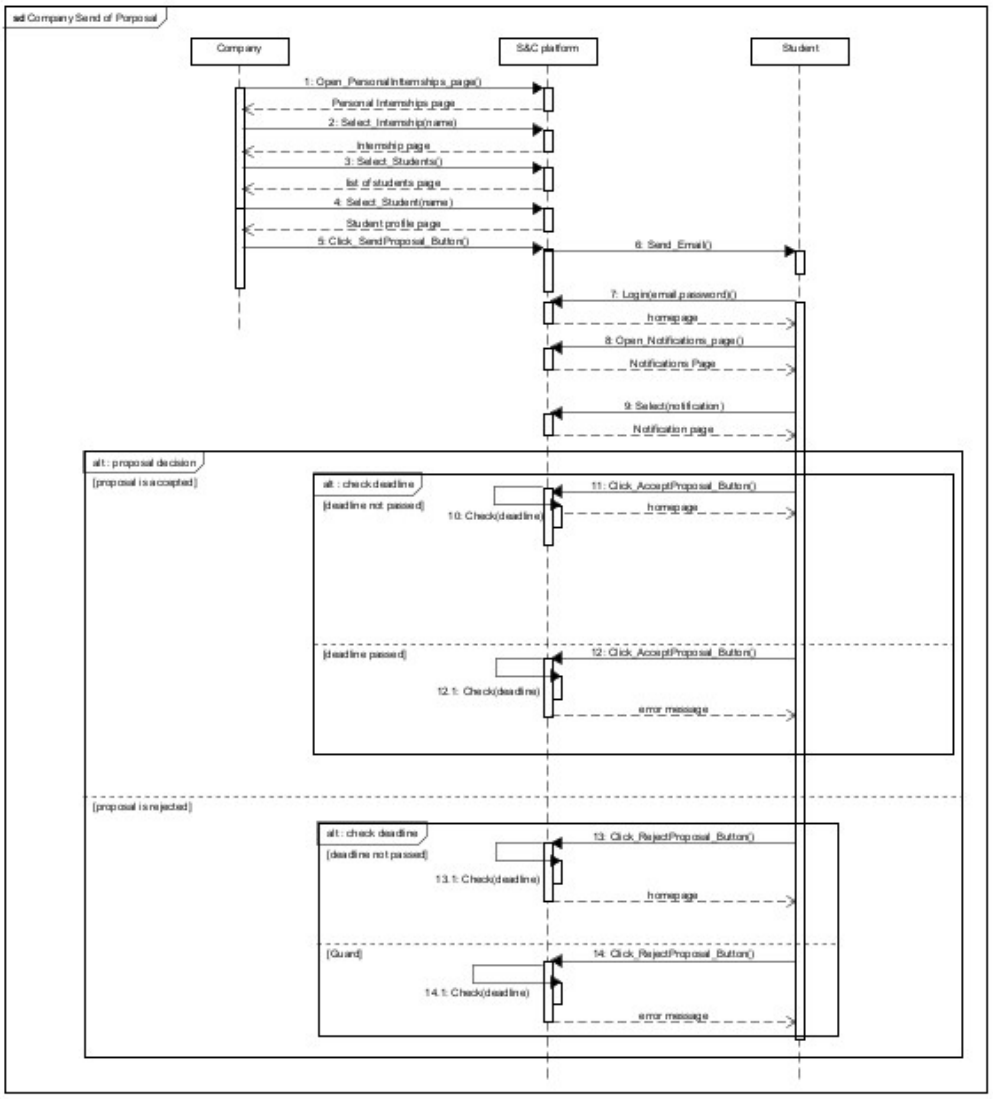
[UC9] - Students Recommendation



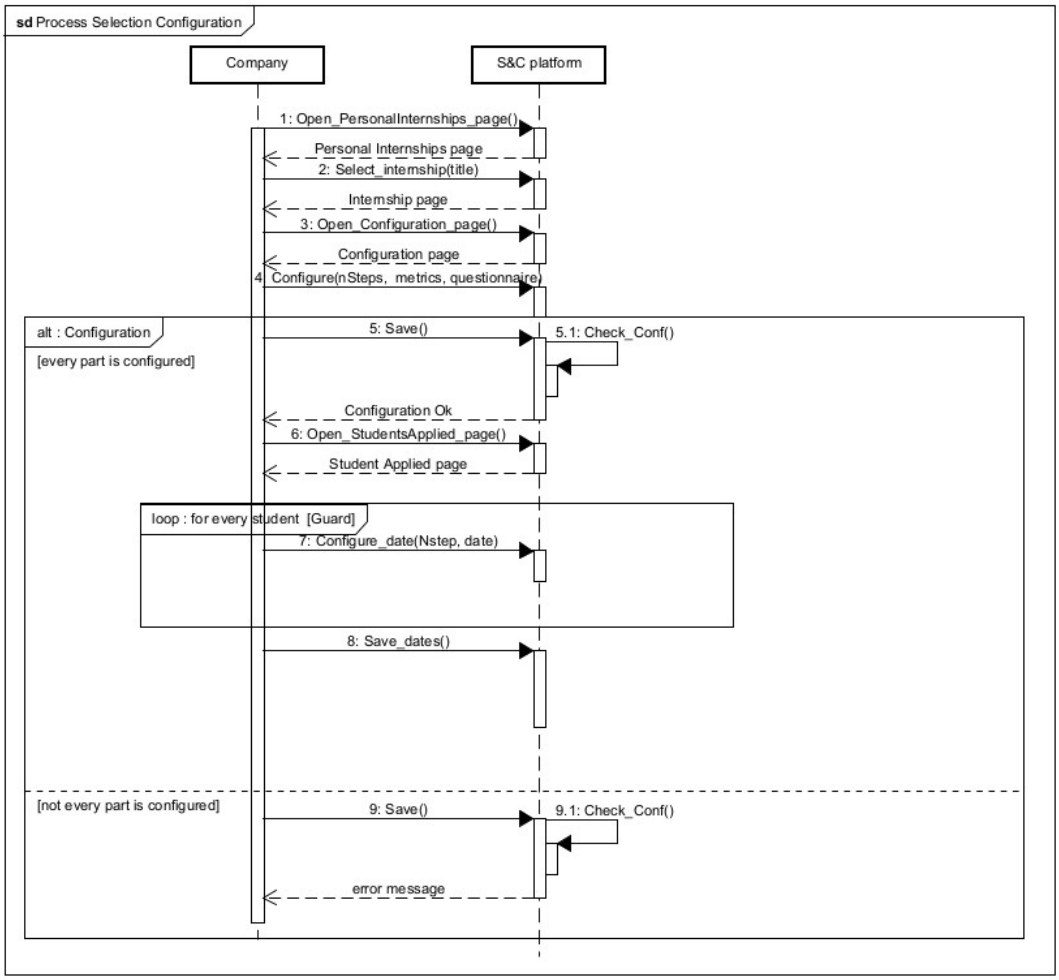
[UC10] - Student Application Process



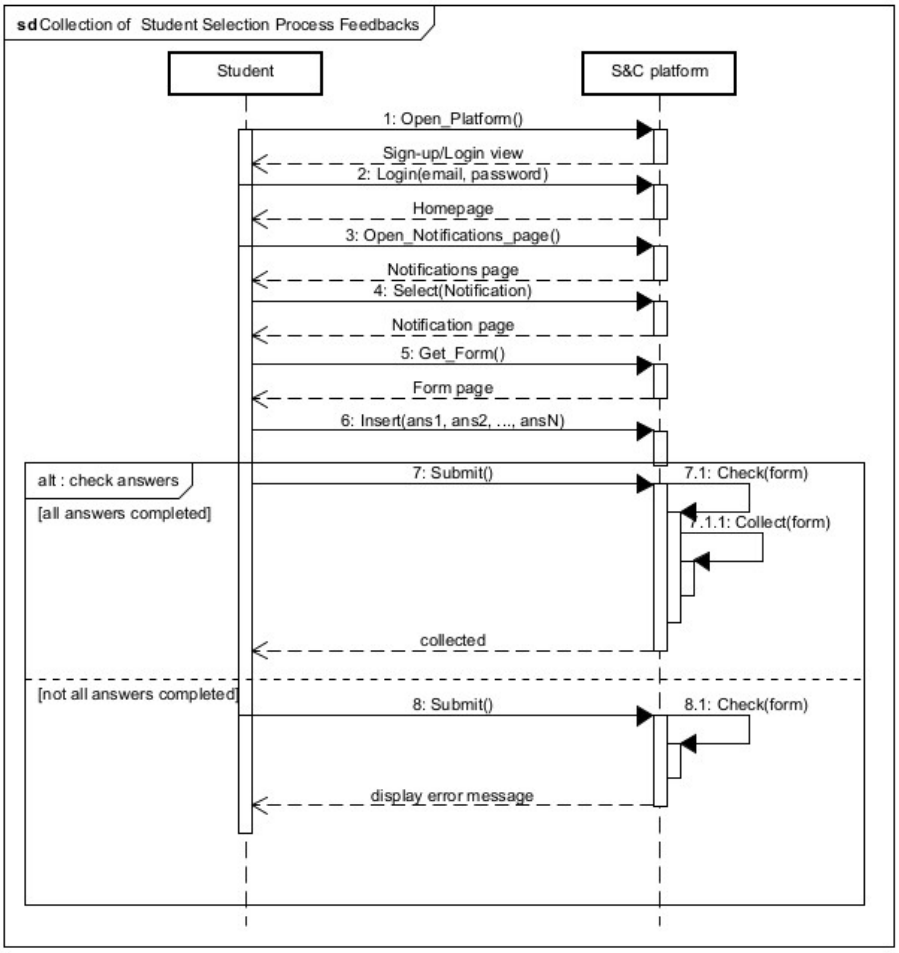
[UC11] - Company Send of Proposal



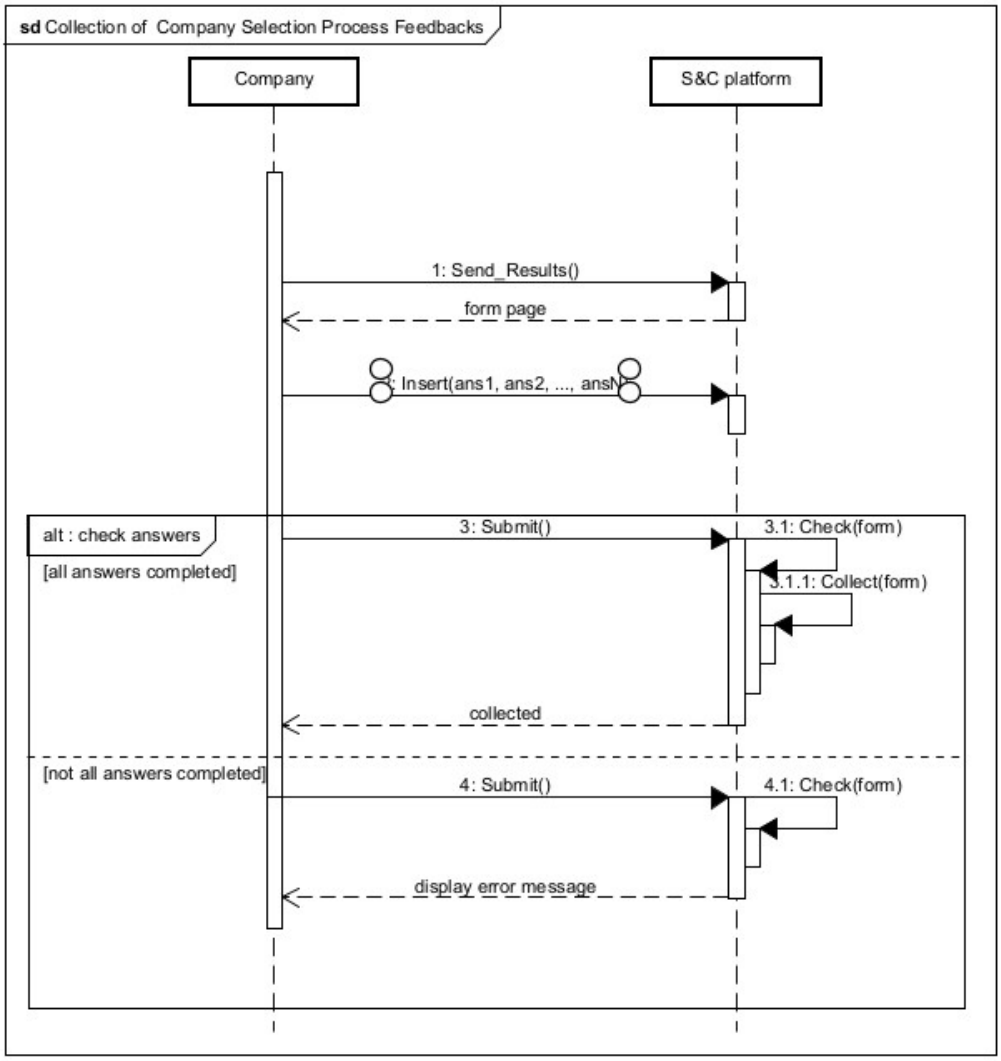
