POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

User-Centric Agile Redesign of the Thesis Management System for Politecnico di Torino

Supervisors
Prof. Marco TORCHIANO
Prof. Riccardo COPPOLA

Candidate
Sylvie MOLINATTO

March/April 2025

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context

The use of online educational services is extremely widespread, and the growing reliance on these tools necessitates the development of platforms that are not only modern and easily maintainable, but also effectively meet users' specific needs.

Unlike most other universities, Politecnico di Torino has an internal Information Technology department called ISIAD (Infrastructure, IT Services, and Digital Administration), which in recent years, through collaboration with students and teachers, has undertaken a process of redesigning the digital services offered. In this context, a new version of the mobile application for students has been developed and released. Similarly, a process of renewing the teaching portal and the university website in general is underway, as the current portal, while functional, is now obsolete, having been built with technologies that are no longer widely used.

For this reason, the project for the Master's course 'Software Engineering II' focused on redesigning the thesis management system within the teaching portal using agile methods. The group with the best outcome was offered a collaboration with the ISIAD department to renovate the thesis management section, with the goal of creating a system that meets the different needs of students, teachers, and administrative staff.

This thesis emerges from that collaboration, with the aim of documenting every stage of the process and highlighting the techniques and strategies used to achieve the final product.

1.2 Goal of the thesis

The goal of this thesis is to analyze, design, and develop a new integrated thesis management procedure within the Politecnico di Torino Teaching Portal.

The current procedure, developed some time ago, requires a revision. As a consequence, this thesis work aims to develop an improved version of this procedure, using user experience design techniques and managing the project with agile development methodologies.

The final outcome must meet high-quality standards and be integrated into the portal.

1.3 Document structure

The following document is divided into various chapters, each representing a different project phase, from preliminary analysis to the validation of the final product. The structure guides the reader through the various steps of the project. Each chapter is briefly described below:

- **Preliminary Analysis:** description of the initial examination and evaluation of the project requirements, constraints, and current state of the system. It focuses on the analysis of the problems that need to be addressed during development.
- **User Needs Identification:** explanation of how user needs and expectations were identified. Highlights the insights collected from the interviews that impacted the design and development process.
- **Prototyping:** Information Architecture analysis and improvements followed by an outlook of the different prototypes that were developed, together with the tools used to design them.
- **API Design:** overview of the API design process, focusing on how the system architecture and specifications were formalized according to the OpenAPI standard. It includes the definition of endpoints, request/response formats, and the relative documentation.
- **Development:** detailed description of the implementation phase, where the actual product is built based on the previous work. This chapter includes

information on the project management, project structure, technological choices, and the final outcome.

- **Verification and Validation:** explanation of the process used to verify and validate the system's functionality and usability through various testing phases.

Each phase of the project is interdependent, with the outcomes of one phase directly influencing the next. It is important to recognize that while the implementation phase is crucial, it represents only a small part of the entire process. The success of the final product is strongly dependent on the completeness and efficacy of the preceding steps.

1.4 Method

This project adopts a user-centered approach that focuses on keeping the users' needs at the center of each work phase. For this reason, each step (from the preliminary analysis to the final testing phase) directly involved end users.

The project involves the transition of an existing system, built with pure JavaScript, to a modern implementation using React.

The final product will be released under an open source license since, for some time now, Politecnico di Torino has decided to make use of this development mode to benefit from the contribution of the entire student community and ensure transparency on the entire life cycle of the development process.

The working methodology was divided into two main phases:

1. **Analysis Phase**

   The primary objective of the analysis phase was to understand the entire thesis management process (AS-IS) and how each operation was carried out through the current interface.

   In order to draw a flow diagram of all the various steps that make up the life cycle of a thesis, some informal meetings were held with faculty and administrative staff to gather all the necessary information. These meetings allowed the main processes to be summarized through the Business Process Model and Notation (BPMN).

   Next came the interface analysis, again in the presence of the main users,

who provided important feedback to be taken into account in the next steps. As a result of this further analysis, a document was prepared with various screenshots of the interfaces.

By analyzing the process flow and interface analysis, the main critical processes and interface issues were identified and documented in detail.

Finally, before starting with the second phase, interviews were conducted with some students to identify their main needs together with any suggestions on some missing or inadequately structured features.

2. Design and Development Phase

The goal of the second phase was to use the knowledge and feedback received to implement the new system (TO-BE).

Initially, an attempt was made to formalize the information architecture of the old portal and then propose a new arrangement of information within the 'Thesis' section. Specifically, the general IA structure of the teaching portal had already been reorganized by some students for their thesis project, and based on their work [1], the content organization for thesis management was defined.

First prototype interfaces were then sketched on Figma, to identify the most suitable type of visualization. In terms of development, the primary goal was to redesign the thesis proposal consultation section, which was particularly problematic as revealed in the interviews.

After selecting the most reliable prototype, the development phase began. Development was carried out iteratively and incrementally, increasing the functionality and final project quality with each iteration. In parallel with interface development, the necessary API specifications were designed, and a back-end was implemented to test the APIs and load an adequate amount of thesis proposals.

Once the product was considered adequate, the final testing phase began to verify and validate the effectiveness of the new interface.

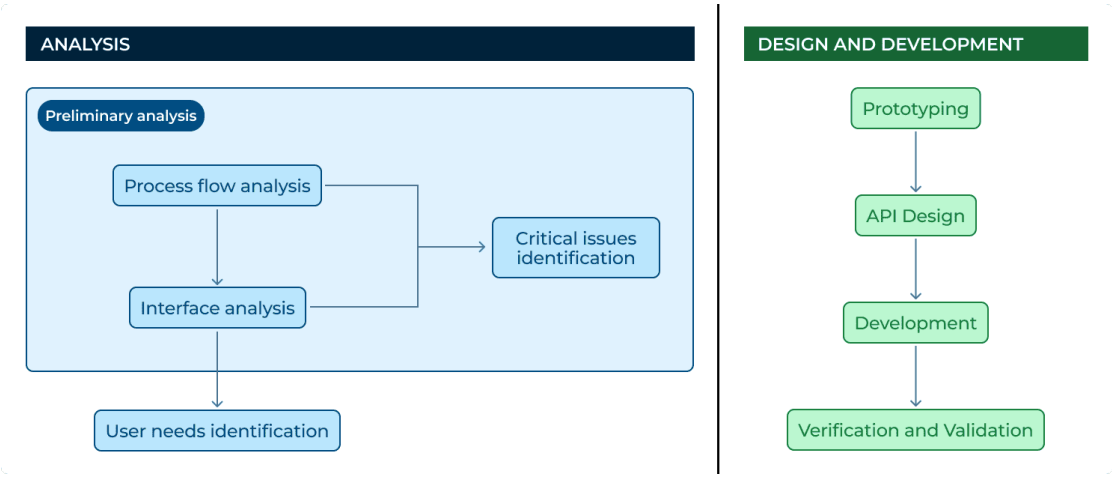


Figure 1.1: Method resume diagram

Chapter 2

Preliminary Analysis

2.1 Process flow analysis

To obtain a complete understanding of the thesis management process, our team consulted with domain experts who are familiar with the operations of the system. Through these meetings, we traced the flows of the entire process using BPMN (Business Process Model and Notation) diagrams. The process was divided into three main stages:

1. **Thesis application:** this stage includes the initial submission and approval of a thesis topic, involving both students and academic supervisors.
2. **Thesis development:** this phase focuses on managing the thesis once it has been started. It involves handling student requests to cancel the thesis or make necessary changes.
3. **Thesis conclusion confirmation request:** this stage involves the submission and approval of the completed thesis (with additional fields such as the abstract, summary, embargo, if any, and license chosen) and several other actions: the execution of the anti-plagiarism software by the thesis supervisor and the final steps necessary to complete the registration for the graduation session (filling out the questionnaires and registering for the final exam with the associated payment of the fee).

These diagrams provided a clear visualization of workflows, enabling the team to identify key touchpoints and potential bottlenecks in the existing process.