[UC12] - Selection Process Configuration



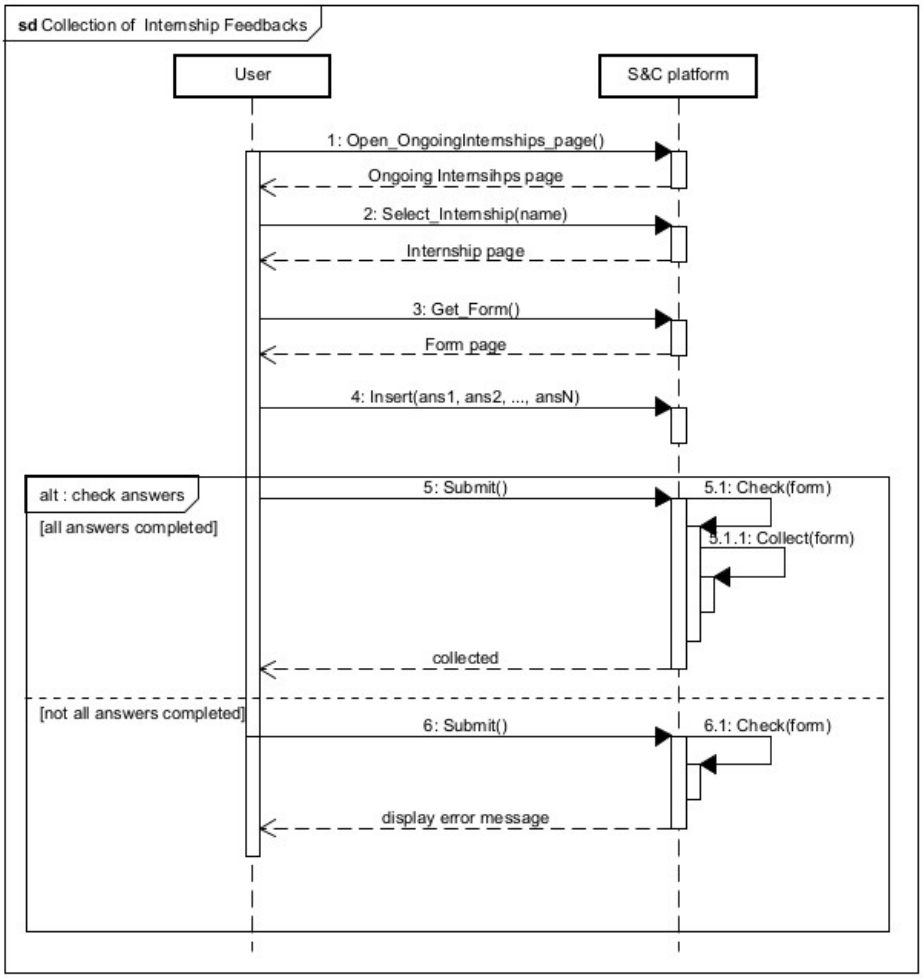
[UC14] - Collection of Student Selection Process Feedbacks



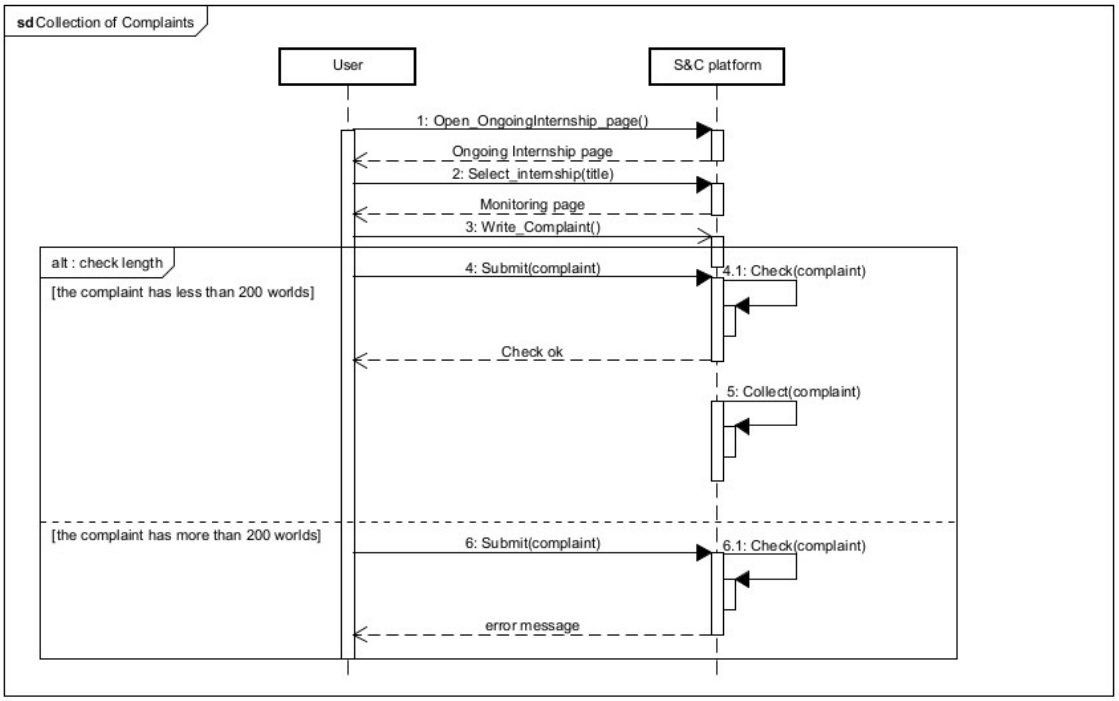
[UC15] - Collection of Company Selection Process Feedbacks



[UC16] - Collection of Internships Feedbacks

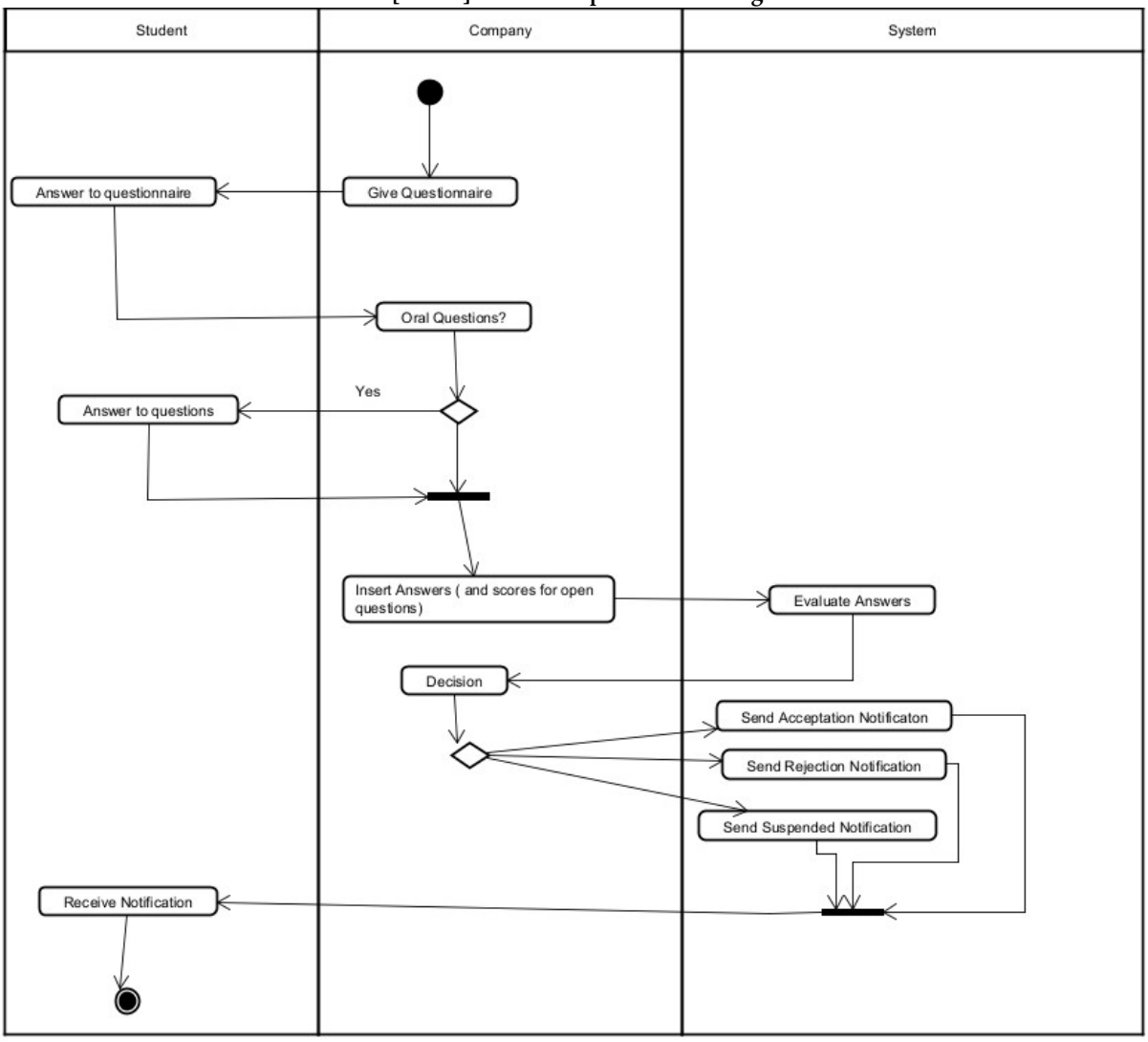


[UC17] - Collection of Complaints



3.2.4 Activity diagrams

[UC13] - Selection process Running



in the figure it is represented the activity diagram associated do the selection process running UC; in this case we prefer use an activity diagram to represent also the interaction, during the selection interview, between the 2 actors without explicitly using the system; in fact, the filling of the questionnaire and the oral answers are all done without the use of the system; then, to collect the answers, the company use the system and the student receive a notification with the outcome of the interview for this step (Accepted, Rejected, Suspended)

3.2.5 Requirements mapping

Table 3.1: Requirements mapping for goal G1

[G1] students and companies establish contacts for doing internships	
[R00000] when a notification of a user is generated, the user receives it on its mailbox (in a more concise version) and can consult it on its notification section	
[R10101] the system allows students to sign up to the platform with their institutional mails	[D10101] students upload their CV in Europass format
[R10102] the system allows a student to set up whether he/she wants to take part into the recommendation	[D10102] information on a student CV do not contradict each other
[R10103] the system allows students upload their CV to the platform	[D10302] information companies insert in internship advice do not contradict each other
[R10104] the system allows students to add on their profile the name, the surname and a brief description of themselves	
[R10105] when a CV is uploaded, the system verifies if it is digitally signed by the profile mail	
[R10106] the system allows students to log in into the system by providing the registration mail and the chosen password	
[R10107] the system allows students to change their profile information (including the CV) and their access information	
[R10201] the system allows companies to sign up to the platform with their company address	
[R10202] the system allows companies to insert the main information regarding their business area and area of expertise	
[R10203] the system allows a company to set up if it wants to take part into the recommendation analysis	
[R10204] the system allows companies to log in into the system by providing the registration mail and the chosen password	
[R10205] the system allows companies to change their profile information and their access information	
[R10301] the system allows companies to publish internship advice where they specify the main information regarding the internship (brief description, experience required, desired skills, main activities involved and the terms), the submission deadline and the max number of applications	
[R10401] the system allows students to search internships advice by name (and also to see the complete list of available advice). The system shall act as a search engine to present also the names of the advice that are similar to the searched one	
[R10402] the system allows students to search companies by name (and also to see the complete list of registered companies) and then access to their profile	
[R10403] the system allows students to filter the results they searched (e.g. ”only paid internships”, ”only companies located in Lombardy”)	
[R10501] when the system recognizes that a new internship advice that might interest a student (that allowed the recommendation option) is published, it notifies that student by sending him an e-mail (to its registration address)	
[R10601] when the system recognizes that a student has a profile that would fit an internship advice, the company that published the advice is notified (for students and companies that both take part into the recommendation analysis)	
[R10701] the system allows students to apply for any internship advice which deadline has not expired	

Table 3.2: Requirements mapping for goal G2

[G2] internships selections can be monitored and supported by the system
[R20101] when the deadline for an internship advice is expired, the system allows the company to set up the selection process by specifying for each step, the relative questionnaire (with metrics for each question) and the date in which provide it to a student (dates may differ between different students) [R20102] the system includes into a selection process only student that had an accepted application for the relative internship advice [R20201] the system notifies students for any interview date [R20202] the system automatically calculates the scores of questionnaire closed answers [R20203] the system allows companies to manually insert scores for questionnaire open answers [R20204] the system allows companies to visualize and compare selections scores [R20205] in any selection phase, the system allows companies to discard a student currently involved in the selection process (discarded students are removed by the selection process) [R20206] in any selection phase, the system allows companies to accept a student currently involved in the selection process (accepted students are removed by the selection process) [R20207] the system allows companies to write a personalized message to communicate the result of a selection [R20208] when a selection result is prepared for a student (with the relative message), it is notified to the student

Table 3.3: Requirements mapping for goal G3

[G3] ongoing internships can be monitored from the system
[R30101] the system allows students and companies to consult the internships (ongoing or finished) [R30102] the system allows students and companies to report complaints on the internships they are involved in [R30103] the system does not allow users different from their creator to consult complains

3.3 Performance Requirements

For the system functions related to user navigation, we require a response time up to 5 seconds.