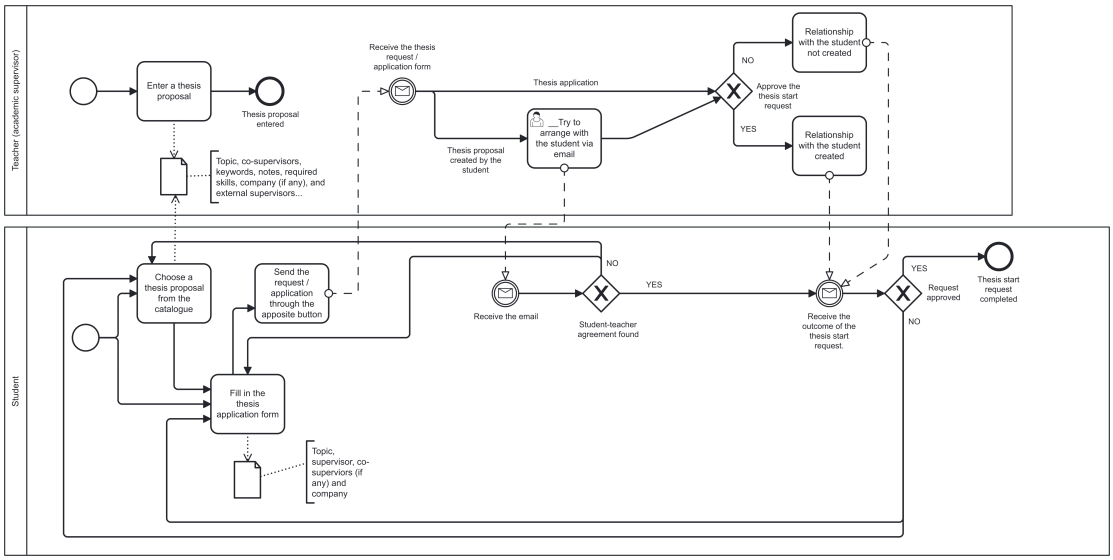


Figure 2.1: Thesis application process

As can be seen in Figure 2.1, the thesis application process carried out by the student can be done in two different ways: the student can choose a thesis from the catalogue of proposals entered by faculty members, or he/she can propose his/her own idea. In either case, a thesis application is sent to the selected thesis advisor who, in turn, can decide to approve or reject the request, possibly discussing it with the student first. The thesis application consists of:

- **Topic** on which the thesis will focus
- **Supervisor**, i.e., the faculty member who will supervise the work
- **Co-supervisors**, if any: these can be either faculty or external persons as long as the external referents, if any, are already registered in the system (otherwise, it is necessary to ask the supervisor to contact the Careers Office so that they can be registered)
- **Company** at which the thesis will be conducted (if any)

Once the request has been approved, the student can begin to carry out the work for writing the paper.

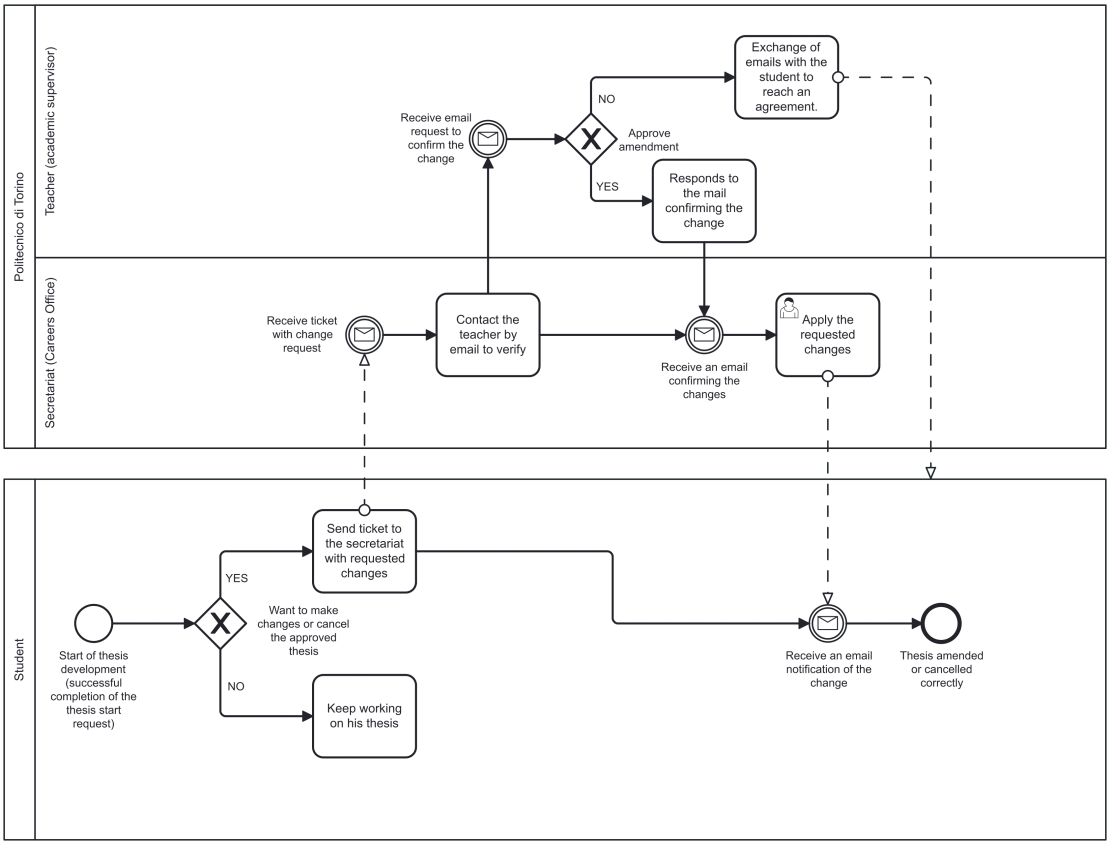


Figure 2.2: Thesis development process

In Figure 2.2, it can be seen that if the student wishes to cancel or modify the current thesis during the development phase, the only way to do so is by sending a ticket to the administrative office requesting the necessary changes. The changes can be implemented once the administrative staff has verified that the professor is aware and in agreement.

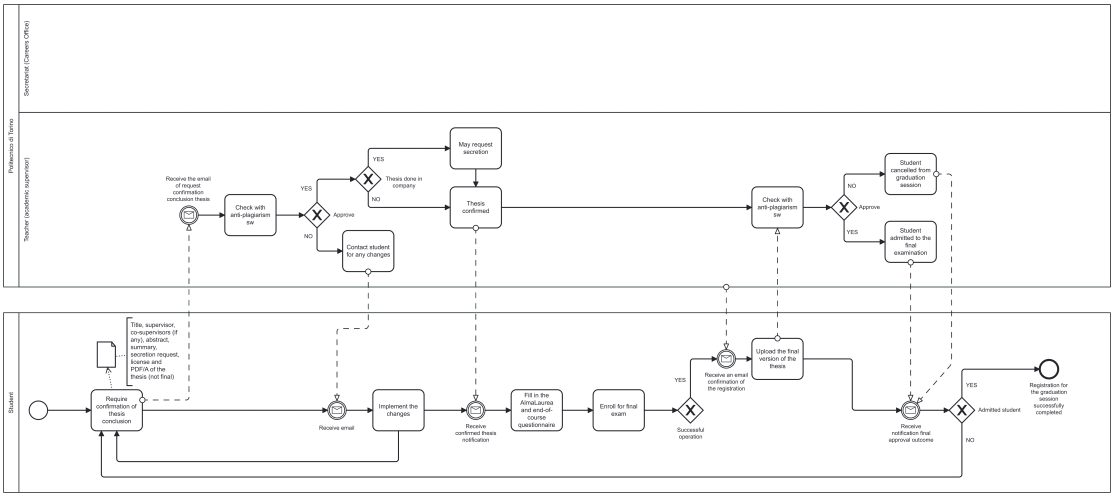


Figure 2.3: Thesis conclusion confirmation request process

Finally, Figure 2.3 shows the process by which the student can make the thesis conclusion confirmation request, that is, the procedure by which the professor confirms the paper, allowing the student to register for the desired graduation session. The thesis conclusion request consists of:

- **Final thesis title** and its English translation
- **Supervisor**
- **Co-supervisors**
- **Final abstract** and related English translation
- **Summary** (not required for all courses of study)
- **Sustainable Development Goals** (SDGs)
- **Authorization/non-authorization for thesis consultation**
- **License**
- **Thesis file** in PDF/A format

After the student submits the thesis conclusion confirmation request, the anti-plagiarism software is automatically launched, and the supervisor can view the report. Based on the metrics in the report and his personal assessment, the teacher

can accept or reject the request. In case the request is accepted, the student can proceed with filling out the questionnaires and finally registering for the final exam. Once the student is registered for the graduation session (i.e., has also made payment of the prescribed amount), he has up to 7 days before the beginning of thesis oral defences to upload his final paper. This paper will also be submitted, like the first one, to the anti-plagiarism check and, if successful, the student is finally admitted to the graduation session.

2.2 Interface analysis

After understanding and analyzing the entire process, the focus shifted to how each action was carried out through the various interfaces. Consequently, through a series of informal meetings with some teachers and some secretaries, the three portal interfaces intended for the three types of users (students, faculty, and administrative staff) were viewed to identify any problems, including through the direct experience of the users. The purpose was to detect any potential issues by drawing on the users' own experiences. The meetings resulted in the creation of a document that brings together all the screenshots from the various interfaces.

The student's interface is available at the following link: `https://www.figma.com/design/gShJl1GJjDJChuDxJsEiXV/Interfaces?m=auto&t=SY4jJO57GeQ35JZC-1`

2.3 Critical issues identification

Based on the gathered information, a document was drafted with all critical issues that emerged, divided between process-related and interface-related problems. Interface-related problems have been further classified according to the three types of user interfaces analyzed: those for students, faculty, and administrative staff.