The mail notification system should send any notification at most 1 minute after the moment in which the notification was generated.

The recommendation system should produce its results with at most 1 week of distance from the last time it produced them.

3.4 Design Constraints

3.4.1 Standards compliance

- Since S&C uses personal data of users, it is necessary that the platform is in compliance with the General Data Protection Regulation (GDPR), a regulation in EU law on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA).
- The CVs uploaded by students on the platform must necessarily be in the EUROPASS format in order to be processed by the system.

3.4.2 Hardware limitations

To make the best use of the platform, it is necessary that the devices used to access it meet the following requirements:

- **RAM**: minimum 4 GB of RAM for users who use the platform in a relatively simple way, such as sending CVs or applying for internships.
- **Internet Connection**: a stable connection of 5 Mbps is required for a smooth experience. This would support navigation on the platform and viewing content. Slower connections may lead to long loading times on pages with a lot of data.
- **Screen Resolution**: Users should have devices with screens of at least 1280x720 px to ensure a good visual experience.

3.4.3 Other constraints

The platform is designed to allow companies to advertise their internships and for students to find internships relevant to their studies. No overly complicated functionality is required; therefore, the UI must be user-friendly to allow users to navigate the platform easily.

3.5 Software System Attributes

3.5.1 Reliability

Considering the criticality of the information managed by the application (e.g. interview dates, CV, e-mail addresses) we require an high level of reliability in each sub-part of the system.

For the recommendation system reliability we ask for a... .

3.5.2 Availability

Since the application does not have real-time interactions or much critical functions to ensure, if the system went down for few hours it would not be an huge concern for most users. However, there some functions that require an higher level of availability than the others:

- notification system: it should be available for at least one hour in a day, in order to guarantee that notifications are not sent to users with a delay higher than one day (since notifications are also sent by email, we can rely on the availability of users mail servers, as stated in the assumption section);
- selection process system: it is highly recommended that the selections calendars and the relative questionnaires are available at least in work hours. As we stated in the assumption section, we always take for grant the fact that companies (and students) have a copy of calendars (and also of the questionnaires) for the companies;
- ongoing internship monitoring: at least in work hours, the monitoring system should be available. Little downtimes are still tolerated but it is highly recommended that for the majority of the time is possible to monitor the ongoing internship status.

As general rule, maintenance should always occur off the work hours of the majority of the companies registered.

3.5.3 Security

In this section we define the main kinds of security concern that the system should address:

- e-mails sent from the system always have to be sent from a certified mail address. Moreover, e-mails sent from the system must be encrypted and must not contain any password;
- attacks related to system availability (e.g. DOS), to data confidentiality, integrity and users authenticity must be taken into consideration, also considering the public nature of the application;
- a CV must be digitally signed from the student that upload it;
- uploaded CV should be scanned to ensure that they don't contain viruses.

3.5.4 Maintainability

For S&C, which manages multiple users (students and companies) and handles a large amount of sensitive data, maintainability is a crucial aspect:

- the code has to be well documented and tested; in particular, a testing routine has to be provided to check if the system still work or not.
- The system structure should provide the ability to deploy updates in the backend without the users noticing it
- to facilitate maintainability, system components should be created to be as modular as possible

3.5.5 Portability

We highlight the fact that the application targets are students and companies that may use operative systems of any kind, therefore portability should be increased, in order to spread the audience. On the other hand, non-desktop devices (such as mobile devices, smartwatches ecc.) are not an huge concern of this kind of application, so we don't put much effort on emphasizing the portability also in this direction. At the end, we encourage portability but we ask for it at least for general purposes desktop operative systems.

4 Formal Analysis

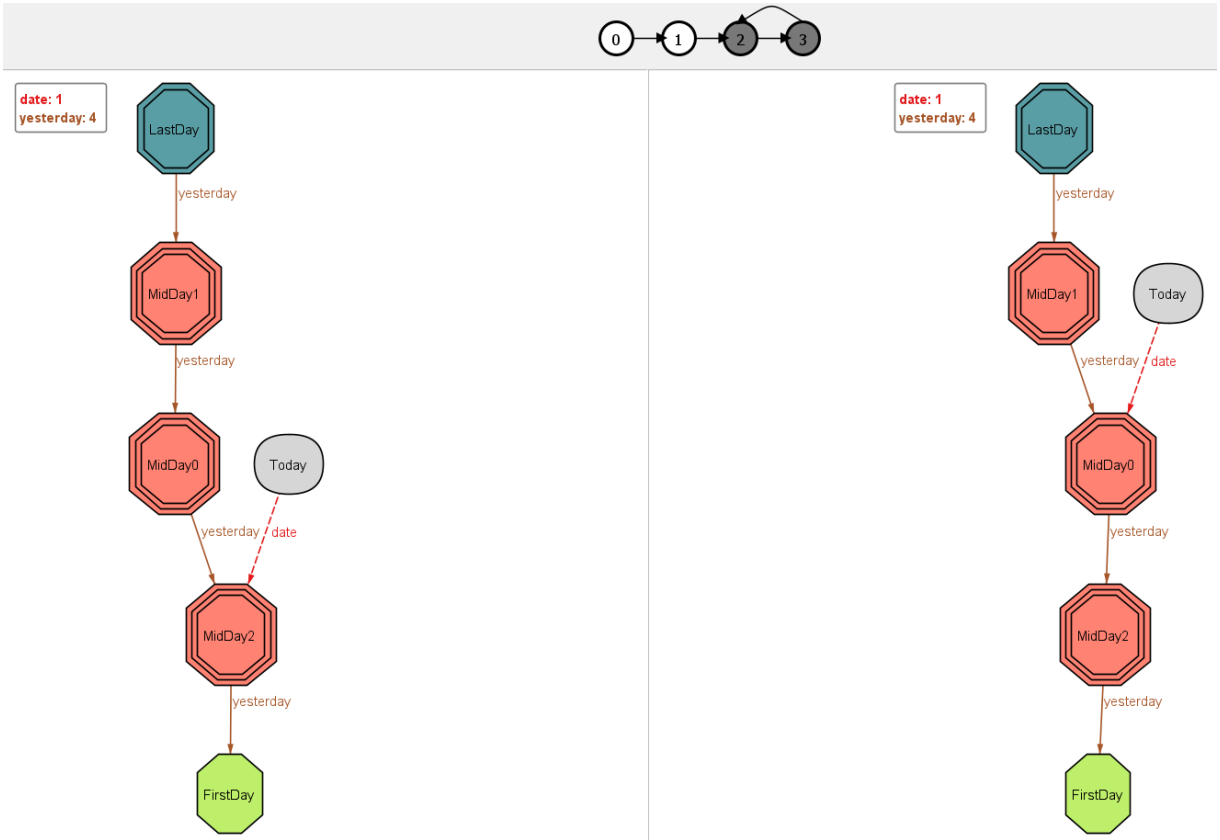
In this chapter we provide a formal modeling of a restricted part of the system. In particular we modeled:

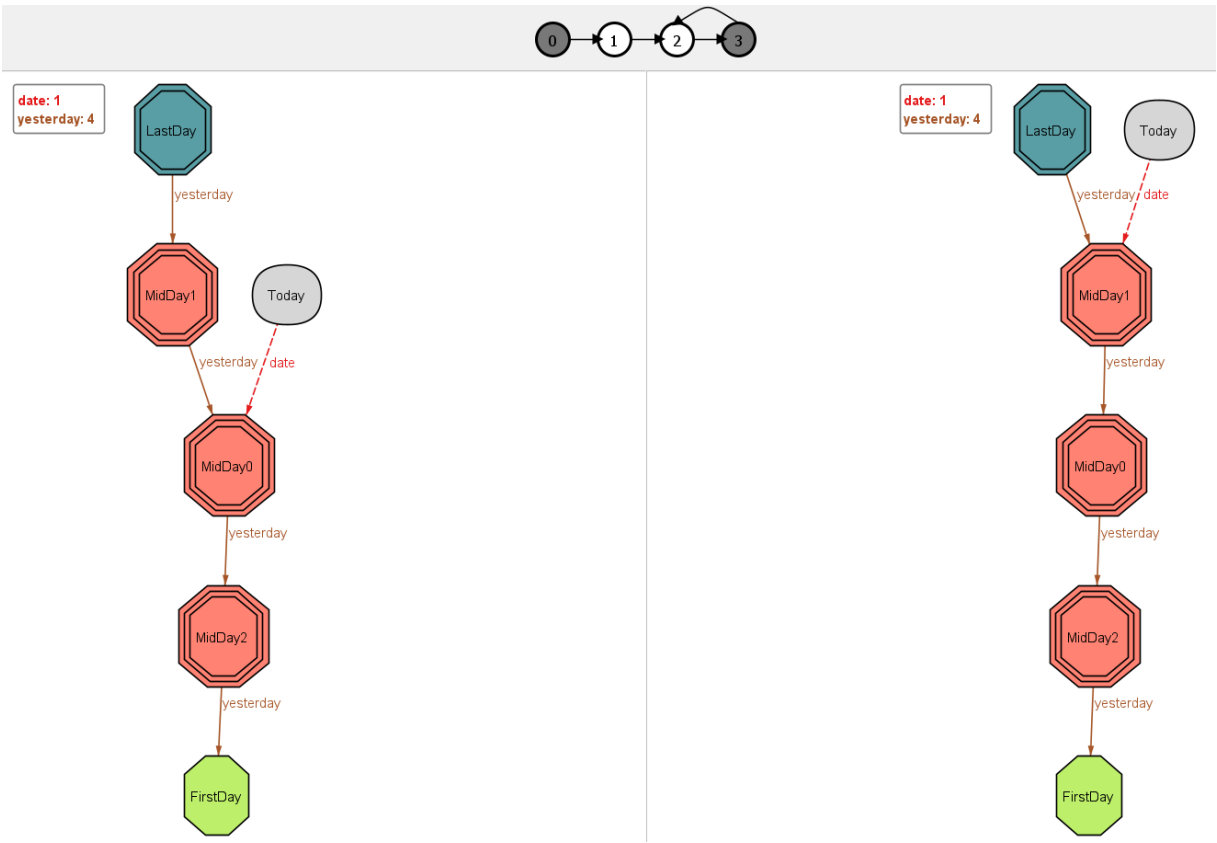
- selection processes structure;
- applications and withdrawals to internship advice (using Alloy 6 temporal operators).