The full document is available at the following link: `https://www.dropbox.com/scl/fi/8yp6b877pa3y1ct0j3pj0/Report-2.docx?rlkey=fqhj0nwrenk858zlwiukzteko&st=4l3ez3r0&dl=0`

2.3.1 Process-related issues

1. **Thesis cancellation and modification process**

   Currently, if a student has received approval for a thesis proposal and has formalized the start of the work, they must send a ticket to the administration for any modification or deletion of the current thesis. The secretary, in turn, must seek confirmation from the thesis advisor before making the necessary changes. This procedure is manual and time-consuming, with the potential for errors and delays.

2. **Thesis confidentiality procedure**

   The request for thesis secrecy must be made concurrently with the request for confirmation of thesis completion. However, the appropriate field often goes unnoticed, leading students already enrolled in the graduation session to belatedly request an embargo of the thesis. The secretary's office must then request confirmation from the thesis advisor, and if there is no agreement between the student and faculty member, a critical situation is created, since registration for the graduation session has already been made.

3. **Procedure for signing the final thesis document**

   Students must upload the final version of their thesis paper within 7 days before the start of thesis discussions. After that, the teacher should review the document using anti-plagiarism software to check its integrity and proceed with approval and signature. However, in some cases, the supervisor confirms the validity of the thesis and allows the student to graduate without actually signing the document. The absence of control over the faculty member's signature not only causes formal and legal issues but also precludes the thesis from being published on Webthesis.

4. **Procedure for recording the final graduation grade**

   Following the graduation session, the administrative staff publishes the candidates' final grades. However, sometimes it happens that the secretaries cannot proceed with the registration of the graduation grade because it appears that the student still has an outstanding loan with the library. Therefore, it is necessary to manually contact the candidate and urge him/her to return any books before the graduation grade can be recorded.

2.3.2 Interface-related issues

Student interface

1. **Thesis filtering**

   Although some filters are present, they are implemented in a manner that is not very intuitive and does not allow students to filter theses effectively. The system does not currently have the ability to narrow down the search for theses using particular parameters such as the thesis type, publication date, deadline, title, description, or any combination of these. Because of this, looking for theses is especially difficult and time-consuming.

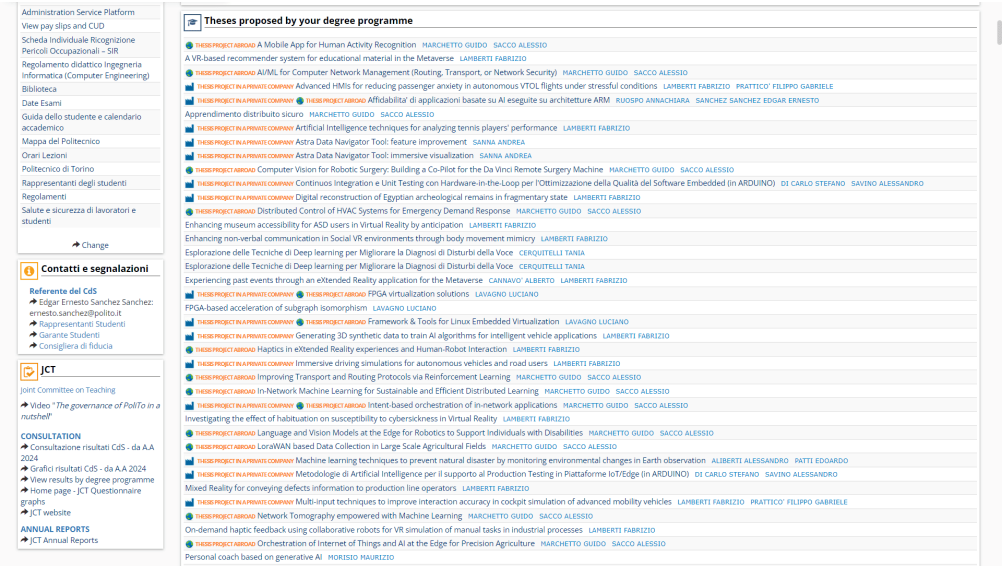


Figure 2.4: Personal thesis proposals list

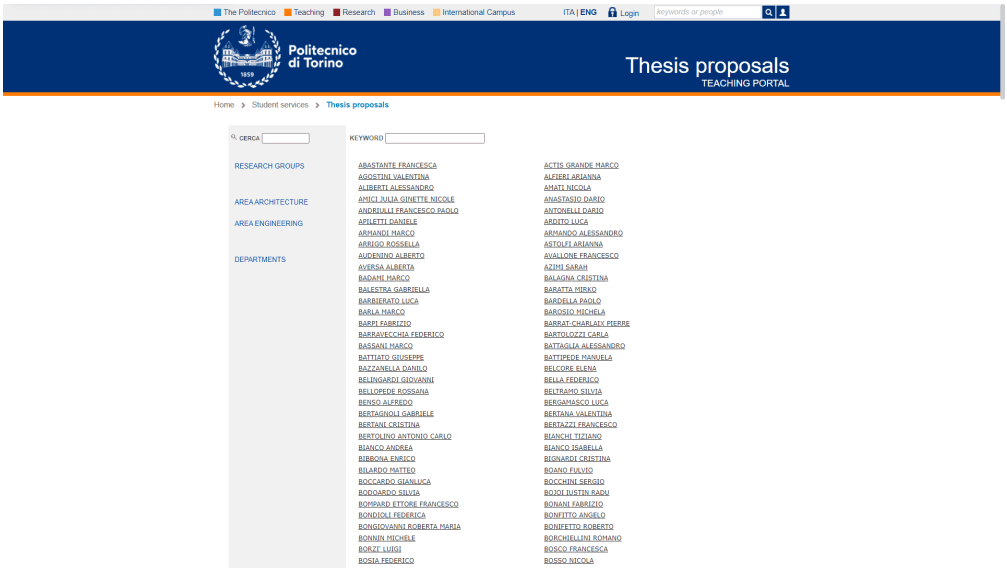


Figure 2.5: Public thesis proposals search page

2. Thesis status display

The dedicated page does not provide a very clear indication of the thesis's progress once it has been started. Students may find it challenging to keep track of their progress and understand the following steps and deadlines of their work.

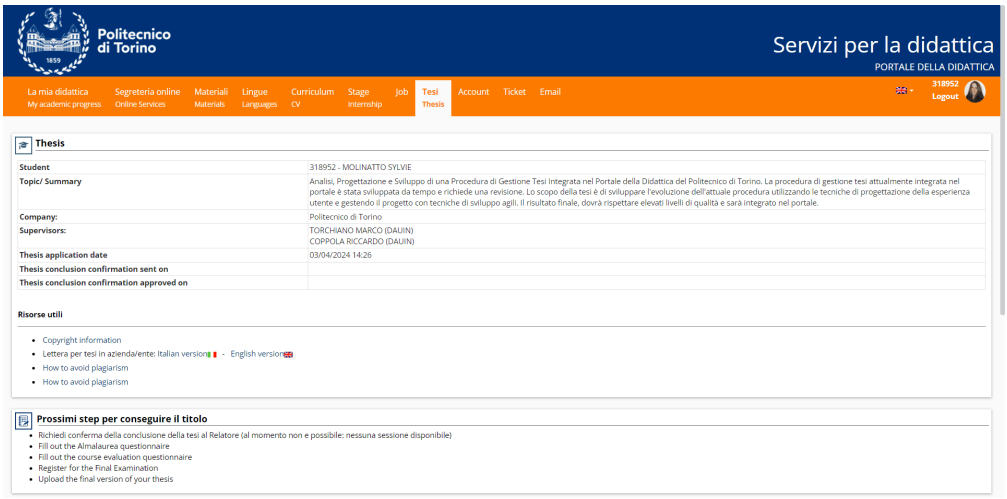


Figure 2.6: Thesis details page

3. Forum and navigation errors

When accessing the Forum on the current thesis page, the tabs at the top show the typical sections of a course (namely 'Announcements', 'Materials', 'Forum' - the selected tab, 'Assignments', and 'Virtual Classroom'). However, clicking on them redirects users to pages about other courses or pages displaying errors such as 'Invalid, expired, or ended session'. Also, when posting a new topic on the forum, the following error appears: 'Bad Request calling sviluppo.pkg_forum.espandi_extra'.

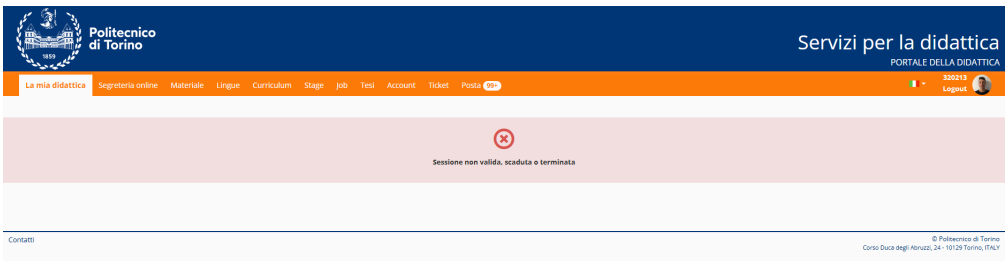


Figure 2.7: Thesis forum navigation error

Figure 2.8: Thesis forum error

4. **Incorrect email addresses** On the thesis page, there is a box called 'Contact' that allows to select the various supervisors to send them a direct email. However, the faculty members' emails show @studenti.polito.it, which is incorrect.