To describe the passing of time (especially in temporal simulations) we built a tiny Alloy *calendar*, which is basically a path of `date` linked each other by means of the relation `yesterday`:

```
sig Date {}
one sig FirstDay extends Date {} //for uniformity of the concept of "yesterday", the calendar begins with the "MidDay" that has as yesterday the "FirstDay"
sig MidDay extends Date
    {
        yesterday: one Date
    }
one sig LastDay extends MidDay {}
one sig Today
    {
        var date: one Date
    }
fact calendar //facts to design the "date chain"
    {
        all d,d1:MidDay | (d!=d1) implies d.yesterday != d1.yesterday  //a day can't be "the yesterday" of more than one day
        all d:Date | d in MidDay or d in FirstDay //a date or is "MidDay" or is a "FirstDay"
        all d:LastDay | no d1:MidDay |  d1.yesterday = d  //the last day has no tomorrows
        all d:MidDay | d not in d.^(yesterday)  //no "loops" ("a day can't stay before itself in the calendar")
    }
fact todayFacts //facts to set up "Today"
    {
        all t:Today|t.date in MidDay and t.date.yesterday in FirstDay //the first "Today" is the "MidDay" that has as yesterday the "FirstDay"
        always (all t: Today | t.date not in LastDay implies t.date'.yesterday = t.date) //"today" must move in steps "one day after the other"
    }
```

Then, What will be generated is a ”path of dates” that represents the calendar and `Today` must ”move” day-by-day following the calendar structure:


0
1
2
3
date: 1
yesterday: 4
LastDay
yesterday
MidDay1
yesterday
MidDay0
Today
yesterday
date
MidDay2
yesterday
FirstDay
date: 1
yesterday: 4
LastDay
yesterday
MidDay1
Today
yesterday
date
MidDay0
yesterday
MidDay2
yesterday
FirstDay


0
1
2
3
date: 1
yesterday: 4
LastDay
yesterday
MidDay1
Today
yesterday
date
MidDay0
yesterday
MidDay2
yesterday
FirstDay
date: 1
yesterday: 4
LastDay
Today
yesterday
date
MidDay1
yesterday
MidDay0
yesterday
MidDay2
yesterday
FirstDay

4.1 Applications and withdrawals simulation (Alloy 6 temporal simulation)

We modeled a simplified version of the sub-part of the system related to students applications to internship advice. Several parts were omitted in order to highlight what we believed were the most interesting constraints, such as the fact that companies have to accept students applications or the invitation mechanism:

```
//profiles modeling
    sig Mail {}
    sig Profile
        {
            mail: one Mail
        }
    sig Student extends Profile {}
    sig Company extends Profile{}
    fact register
        {
            all p: Profile | p in Company or p in Student
        }
    fact noDuplicateMails
        {
            all p1, p2: Profile | (p1 != p2) implies (p1.mail != p2.mail)
        }
```

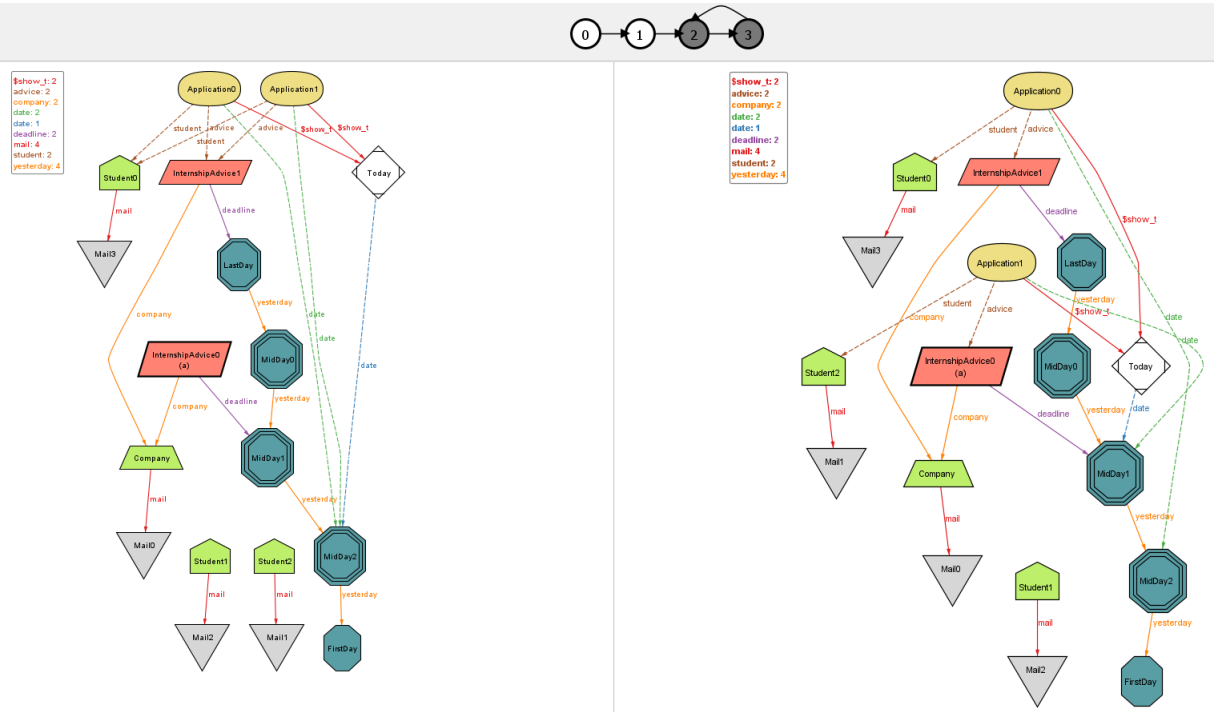
```
//application modeling
    sig InternshipAdvice
        {
            company: one Company,
            deadline: one Date
        }
    sig Application
        {
            var date: one Date,
            var advice: one InternshipAdvice,
            var student: one Student
        }
    fact applicationFacts
        {
            always (all a:Application| ((a.advice!=a.advice' or a.student!=a.student') ) implies (some t:Today|a.date'=t.date'))
                //if an application "changes", its date must set to "Today"
            always (all a:Application| ((a.advice=a.advice' and a.student=a.student') ) implies (a.date' = a.date))
                //if an application "does not change", its date must not change
            always (all a:Application| a.date = a.advice.deadline or a.date in a.advice.deadline.^(yesterday))
                //any application must be sent within the advices deadlines
        }
```