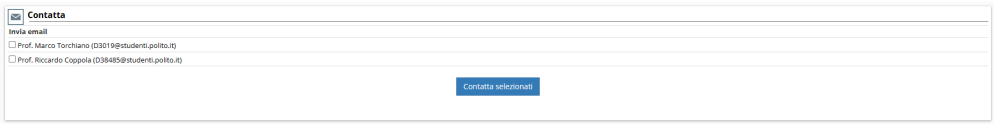


Figure 2.9: Email error

5. **Completion of the AlmaLaurea questionnaire**

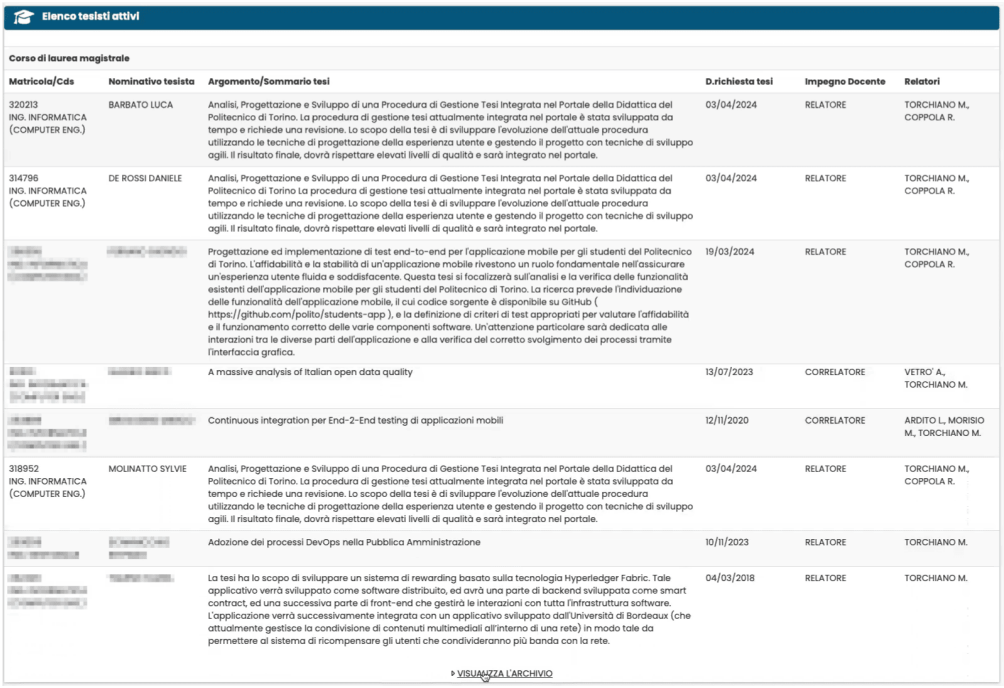
   During the completion of the AlmaLaurea questionnaire, there is often language confusion among international students. These students frequently mistakenly select 'Master' instead of 'Laurea Magistrale'. This error prevents the administration from proceeding with the registration for the graduation

session, as it appears that the student has not correctly completed the questionnaire. To solve this, the student must contact the AlmaLaurea helpline to unlock the questionnaire, allowing them to correct the error and complete their registration for the graduation session.

Teacher interface

1. **Lack of filters for thesis**

   Currently, faculty members do not have the option to filter the active theses they supervise on the portal. The inability to search for specific theses based on factors like topic or student assigned to the thesis limits supervisors' ability to efficiently manage and monitor the theses they are supervising. The problem becomes particularly visible when there are a lot of active theses being supervised simultaneously.



Elenco tesisti attivi

Corso di laurea magistrale

Matricola/Cds	Nominativo tesista	Argomento/Sommario tesi	D.richiesta tesi	Impegno Docente	Relatori
320213 ING. INFORMATICA (COMPUTER ENG.)	BARBATO LUCA	Analisi, Progettazione e Sviluppo di una Procedura di Gestione Tesi Integrata nel Portale della Didattica del Politecnico di Torino. La procedura di gestione tesi attualmente integrata nel portale è stata sviluppata da tempo e richiede una revisione. Lo scopo della tesi è di sviluppare l'evoluzione dell'attuale procedura utilizzando le tecniche di progettazione della esperienza utente e gestendo il progetto con tecniche di sviluppo agili. Il risultato finale, dovrà rispettare elevati livelli di qualità e sarà integrato nel portale.	03/04/2024	RELATORE	TORCHIANO M., COPPOLA R.
314796 ING. INFORMATICA (COMPUTER ENG.)	DE ROSSI DANIELE	Analisi, Progettazione e Sviluppo di una Procedura di Gestione Tesi Integrata nel Portale della Didattica del Politecnico di Torino La procedura di gestione tesi attualmente integrata nel portale è stata sviluppata da tempo e richiede una revisione. Lo scopo della tesi è di sviluppare l'evoluzione dell'attuale procedura utilizzando le tecniche di progettazione della esperienza utente e gestendo il progetto con tecniche di sviluppo agili. Il risultato finale, dovrà rispettare elevati livelli di qualità e sarà integrato nel portale.	03/04/2024	RELATORE	TORCHIANO M., COPPOLA R.
[redacted]	[redacted]	Progettazione ed implementazione di test end-to-end per l'applicazione mobile per gli studenti del Politecnico di Torino. L'affidabilità e la stabilità di un'applicazione mobile rivestono un ruolo fondamentale nell'assicurare un'esperienza utente fluida e soddisfacente. Questa tesi si focalizzerà sull'analisi e la verifica delle funzionalità esistenti dell'applicazione mobile per gli studenti del Politecnico di Torino. La ricerca prevede l'individuazione delle funzionalità dell'applicazione mobile, il cui codice sorgente è disponibile su GitHub (https://github.com/polito/students-app), e la definizione di criteri di test appropriati per valutare l'affidabilità e il funzionamento corretto delle varie componenti software. Un'attenzione particolare sarà dedicata alle interazioni tra le diverse parti dell'applicazione e alla verifica del corretto svolgimento dei processi tramite l'interfaccia grafica.	19/03/2024	RELATORE	TORCHIANO M., COPPOLA R.
[redacted]	[redacted]	A massive analysis of Italian open data quality	13/07/2023	CORRELATORE	VETRO' A., TORCHIANO M.
[redacted]	[redacted]	Continuous integration per End-2-End testing di applicazioni mobili	12/11/2020	CORRELATORE	ARDITO L., MORISIO M., TORCHIANO M.
318952 ING. INFORMATICA (COMPUTER ENG.)	MOLINATTO SYLVIE	Analisi, Progettazione e Sviluppo di una Procedura di Gestione Tesi Integrata nel Portale della Didattica del Politecnico di Torino. La procedura di gestione tesi attualmente integrata nel portale è stata sviluppata da tempo e richiede una revisione. Lo scopo della tesi è di sviluppare l'evoluzione dell'attuale procedura utilizzando le tecniche di progettazione della esperienza utente e gestendo il progetto con tecniche di sviluppo agili. Il risultato finale, dovrà rispettare elevati livelli di qualità e sarà integrato nel portale.	03/04/2024	RELATORE	TORCHIANO M., COPPOLA R.
[redacted]	[redacted]	Adozione dei processi DevOps nella Pubblica Amministrazione	10/11/2023	RELATORE	TORCHIANO M.
[redacted]	[redacted]	La tesi ha lo scopo di sviluppare un sistema di rewarding basato sulla tecnologia Hyperledger Fabric. Tale applicativo verrà sviluppato come software distribuito, ed avrà una parte di backend sviluppata come smart contract, ed una successiva parte di front-end che gestirà le interazioni con tutta l'infrastruttura software. L'applicazione verrà successivamente integrata con un applicativo sviluppato dall'Università di Bordeaux (che attualmente gestisce la condivisione di contenuti multimediali all'interno di una rete) in modo tale da permettere al sistema di ricompensare gli utenti che condivideranno più banda con la rete.	04/03/2018	RELATORE	TORCHIANO M.

VISUALIZZA L'ARCHIVIO

Figure 2.10: Active thesis list

2. **Publication of incomplete thesis proposals**

   Faculty members can publish a thesis proposal without filling out any mandatory fields. This may result in the creation of incomplete and uninformative

proposals.

3. **Poor visibility of thesis request notifications**

   Notifications related to thesis requests are not easily noticeable on the portal, as they are only found within the designated section ('Thesis Management'). There is no external indicator to alert users to new notifications within the section, making it easy to overlook new requests unless regularly checking the page.

Figure 2.11: Thesis Management section

4. **Problematic announcements section**

   Currently, in the 'Announcements' section, all notifications display the current date, which is not intuitive since these may be outdated alerts.

5. **Colour contrast**

   In the 'Thesis Management' section, there is the list of thesis requests submitted by students. However, when hovering over a thesis entry, the contrast between the text colour and the background is insufficient. This lack of contrast makes the text difficult to read, which may cause problems in swiftly recognizing and reviewing specific thesis demands. Poor contrast can reduce accessibility, particularly for users with visual impairments, and can lead to a frustrating user experience overall.

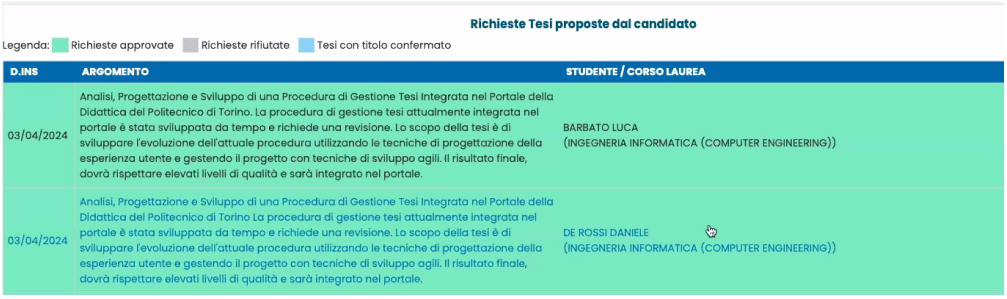


Figure 2.12: Thesis requests submitted by students

6. **Overlapping elements in the sidebar**

   In certain sections of the teacher interface, the icons and text in the sidebar overlap unnecessarily, even though there is sufficient space to display both

elements side by side. This overlap creates a cluttered appearance and reduces the readability and usability of the interface, despite the available space for proper alignment.

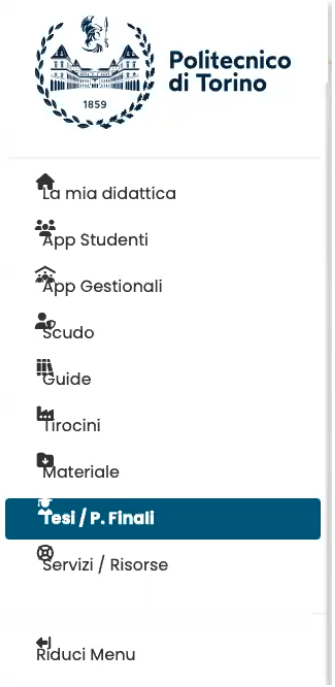


Figure 2.13: Sidebar error

Administrative staff interface

1. **Difficulty accessing transactions for new users**

   Currently, administrative staff use a series of transactions (such as GTESI – Thesis Management, ISEFI – Final Exam Registration, GESOS – Management of Conditional Registrations for the Graduation Exam) to view information regarding a student's thesis process. Accessing these transactions requires entering the transaction name in a text box that provides some suggestions. While this method is not problematic for experienced users, it can be unintuitive for first-time users, as it requires memorizing the names of the transactions.

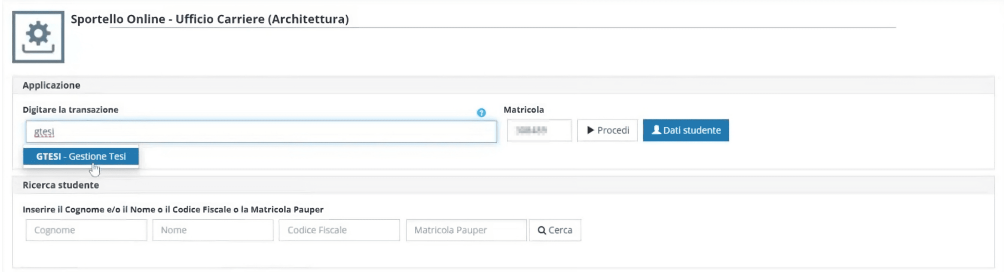


Figure 2.14: Transaction access form

2. Misleading button labels

In the GESOS transaction interface, it is possible to view conditional registrations for the graduation exam. The system automatically removes registrations for which the condition has been lifted, but administrative staff have the option to manually lift the condition using two buttons: 'Proceed with Registration' to confirm the candidate's final registration for the graduation session and 'Remove Suspension' to unregister the student due to lack of requirements. These labels are unclear for less experienced users, compromising the usability of the interface.

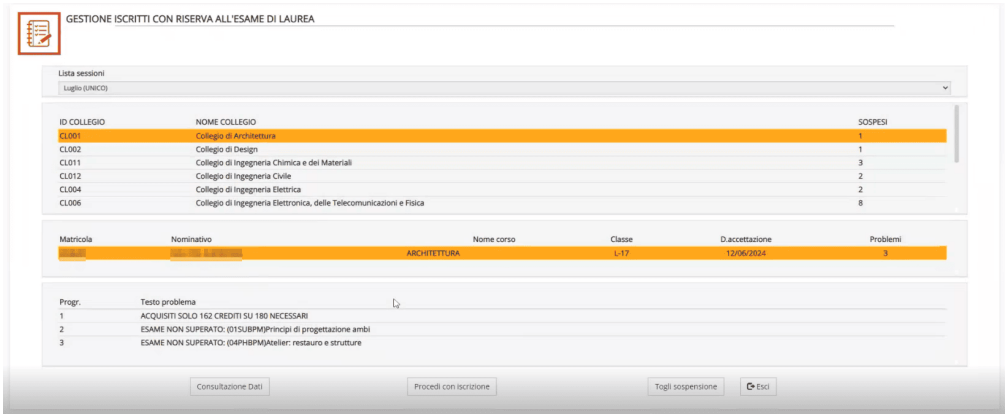


Figure 2.15: GESOS transaction interface

3. Colour contrast

In the GESOS transaction, when a candidate's conditional registration for the graduation session is automatically lifted, the text in the candidate's row turns green. However, due to the orange background, it becomes difficult to perceive the change and read the text clearly.

Figure 2.16: GESOS transaction interface if student's reserve is lifted

4. Lack of change history

Currently, the administration does not have an effective history of the various changes made to a thesis. To reconstruct the complete history of a thesis, staff must refer to various tickets, making the process complicated and time-consuming.

Chapter 3

User Needs Identification

User-Centered Design (UCD) is a design methodology that prioritizes the end user's needs throughout the design and development process to make systems more approachable and user-friendly. Understanding the end users and their context is the first stage in UCD's four-step iterative process, which concludes with a solution that needs to be assessed. After each evaluation, it can start a new iteration to achieve a solution that better fits users' needs, as can be seen in the figure below [2].

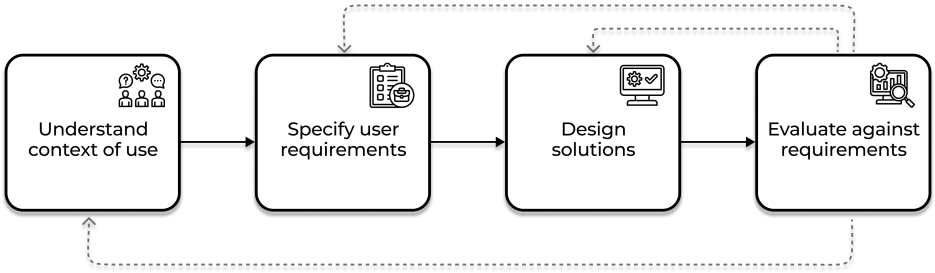


Figure 3.1: User-Centered Design process

A user-centric strategy was taken from the beginning of this project, and the integration of the needfinding activity was a crucial part. Needfinding is a process that focuses on understanding system requirements and potential user needs to ensure that the end product successfully satisfies them.

Following preliminary analysis, it was decided to engage directly with end users, primarily students, to gain a deeper understanding of their needs related to the thesis portal. By interviewing students, it was possible to gather concrete feedback, essential for improving the user experience during the migration to the new system.

3.1 Interview methodology

To collect relevant data, efforts were made to interview a diverse group of students from various academic backgrounds. It was crucial to select students who had recently completed their degree or were near graduation, as they were best positioned to provide meaningful feedback on the thesis management process.

The interviews were conducted mainly via Google Meet video calls, offering flexibility and convenience in scheduling. Each session was recorded, with prior consent from the interviewees, and transcribed for detailed qualitative analysis. This approach ensured the collection of valuable qualitative data, enabling a better understanding of the challenges and expectations regarding the thesis portal.

The interviews lasted an average of 30 minutes. We chose to ask relatively generic questions to allow the participants the freedom to express themselves. Based on their responses, we determined whether follow-up questions were necessary to have further clarifications.

Before each interview, the participants signed an informed consent form, outlining the purpose of the interview and the usage of the collected data, in full compliance with privacy regulations.

The following questions were selected for the interviews:

1. How did you find the thesis most suited to your interests?
    - Follow up: If you consulted the proposals on the portal, do you think the 'Thesis' section of the portal is structured appropriately?
2. What are the most difficult aspects of managing the thesis process?
    - Follow up: Are there specific points in the process where you feel or have felt stuck or frustrated?
3. How important is it for you to have a clear overview of the progress and deadlines of your thesis?

- ❍ Follow up: What format or visualization would help you the most in keeping track of your progress?