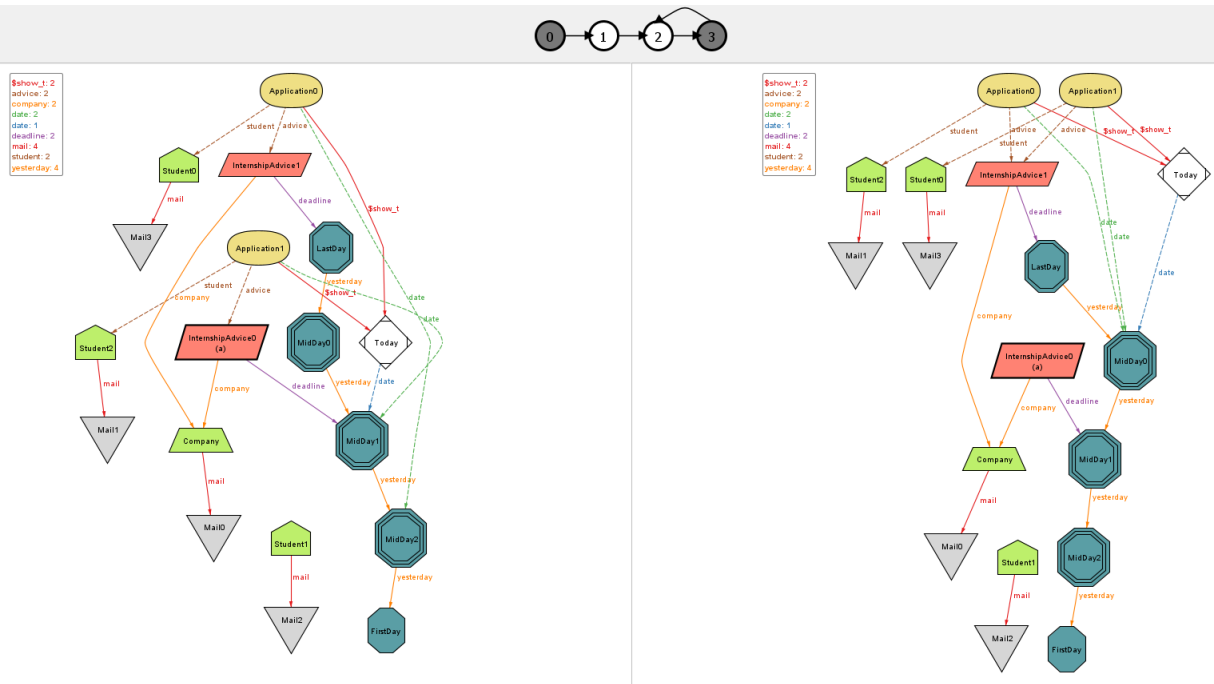
By setting up a quite self-explainable predicate to show worlds are trades that can show clearly the expected behavior of the system:

```
pred show
    {
        all a:Application| some t:Today | a.date = t.date //all pre-simulation applications are submitted in first simulation day
        some a:InternshipAdvice|  a.deadline not in FirstDay //some pre-simulation advices can't have a deadline in "FirstDay"
        always (some a:Application| a.date != a.date') //in this way, at least one application has to "change" each day

        #(Application) = 2
        #(InternshipAdvice) = 2
        #(Date) = 5
    }
run show for 5
```

we can simulate a classic scenario where students sends applications for available advice within deadlines:





0. the trace starts with two application enrolled `MidDay2` (which is `Today` in this step). These two applications refer to the same `InternshipAdvice` (`InternshipAdvice1`);
1. in this step, `Application0` does not change while `Application1` has its student changed and its advice change. We notice that even the application date is changed to the date point by actual `Today`;
2. in this step each application changes somehow. Their dates are properly updated and they are not related to advice which deadline has already expired.

4.2 Selection process structure (static simulation)

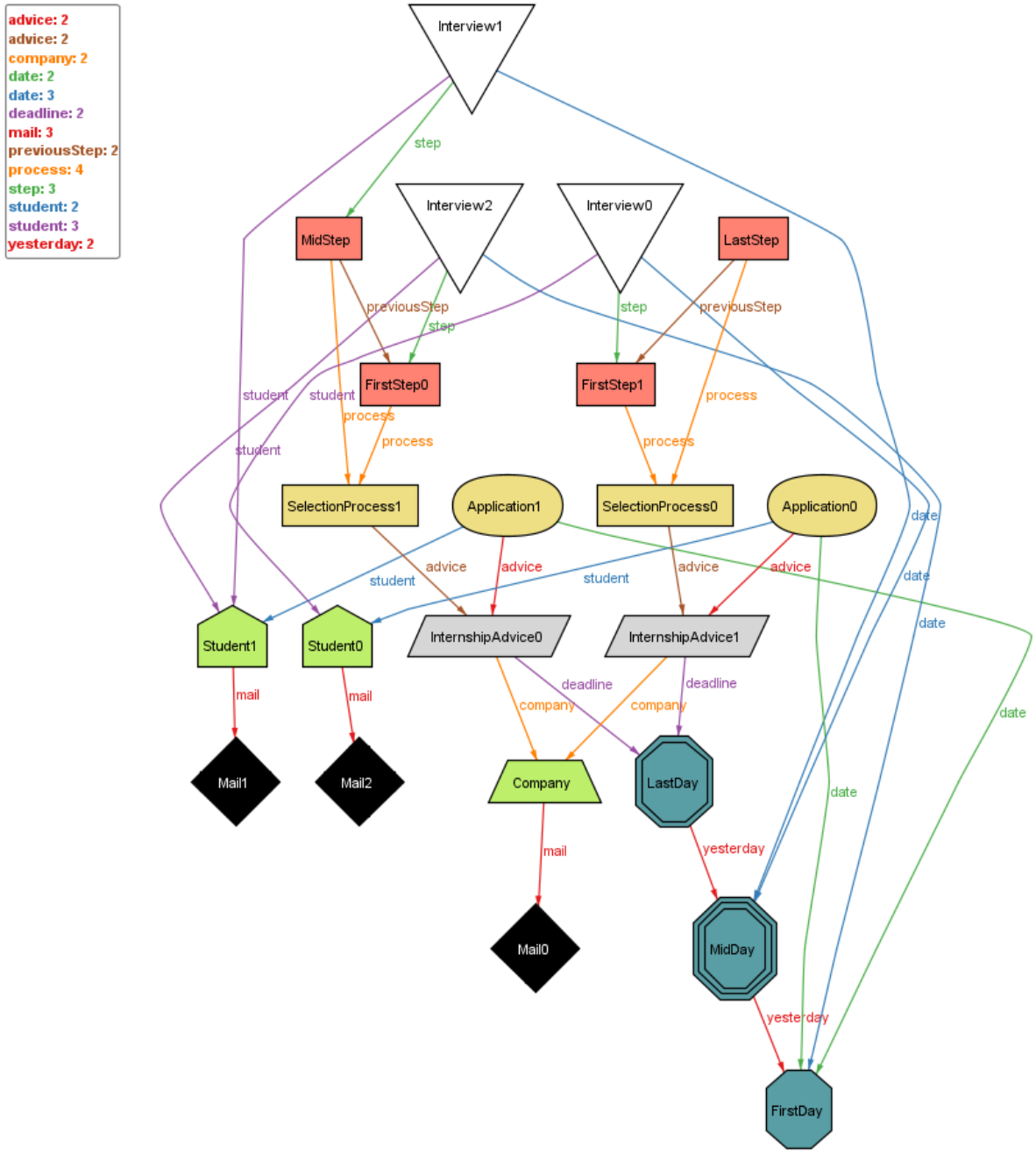
Although a selection process has without a doubt a dynamic behavior, we preferred to focus on modeling the constraints related to the process design:

```
//selection processes modeling
sig SelectionProcess
{
advice: one InternshipAdvice
}
sig SelectionStep
{
process: one SelectionProcess
}
sig FirstStep extends SelectionStep {}
sig MidStep extends SelectionStep
{
previousStep: one SelectionStep
}
one sig LastStep extends MidStep {}
```

```
fact selectionCalendar //facts to design the "selection process"
    {
        all f1,f2:FirstStep|(f1.process = f2.process) implies (f1 = f2)
        all d,d1:MidStep | (d!=d1) implies d.previousStep != d1.previousStep
        all d:SelectionStep | d in MidStep or d in FirstStep
        all d:LastStep | no d1:MidStep |  d1.previousStep = d
        all d:MidStep | d not in d.^(previousStep)
    }
sig Interview
    {
        date: one Date,
        step: one SelectionStep,
        student: one Student,
    }
```

```
fact selectionFacts
    {
        all s:Student |(all iv:Interview | (s = iv.student) iff (some a:Application|s in a.student and iv.step.process.advice=a.advice))
            //only student applied for an advice can take part into the selection process related to that advice
        all i:Interview| i.date  in i.step.process.advice.deadline.^yesterday
            //an interview can't be put in a date before the deadline advice
        all i,i1:Interview | (i.step.process = i1.step.process and i!=i1 and i.step in i1.step.^previousStep) implies (i.date in i1.date.^yesterday)
            //two interviews (of the same process) must have dates that "respect" the order of the selection process
        no s1,s2:SelectionProcess|s1!=s2 and s1.advice = s2.advice
            //like one selection process is related to an advice, an advice can't have more than one selection process related
        all s1:SelectionStep| s1 not in FirstStep implies s1.process = s1.previousStep.process
            //a selection steps "chain" must belong to one selectionStep
        all i,i1:Interview| (i!=i1 and i.student = i1.student) implies (i.step != i1.step)
            //a step has only one interview for each student that take part into the process
        all s:Student| (some i:Interview|i.student = s) implies (some a:Application|a.student = s)
            //a student enrolled in a selection process must have an application for the related advice
    }
```

These constraints ensure that the process designed by the company does not rise contradictions, such as interviews ordered differently that the related steps. Dynamic aspects related to selection processes such as score assignments are generally more interesting from a coherence point of view rather than logical contradictions (e.g. two identical open answers written by two different students should be evaluated in the same way in the context of the same selection process), a part for closed answers.