4. Have you ever had difficulty meeting your thesis deadlines? If yes, why?
   - ❍ Follow up: What type of support or tool could have helped you avoid this issue?
5. How do you handle your supervisor's revisions and feedback?
   - ❍ Follow up: Is there a tool or process that you think could improve this interaction?
6. Have you ever had difficulty finding specific information in the 'Thesis' section of the portal?
   - ❍ Follow up: What information did you find most difficult to locate?
7. How do you manage documents such as drafts, final versions, and references?
   - ❍ Follow up: What would make this process more efficient for you?
8. How do you organize meetings with your supervisor?
   - ❍ Follow up: Is there any feature or tool that could improve this process?
9. How useful do you find the current content in the 'Thesis' section of the portal?
   - ❍ Follow up: Are there resources or information that you think are missing?
10. What features or functionalities do you consider most valuable in a thesis management system?
    - ❍ Follow up: Why do you think these features are important?
11. Is there any functionality you would like to add to the 'Thesis' section of the portal?
    - ❍ Follow up: How do you think this feature would enhance the user experience?

3.2 Sample analyzed

The sample of interviewees consisted of eight students from different engineering courses at Politecnico di Torino. The selection was made to capture diverse perspectives on the thesis management process; in particular, an attempt was made to interview students with different backgrounds and types of theses. All participants were either in the final stages of their degree program or had recently completed their thesis, providing relevant and fresh insights into the thesis management experience. The graphs below show the variety of interviewed students according to their course of study, gender, and type of thesis chosen.

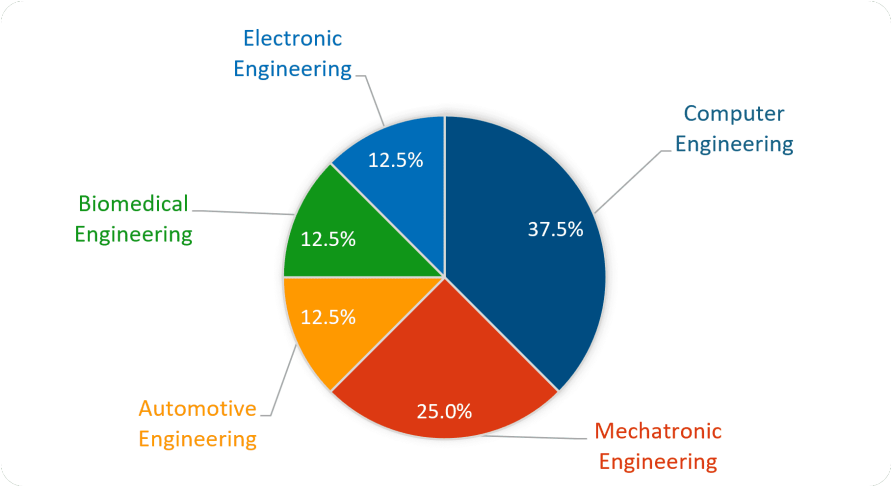


Figure 3.2: Course of study breakdown

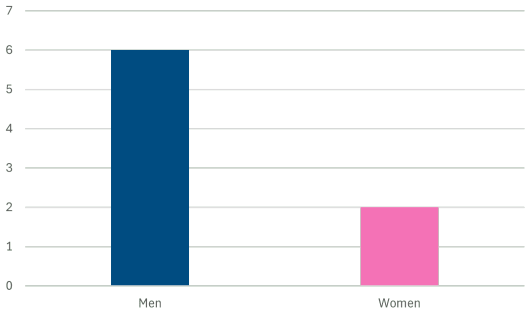


Figure 3.3: Gender breakdown

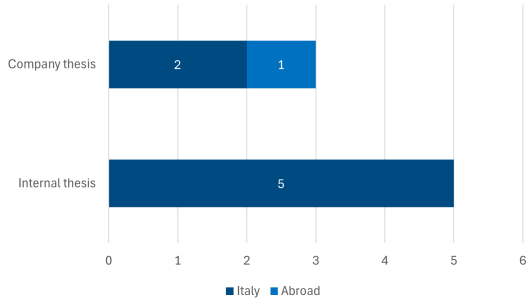


Figure 3.4: Thesis type breakdown

3.3 Interview summary

The current section shows the summarized responses of the interviewed students. Additionally, after all the answers to each question, there is a brief summary of what emerged with some commentary or reflection.

1. How did you find the thesis most suited to your interests?

- **Student 1**: *I directly contacted the professor I wanted as my supervisor and chose the proposal that was closest to my interests. The proposal was neither available on the Teaching Portal nor presented during lectures.*
- **Student 2**: *Through the presentations of thesis proposals managed by the professors of my degree program.*
- **Student 3**: *Since I wanted to do a thesis in a company, I initially started looking at the various available proposals in the thesis section. However, I found it difficult to search because there is no filter or search function, and many listings were actually for theses that had already been assigned. So, I moved to the internship section, looking first at the companies and then thinking of a possible thesis topic.*
- **Student 4**:*To find the thesis, I wrote to several professors of the courses I found most interesting, asking them if they had any theses available, and I was able to choose from the proposals offered.*
- **Student 5**: *I found it through the portal.*
- **Student 6**: *By speaking with various professors whose research areas aligned with my interests. Each of them made one or more thesis proposals, and I chose the one I felt was most in line with my capabilities.*
- **Student 7**: *I found it on the portal. Initially, I simply went to the 'Thesis' section where all the proposals are listed, and I started scrolling through them. However, there were too many, so I began searching with specific keywords. I think I searched for the keyword 'robotics' or something related to robots, and this way, if I remember correctly, about ten proposals came up. From there, I read them and emailed the professors to get more information. Some professors didn't respond or reply after 8 weeks, so I focused on the thesis I was most interested in and for which I received a response.*
- **Student 8**: *I found the thesis on the portal, where all the proposals are listed. I was specifically looking for a thesis abroad, in a company, so I was quite*

focused in my search because I knew that my supervisor had contacts with a particular company, and he was the link that helped me secure the position. My path is a bit unusual because I already had a thesis. I had asked a professor from one of my courses if he had any available theses, and he said yes. We had started working on it, but I wasn't convinced and wanted to do something more. So, I started looking for other opportunities on the platform. In fact, I cancelled the previous thesis to look for a new one. The process was lengthy because I had to go to Switzerland, and the bureaucracy there was slow: it took months of paperwork. I had to go through interviews, but honestly, apart from that, it wasn't particularly difficult. To cancel my previous thesis, I sent a ticket to the administration, and they quickly deleted my thesis, allowing me to submit a new request. One thing I want to mention is that since I did my thesis in a company, I had to split my credits (18+12), and this process took three weeks, almost a month, during which I couldn't move forward with the request.

Follow up: If you consulted the proposals on the portal, do you think the 'Thesis' section of the portal is structured appropriately?

- **Student 1**: *I only consulted the thesis proposals on the portal at first, but I didn't find anything particularly interesting. It's challenging to search more specifically for something that fits your interests. To view the proposals from the professors I was interested in, I had to scroll through the entire list of proposals using CTRL+F to see only those from the professors I was interested in.*
- **Student 2**: *The 'Thesis' section of the portal is absolutely not well-structured: there is no clear division between the various theses (recommended degree program, professor's department, etc.), and often the theses descriptions are poorly formatted, with text overlapping. Additionally, there is no way to filter the proposals.*
- **Student 3**: *No, theses should be better categorized, and there should be a filtering function.*
- **Student 4**: *No, I didn't consult it much as I heard several negative opinions about it (for example, that there are often proposals that are already assigned or expired).*
- **Student 5**: *Yes, I think the way it is structured is fine.*
- **Student 6**: *I consulted it very quickly, but I preferred to reach out to professors with whom I had already interacted during my university journey, and none*

of the proposals on the portal came from them. Additionally, none of the proposals particularly impressed me.

- **Student 7**: *I think the section dedicated to proposals is confusing because there are so many proposals, and only a small portion is truly relevant to one's field of study.*
- **Student 8**: *Yes, I did consult them, and since there wasn't proper filtering, I ended up scrolling through and reading most of the proposals. However, I realized that some were not related to my field of study, some were no longer available, or the professor didn't respond to clarification requests regarding their proposal.*

Summary

Three ways were mainly used by students interviewed to find their thesis:

- 4 students contacted professors (students 1, 2, 4, 6)
- 3 students used the 'Thesis' section (students 5, 7, 8)
- 1 student used 'Internship' section (student 3)

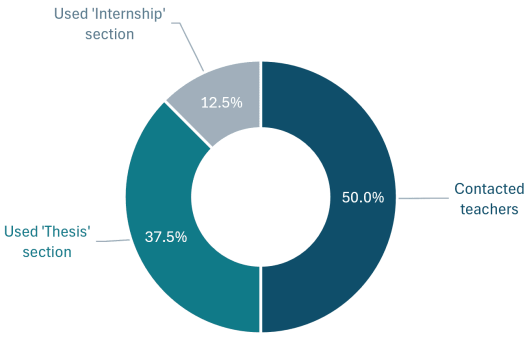


Figure 3.5: Methods used by students to find their thesis

Moreover, seven students out of eight expressed dissatisfaction with the current 'Thesis' section structure, citing difficulties in searching, lack of filtering, confusing organization, and irrelevant or expired proposals. Only one student (Student 5) was satisfied with the current structure.

One issue to add is the fact that one student reported that the description of theses often comes out poorly formatted and overlaid with other text. This issue is due

to the use of PoliTools, an extension for Google Chrome, which the student said he uses primarily to enable a dark mode on the portal. Although this extension offers the ability to have a dark interface, it compromises the page content layout, as shown in the following image:

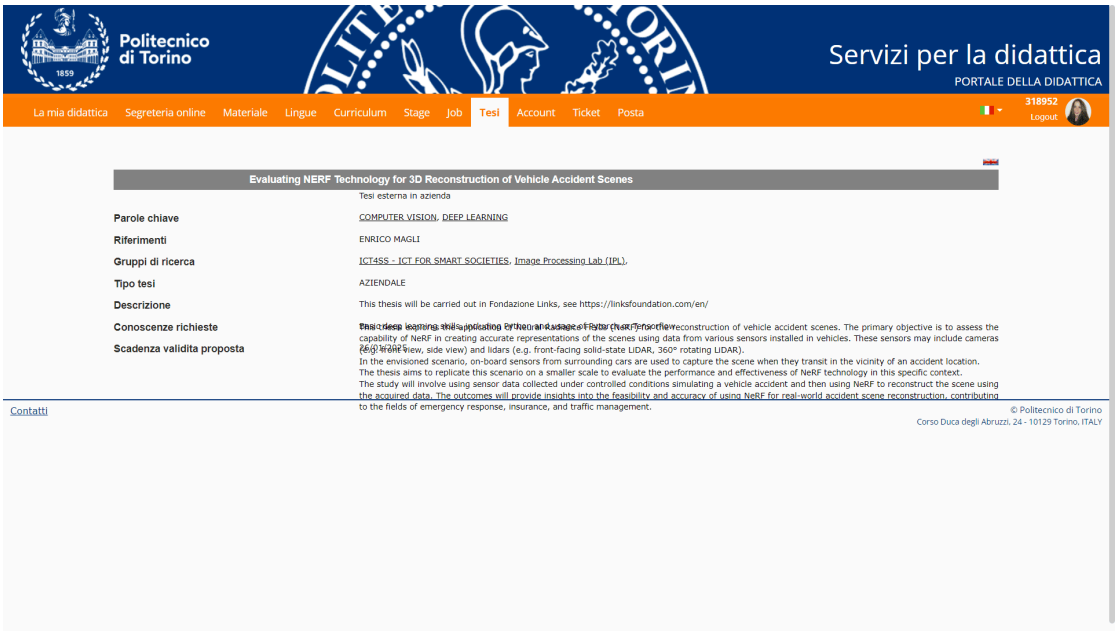


Figure 3.6: Thesis page with PoliTools extension

This suggests that there is a demand among users for a dark-themed interface, motivated by some students' preference for a mode more suitable for prolonged reading in dimly lit environments. Instead of relying on third-party tools, which can cause technical problems, it might be worth considering the introduction of an official dark palette in the portal. This feature would improve the user experience by reducing eye fatigue and maintaining proper content layout.

---

2. What are the most difficult aspects of managing the thesis process?

- **Student 1**: *It's not immediately clear what you need to do; you have to search through many pages to figure out how to proceed or contact the supervisor, who isn't always easily reachable. Additionally, I'm aware of issues regarding discrepancies between the thesis title proposed by the professor and the version published on the portal.*

- **Student 2**: *From my point of view, the most challenging part was the lack of a defined timeline, especially regarding when to choose the thesis (as I didn't go to a company).*
- **Student 3**: *I didn't face significant issues in managing the thesis process, except for the initial search.*
- ***Student 4**: As of now, I haven't encountered any difficulties (I hope not to have any during the submission process; for now, I have only made the request specifying the title of the thesis).*
- **Student 5**: *In my case, the difficult part was not having a clear idea from the beginning of what to do to develop the thesis and what results were expected. Especially the results, because knowing which graphs/tables were needed would have helped me understand the steps to take and which data to save.*
- **Student 6**: *Managing timelines: understanding when to start looking around, when to begin working on it.*
- **Student 7**: *I didn't encounter major issues with the process; however, I think it would be helpful to know, before starting the work, when the professor expects it to be completed, as this could avoid possible conflicts.*
- **Student 8**: *Some things aren't very clear; they aren't explained that well, so I had to search for information or ask in various Telegram groups. For example, regarding the thesis in a company, it didn't affect me much because I had started in January, but I know students who were frustrated because they couldn't start their thesis due to issues with splitting the credits. Aside from that, it was a bit annoying when I had to submit the thesis conclusion request. Since my thesis was done in a company, it wasn't clear to me that I had to fill out a daily log of my internship activities. Without that, the internship couldn't be registered, and it showed that I had an exam pending, even though I had finished my exams much earlier. In my opinion, having more guidelines on the thesis and the necessary documents would be helpful, especially regarding the summary for the thesis committee, because while it says that only some degree programs require it, it doesn't specify which ones. Additionally, I panicked a bit when I couldn't add my external co-supervisor during the thesis conclusion process. I had to ask my supervisor to add him because he wasn't registered yet.*

Follow up: Are there specific points in the process where you feel or have felt stuck or frustrated?

- **Student 1**: *I often had difficulty contacting my supervisor.*

- **Student 2**: *No.*
- **Student 3**: *The beginning, as mentioned.*
- **Student 4**: *No.*
- **Student 5**: *No.*
- **Student 6**: *No, I was afraid I wouldn't have enough time, so I started moving very early and therefore had time to clarify any doubts that arose along the way.*
- **Student 7**: *No.*
- **Student 8**: *Apart from the situations described above, no.*

Summary

One problem that emerged from the answers to the second question was the **lack of clarity and guidance** on the stages of the process. Students 1, 5, 7, and 8 expressed frustration with unclear directions and uncertainty about the different tasks they needed to take. In particular, student 8 highlighted the need for clearer guidelines regarding the summary of the paper, the need for the activity daily log in case of company thesis, and the procedure for including a person from outside the university as a co-supervisor.

Another critical aspect mentioned by students 2 and 6 is **time management**. They expressed a desire to have a defined timetable that would help them understand when to start their thesis work, which is especially important for those who are not tied to a company.

On the other hand, students 3, 4, and 7 reported that they had not encountered significant difficulties so far. Student 3 acknowledged some initial uncertainty but overall felt that the process was manageable.

In summary, while many students reported that they struggled with unclear information and time management, others were able to manage the process with relatively few problems. This feedback suggests the need for improved communication and resources to guide students through the complexities of their thesis process.

---

3. How important is it for you to have a clear overview of the progress and deadlines of your thesis?

- **Student 1**: *It's very important. I found myself having to consult many pages to figure out the deadlines. Additionally, I had to discuss with the professor because there's a difference between submitting the thesis in its final version and having a deadline while still being able to make modifications (which is not very clear).*
- **Student 2**: *Personally, since I developed an experimental thesis with a professor from Politecnico, it was easy to track the progress through a weekly meeting. As for deadlines, the portal does a good job of making the most important dates easily accessible.*
- **Student 3**: *It's fairly important; it would be useful to have reminders of various deadlines without having to look them up in the thematic calendar.*
- **Student 4**: *Very important; currently, the only way I found to obtain this information is from the website, not directly from the portal (I could be wrong; I haven't actually searched for it). The ability to see the deadlines for the upcoming graduation sessions directly from the teaching portal would definitely be the most convenient way.*
- **Student 5**: *Yes, it's definitely important to have the deadlines clearly in mind, but I didn't have any issues with the method the University already uses.*
- **Student 6**: *It is a fundamental aspect, but accessing this information from the portal is not very intuitive.*
- **Student 7**: *Very important. I'm not sure if I received reminders, but they definitely would have been useful. I had checked the deadlines about a month and a half earlier and then got confused as time passed. Luckily, I submitted my thesis conclusion confirmation request on the deadline day without realizing it was the last day. After that, I downloaded the PDF from the Telegram group for my degree program to avoid missing future deadlines. Searching for the deadlines in the student guide is quite inconvenient.*
- **Student 8**: *It's very important, but I must say that the thesis deadlines are quite clear. It was always clear to me that everything had to be done by the 17th of October, seven days before the start of the thesis discussions, regardless of the actual defence date.*

Follow up: What format or visualization would help you the most in keeping track of your progress?

- **Student 1**: *A minimalist style is fine, as long as it's well organized. The first visualization format that comes to mind is a simple horizontal timeline*

that identifies where I am in the process and what I still need to do, with the corresponding deadlines.

- **Student 2**: *I don't know.*
- **Student 3**: *I think a table in the thesis section with the various deadlines would be useful.*
- **Student 4**: *I don't know, even just a side section with a deadline reminder for the nearest graduation session would suffice.*
- **Student 5**: *It could be useful to send the deadlines via email along with the email that arrives when the supervisor accepts the thesis proposal.*
- **Student 6**: *Probably setting deadlines in my agenda where the classes are and receiving notifications.*
- **Student 7**: *I'm not sure. For me, it would be enough to see the deadlines for the closest graduation session on the Thesis page.*
- **Student 8**: *I'm not sure; I didn't have any issues with the current deadline format.*

Summary

The majority of students interviewed felt it was important to have a clear view of progress and deadlines related to their thesis. In addition, many stated that it would be very convenient to be able to view deadlines directly from their page without having to go to the student guide each time to consult them.