At the end, one of the main purposes of this static modeling is also to clarify the selection process structure (also visually).

References 5

	Andrea		Alessandro	
Date	Time	Category	Time	Category

Design and Additional Features

6 Page Design

6.1 Headings

So far, in this document I used two different styles for the chapter headings: one has the chapter name, a rule and, in the margin, the chapter number; the other has an image at the top of the page, and the chapter title is printed in a box (like for this chapter). There is one additional style, which I used only in the Chapter 7.3 (Appendix); there, the chapter title is enclosed in two horizontal rules, and the chapter number (or letter, in the case of the appendix) is above it.[1]

Every book is unique, so it makes sense to have different styles from which to choose. Actually, it would be awesome if whenever a `kao`-user designs a new heading style, he or she added it to the three styles already present, so that it will be available for new users and new books.

The choice of the style is made simple by the `\setchapterstyle` command. It accepts one option, the name of the style, which can be: "plain", "kao", "bar", or "lines".[2] If instead you want the image style, you have to use the command `\setchapterimage`, which accepts the path to the image as argument; you can also provide an optional parameter in square brackets to specify the height of the image. `\setchapterimage` automatically sets the chapter style to "bar" for that chapter (and also for subsequent chapters).

Let us make some examples. In this book, I begin a normal chapter with the lines:

```
1 \setchapterstyle{kao}
2 \setchapterpreamble[u]{\margintoc}
3 \chapter{Title of the Chapter}
4 \labch{title}
```

In Line 1 I choose the style for the title to be "kao". Then, I specify that I want the margin toc. The rest is ordinary administration in LaTeX, except that I use my own `\labch` to label the chapter. Actually, the `\setchapterpreamble` is a standard KOMA-Script one, so I invite you to read about it in the KOMA documentation. Once the chapter style is set, it holds until you change it.[3] Whenever I want to start a chapter with an image, I simply write:

```
1 \setchapterimage[7cm]{path/to/image.png} % Optionally specify the height
2 \setchapterpreamble[u]{\margintoc}
3 \chapter{Catchy Title} % No need to set a chapter style
4 \labch{catchy}
```

If you prefer, you can also specify the style at the beginning of the main document, and that style will hold until you change it again.

6.2 Headers & Footers

Headers and footers in KOMA-Script are handled by the `scrlayer-scrpage` package. There are two basic style: "scrheadings" and "plain.scrheadings". The former is used for normal pages, whereas the latter is used in title pages (those

1: To be honest, I do not think that mixing heading styles like this is a wise choice, but in this document I did it only to show you how they look.

2: Plain is the default LaTeX title style; the other ones are self explanatory.

where a new chapter starts, for instance) and, at least in this book, in the front matter. At any rate, the style can be changed with the `\pagestyle` command, *e.g.* `\pagestyle{plain.scrheadings}`.

In both styles, the footer is completely empty. In plain.scrheadings, also the header is absent (otherwise it wouldn't be so plain...), but in the normal style the design is reminiscent of the "kao" style for chapter titles.

To Do

The `twoside` class option is still unstable and may lead to unexpected behaviours. As always, any help will be greatly appreciated.

6.3 Table of Contents

Another important part of a book is the table of contents. By default, in `kaobook` there is an entry for everything: list of figures, list of tables, bibliographies, and even the table of contents itself. Not everybody might like this, so we will provide a description of the changes you need to do in order to enable or disable each of these entries. In the following Table 6.1, each item corresponds to a possible entry in the TOC, and its description is the command you need to provide to have such entry. These commands are specified in the attached style package,[4] so if you don't want the entries, just comment the corresponding lines.

Of course, some packages, like those for glossaries and indices, will try to add their own entries. In such cases, you have to follow the instructions specific to that package. Here, since we have talked about glossaries and notations in Chapter ??, we will briefly see how to configure them.

For the `glossaries` package, use the "toc" option when you load it: `\usepackage[toc]{glossaries}`. For `nomencl`, pass the "intoc" option at the moment of loading the package. Both `glossaries` and `nomencl` are loaded in the attached "packages" package.

Additional configuration of the table of contents can be performed through the packages `etoc`, which is loaded because it is needed for the margintocs, or the more traditional `tocbase`. Read the respective documentations if you want to be able to change the default TOC style.[5]

Table 6.1: Commands to add a particular entry to the table of contents.

Entry	Command to Activate
Table of Contents	`\setuptoc{toc}{totoc}`
List of Figs and Tabs	`\PassOptionsToClass{toc=listof}{\@baseclass}`
Bibliography	`\PassOptionsToClass{toc=bibliography}{\@baseclass}`

3: The `\margintoc` has to be specified at every chapter. Perhaps in the future this may change; it all depends on how this feature will be welcomed by the users, so keep in touch with me if you have preferences!

5: (And please, send me a copy of what you have done, I'm so curious!)

6.4 Paper Size

Recent versions of Kaobook support paper sizes different from the default A4. It is possible to pass the name of the paper as an option to the class, as we are accustomed for any other LaTeX class. For example, the class option `b5paper` would set the paper size to the B5 format.

We also support the paper sizes specified in this web page and some additional sizes requested by the users, with the option names specified in Table 6.2.

For instance, to use the "smallpocketpaper" add the correct description at the beginning of the documentclass instruction:

```
\documentclass[
        smallpocketpaper,
        fontsize=10pt,
        twoside=false,
        %open=any,
        secnumdepth=1,
]{kaobook}
```

Table 6.2: Some non-standard paper sizes supported by kaobook

13.5cm x 22.0cm	juvenilepaper
17.0cm x 17.0cm	smallphotopaper
21.0cm x 15.0cm	appendixpaper
17.0cm x 22.0cm	cookpaper
19.0cm x 27.0cm	illustratedpaper
17.0cm x 17.0cm	photopaper
16.0cm x 24.0cm	p4paper

6.5 Page Layout

Besides the page style, you can also change the width of the content of a page. This is particularly useful for pages dedicated to part titles, where having the 1.5-column layout might be a little awkward, or for pages where only figures, where it is important to exploit all the available space.

In practice, there are two layouts: "wide" and "margin". The former suppresses the margins and allocates the full page for contents, while the latter is the layout used in most of the pages of this book, including this one. The wide layout is also used automatically in the front and back matters.

To change page layout, use the `\pagelayout` command. For example, when I start a new part, I write:

```
\pagelayout{wide}
\addpart{Title of the New Part}
\pagelayout{margin}
```

Beyond these two basic layouts, it is also possible to finely tune the page layout by redefining the `\marginlayout` command. This command is called internally by the higher-level `\pagelayout`, and it is responsible for setting the width of the margins and of the text. The default definition is:

```
\newcommand{\marginlayout}{%
    \newgeometry{
        top=27.4mm,             % height of the top margin
        bottom=27.4mm,          % height of the bottom margin
        inner=24.8mm,           % width of the inner margin
        textwidth=107mm,        % width of the text
        marginparsep=8.2mm,     % width between text and margin
        marginparwidth=49.4mm,  % width of the margin
    }%
}
```

so if you want to, say, decrease the width of the margin while increasing the width of the text, you could write in the preamble of your document something like:

```
\renewcommand{\marginlayout}{%
    \newgeometry{
        top=27.4mm,             % height of the top margin
        bottom=27.4mm,          % height of the bottom margin
        inner=24.8mm,           % width of the inner margin
        textwidth=117mm,        % width of the text
        marginparsep=8.2mm,     % width between text and margin
        marginparwidth=39.4mm,  % width of the margin
```

4: In the same file, you can also choose the titles of these entries. In a later section, we will see how you can define your own floating environment, and endow it with an entry in the TOC.

```
    }%
}
```

where the text width has been increased by 10mm and the margin width has been decreased by 10mm.

Sometimes it is desirable to increase the width for just one or a few paragraphs; the `widepar` environment does that: wrap your paragraphs in this environment, and they will occupy the full width of the page.

6.6 Numbers & Counters

In this short section we shall see how dispositions, sidenotes and figures are numbered in the `kaobook` class.

By default, dispositions are numbered up to the section in `kaobook` and up to the subsection in `kaohandt`. This can be changed by passing the option `secnumdepth` to`kaobook` or `kaohandt` (e.g. 1 corresponds to section and 2 corresponds to subsections).

The sidenotes counter is the same across all the document, but if you want it to reset at each chapter, just uncomment the line

```
\counterwithin*{sidenote}{chapter}
```

in the `styles/style.sty` package provided by this class.

Figure and Table numbering is also per-chapter; to change that, use something like:

```
\renewcommand{\thefigure}{\arabic{section}.\arabic{figure}}
```

6.7 White Space

One of the things that I find most hard in LaTeX is to finely tune the white space around objects. There are not fixed rules, each object needs its own adjustment. Here we shall see how some spaces are defined at the moment in this class.

Attention! This section may be incomplete.

Space around sidenotes and citations marks

There should be no space before or after sidenotes and citation marks, like so:

sidenote[6]sidenote
citation[**James2013**]citation

Space around figures and tables

```
\renewcommand\FBaskip{.4\topskip}
\renewcommand\FBbskip{\FBaskip}
```

Space around captions

```
\captionsetup{
    aboveskip=6pt,
    belowskip=6pt
}
```

Space around displays (*e.g.* equations)

```
\setlength\abovedisplayskip{6pt plus 2pt minus 4pt}
\setlength\belowdisplayskip{6pt plus 2pt minus 4pt}
\abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
```

7 Mathematics and Boxes

7.1 Theorems

Despite most people complain at the sight of a book full of equations, mathematics is an important part of many books. Here, we shall illustrate some of the possibilities. We believe that theorems, definitions, remarks and examples should be emphasised with a shaded background; however, the colour should not be to heavy on the eyes, so we have chosen a sort of light yellow.[1]

Definition 7.1.1 *Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X, d).*

Definition 7.1.1 is very important. I am not joking, but I have inserted this phrase only to show how to reference definitions. The following statement is repeated over and over in different environments.

Theorem 7.1.1 *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

Proposition 7.1.2 *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

Lemma 7.1.3 *A finite intersection[a] of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

[a] I'm a footnote

You can safely ignore the content of the theorems...I assume that if you are interested in having theorems in your book, you already know something about the classical way to add them. These example should just showcase all the things you can do within this class.

Corollary 7.1.4 (Finite Intersection, Countable Union) *A finite intersection of open sets of (X, d) is an open set of (X, d), i.e τ_d is closed under finite intersections. Any union of open sets of (X, d) is an open set of (X, d).*

Proof. The proof is left to the reader as a trivial exercise. Hint: Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. □

Definition 7.1.2 *Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X, d).*

Example 7.1.1 Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such that $B(x, r) \subset U$. We call the topology associated to d the set τ_d of all the open subsets of (X, d).

Remark 7.1.1 Let (X, d) be a metric space. A subset $U \subset X$ is an open set if, for any $x \in U$ there exists $r > 0$ such

6: This paragraph can be used to diagnose any problems: if you see whitespace around sidenotes or citation marks, probably a % sign is missing somewhere in the definitions of the class macros.

1: The boxes are all of the same colour here, because we did not want our document to look like Harlequin.

that $B(x, r) \subset U$. We call the topology associated to d the set τ_{d} of all the open subsets of (X, d).

As you may have noticed, definitions, example and remarks have independent counters; theorems, propositions, lemmas and corollaries share the same counter.

Remark 7.1.2 Here is how an integral looks like inline: $\int_a^b x^2 dx$, and here is the same integral displayed in its own paragraph:

$$\int_a^b x^2 dx$$

There is also an environment for exercises.

Exercise 7.1.1 Prove (or disprove) the Riemann hypothesis.

We provide one package for the theorem styles: kaotheorems.sty, to which you can pass the `framed` option you do want coloured boxes around theorems, like in this document.[2] You may want to edit this files according to your taste and the general style of the book. However, there is an option to customise the background colour of the boxes if you use the `framed` option: when you load this package, you can pass it the `background=mycolour` option (replace "mycolour" with the actual colour, for instance, "red!35!white"). This will change the colour of all the boxes, but it is also possible to override the default colour only for some elements. For instance, the `propositionbackground=mycolour` option will change the colour for propositions only. There are similar options for theorem, definition, lemma, corollary, remark, and example.

7.2 Boxes & Custom Environments [3]

Say you want to insert a special section, an optional content or just something you want to emphasise. We think that nothing works better than a box in these cases. We used `mdframed` to construct the ones shown below. You can create and modify such environments by editing the provided file environments.sty.

Title of the box

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

If you set up a counter, you can even create your own numbered environment.

Comment 7.2.1

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

[2] You can even insert footnotes inside the theorem environments; they will be displayed at the bottom of the box. Here is a random equation, just because we can:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{}{a_2 +}}$$

[3] Notice that header, the name of this achieved this with the optional argument of the `section` command.

7.3 Experiments

It is possible to wrap marginnotes inside boxes, too. Audacious readers are encouraged to try their own experiments and let me know the outcomes.

I believe that many other special things are possible with the `kaobook` class. During its development, I struggled to keep it as flexible as possible, so that new features could be added without too great an effort. Therefore, I hope that you can find the optimal way to express yourselves in writing a book, report or thesis with this class, and I am eager to see the outcomes of any experiment that you may try.

title of margin note As a note inside a kaobox. (Actually, kaobox inside a marginnote!)

2: The styles without `framed` are not showed, but actually the only difference is that they don't have the yellow boxes.

Appendix

A

Heading on Level 0 (chapter)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1 Heading on Level 1 (section)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1.1 Heading on Level 2 (subsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 3 (subsubsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 4 (paragraph) Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.2 Lists

A.2.1 Example for list (itemize)

- First item in a list
- Second item in a list
- Third item in a list
- Fourth item in a list
- Fifth item in a list

Example for list (4*itemize)

- First item in a list
  - First item in a list
    - First item in a list
      - First item in a list
      - Second item in a list
    - Second item in a list
  - Second item in a list
- Second item in a list

A.2.2 Example for list (enumerate)

1. First item in a list
2. Second item in a list
3. Third item in a list
4. Fourth item in a list
5. Fifth item in a list

Example for list (4*enumerate)

1. First item in a list
   a) First item in a list
      i. First item in a list
         A. First item in a list
         B. Second item in a list
      ii. Second item in a list
   b) Second item in a list
2. Second item in a list

A.2.3 Example for list (description)

First item in a list
Second item in a list
Third item in a list
Fourth item in a list
Fifth item in a list

Example for list (4*description)

First item in a list

First item in a list

First item in a list

First item in a list

Second item in a list

Second item in a list

Second item in a list

Second item in a list

B

Fonts Testing

B.1 Font Sizes

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

B.2 Font Families

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

The quick brown fox jumps over the lazy dog. Medium.

The quick brown fox jumps over the lazy dog. Bold.

The quick brown fox jumps over the lazy dog. Upright.

The quick brown fox jumps over the lazy dog. Italics.

The quick brown fox jumps over the lazy dog. Slanted.

The quick brown fox jumps over the lazy dog. Small Caps.

```
Hello, here is some text without a meaning. This text should show what a printed text will
look like at this place. If you read this text, you will get no information. Really? Is
there no information? Is there a difference between this text and some nonsense like ``Huardest
gefburn''? Kjift -- not at all! A blind text like this gives you information about the selected
font, how the letters are written and an impression of the look. This text should contain
all letters of the alphabet and it should be written in of the original language. There
is no need for special content, but the length of words should match the language.
```

The quick brown fox jumps over the lazy dog. Medium.

The quick brown fox jumps over the lazy dog. Bold.

The quick brown fox jumps over the lazy dog. Upright.

The quick brown fox jumps over the lazy dog. Italics.

The quick brown fox jumps over the lazy dog. Slanted.

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. SMALL CAPS.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

The quick brown fox jumps over the lazy dog. Medium.

The quick brown fox jumps over the lazy dog. Bold.

The quick brown fox jumps over the lazy dog. Upright.

The quick brown fox jumps over the lazy dog. Italics.

The quick brown fox jumps over the lazy dog. Slanted.

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. SMALL CAPS.

Greek Letters with Pronunciations

Character	Name	Character	Name
α	alpha *AL-fuh*	ν	nu *NEW*
β	beta *BAY-tuh*	ξ, Ξ	xi *KSIGH*
γ, Γ	gamma *GAM-muh*	o	omicron *OM-uh-CRON*
δ, Δ	delta *DEL-tuh*	π, Π	pi *PIE*
ϵ	epsilon *EP-suh-lon*	ρ	rho *ROW*
ζ	zeta *ZAY-tuh*	σ, Σ	sigma *SIG-muh*
η	eta *AY-tuh*	τ	tau *TOW (as in cow)*
θ, Θ	theta *THAY-tuh*	υ, Υ	upsilon *OOP-suh-LON*
ι	iota *eye-OH-tuh*	ϕ, Φ	phi *FEE, or FI (as in hi)*
κ	kappa *KAP-uh*	χ	chi *KI (as in hi)*
λ, Λ	lambda *LAM-duh*	ψ, Ψ	psi *SIGH, or PSIGH*
μ	mu *MEW*	ω, Ω	omega *oh-MAY-guh*

Capitals shown are the ones that differ from Roman capitals.