Regarding the best display format to be able to track the progress of work, opinions varied: some said they preferred a tabular form, some a horizontal timeline, and some a side section or reminders.

What emerges is the need for a quick and intuitive way to consult the many deadlines and perhaps some indication as to the differences between the various deliverables, since, following some deadlines, it is possible to further edit some fields as opposed to others.

---

4. Have you ever had difficulty meeting your thesis deadlines? If yes, why?

- **Student 1**: *Not exactly, but I found myself having to do a "marathon" at the last minute because I learned about the exact deadlines late.*
- **Student 2**: *In my specific case, I didn't encounter any particular difficulties, since all the deadlines related to the project development were quite flexible.*
- **Student 3**: *No.*
- **Student 4**: *Not so far, but we'll see.*
- **Student 5**: *No, I didn't have any difficulties. I used my phone reminders to help me.*
- **Student 6**: *No.*
- **Student 7**: *Yes, see previous answer.*
- **Student 8**: *No.*

Follow up: What type of support or tool could have helped you avoid this issue?

- **Student 1**: *As mentioned earlier, having a clear visualization of my deadlines on the thesis page would have helped avoid the last-minute rush.*
- **Student 2**: -
- **Student 3**: -
- **Student 4**: -
- **Student 5**: -
- **Student 6**: -
- **Student 7**: *See previous answers.*
- **Student 8**: -

Summary

Most respondents had no difficulty meeting deadlines, but students 1 and 7 reported finding out some key dates late, thus having to speed up some steps of the work. This was because, having consulted the calendar in advance, they could not remember exactly the days of the deadlines.

To avoid such situations, it would be useful to offer students a more immediate and clear view of all deadlines, ideally integrated into the thesis page within the teaching portal. Such a feature, which makes important dates easily accessible without having to look elsewhere, could significantly improve the management of the thesis process and reduce the risk of errors or delays.

---

5. How do you handle your supervisor's revisions and feedback?

- **Student 1**: *At first, I had to upload my versions to the shared drive, and the professor would upload the version with comments, but this only happened once. In this case, the comments were inserted as LaTeX comments, making it cumbersome to compare them with my document to see what needed to be changed. Later, I simply uploaded the version, the professor would review it, then I'd go to office hours, and the supervisor would orally explain what needed to be revised and what was fine. There were also instances when the professor sent feedback via email.*
- **Student 2**: *Weekly, along with the other thesis students, we met with the professor to present the work done, and we proceeded to modify, integrate, or add the requested changes and comments to the work.*
- **Student 3**: *My thesis project is on Overleaf and shared with my supervisor. Therefore, I receive feedback either via email or during meetings.*
- **Student 4**: *I have weekly appointments via call for quick feedback, which have lasted both during the thesis work period and the writing period. Communications mainly occur through Microsoft Teams, sometimes via email. I found this arrangement very effective, but I believe it varies from supervisor to supervisor.*
- **Student 5**: *We had a meeting once a week, and as soon as it was over, I would immediately write down on my phone what was discussed. I also used the Trello app, which allows you to create a to-do list and check off completed tasks.*
- **Student 6**: *My supervisor was extremely available and proactive, so we always met in person for revisions.*
- **Student 7**: *In my case, I managed the thesis quite independently. The advisor gave me the project, and I started working on it. Every week or two, we had a meeting where my work was reviewed, and the professor either gave me new*

tasks or told me if anything needed modifying. I always shared the thesis with the professor via email.

- **Student 8**: *In my case, my supervisor wasn't very knowledgeable about the topics I dealt with in the company—he was more of a link. So, the content was reviewed and corrected by my company supervisor, especially since I had signed an NDA and couldn't disclose certain information. I had to normalize the graphs and be careful with some data. Therefore, everything had to be approved by the company. During my months at the company, I prepared a presentation every month or two, depending on the period, and we had calls with the supervisor to review the progress. For the rest, we communicated via email.*

Follow up: Is there a tool or process that you think could improve this interaction?

- **Student 1**: *It would be very helpful to have all comments related to thesis revisions in one section, perhaps within the shared drive itself, instead of having to search through email threads with the supervisor or compare the commented file with the original.*
- **Student 2**: *Having some sort of shared board between the supervisor and students with notes related to the latest revision could help keep track of comments and future work.*
- **Student 3**: *Maybe a direct messaging system within the thesis section.*
- **Student 4**: -
- **Student 5**: *I don't know of a tool that could help with this process, but I think it's very important because I noticed that we often weren't aligned with the PhD student. Sometimes, he didn't understand what we were saying, and other times we didn't understand him. Sometimes the PhD student didn't align with the professor's ideas either. This led to redoing the same task in different ways multiple times. So, it would be useful to have an app where the student, PhD student, and professor can all access and input brief descriptions of what is being done or planned, with the ability for others to comment and express their opinions. This would allow for quicker feedback without waiting weeks or months, with less risk of having to redo completed work, and it would help both the professor and PhD student to be aware of the methods and algorithms the student is using.*
- **Student 6**: I *think it's a subjective aspect, so perhaps deadlines should also be set for supervisors; I think of classmates who had to delay their theses due*

to a lack of responses from their respective supervisors.

- **Student 7**: I'm not sure; I was comfortable with this method.
- **Student 8**: I don't know; I didn't encounter significant issues with this process.

Summary

Most students handled the supervisor's feedback through regular meetings, both online and in-person, using different tools for sharing the paper, such as Overleaf, Google Drive, the Polytechnic's shared disk, or e-mail. However, there is no standard procedure for managing and sharing materials, leaving each faculty member to choose independently the method he or she prefers.

From this perspective, it would be appropriate to evaluate whether it may be useful to implement a dedicated space for the sharing of material between students and teachers in the thesis portal. A central space could facilitate the sharing of material, including using external software contracted with the university, making the process more uniform. The question remains whether this solution would be necessary or whether the current tools are already sufficient.

Suggestions for improving the process included the idea of a section for comments on reviews, a shared whiteboard to track feedback, direct messaging in the thesis portal, and an app to synchronize feedback between students, teachers, and collaborators. Another proposal concerns the setting of deadlines for supervisors, to ensure more timely responses.

---

6. Have you ever had difficulty finding specific information in the 'Thesis' section of the portal?

- **Student 1**: *Yes.*
- **Student 2**: *Yes.*
- **Student 3**: *Yes.*
- **Student 4**: *I haven't searched for anything regarding this yet, so there are no issues to report.*
- **Student 5**: *No, it seems well organized.*
- **Student 6**: *Yes.*

- **Student 7**: *No, except for the deadlines as mentioned earlier.*
- **Student 8**: *Yes.*

What information did you find most difficult to locate?

- **Student 1**: *For the thesis proposals uploaded by professors, it would have been helpful to have filters and to only see active theses in my field, not those from other degree programs (or at least to filter them quickly). Additionally, I would have liked a clear distinction between different types of theses (experimental, company-based, etc.) so I could filter them according to my preferences. Besides this, nothing else.*
- **Student 2**: *Information on how to handle enrolment for the new academic year if you only have to submit the thesis.*
- **Student 3**: *I found it difficult searching for specific theses, for example, by topic (e.g., Cybersecurity) or for specific supervisors.*
- **Student 4**: -
- **Student 5**: -
- **Student 6**: *The deadlines.*
- **Student 7**: -
- **Student 8**: *Information about the thesis in a company, the credit splitting process, adding external supervisors, and the documents needed to register the internship.*

Summary

Many students reported difficulties in finding specific information in the 'Thesis' section of the portal. Regarding the specific material they were unable to locate, Student 1 emphasized the need for filters for thesis proposals and clearer distinctions between different types of theses. Student 2 found it difficult to locate information about the registration procedures for the new academic year when there is only the thesis in the annual personal study plan. Student 3 struggled in searching for theses by topic or teacher, while student 6 had difficulty in finding deadlines. Finally, student 8 highlighted the challenges related to company theses, such as the subdivision of credits, lack of information on the necessary documents, and the addition of external co-supervisors.

7. How do you manage documents such as drafts, final versions, and references?

- **Student 1**: *I uploaded all the different versions to the shared drive, receiving feedback from my supervisor in the ways described earlier. However, I had more difficulty with the process of uploading the thesis in its final version because I didn't know the difference between PDF and PDF/A. Since I wrote the thesis in LaTeX, the version was already in PDF/A, but I had no way of knowing that, so the conversion was unnecessary.*
- **Student 2**: *Since I haven't started writing the actual thesis yet, I can't say for sure, but most likely a series of folders on a drive, perhaps shared with the supervisor to get real-time feedback.*
- **Student 3**: *I have my thesis project on Overleaf.*
- **Student 4**: *Everything will be managed on Overleaf, but I haven't started the actual writing yet, so I can't say.*
- **Student 5**: *I used Google Drive because I did a joint thesis, and this allowed us to see each other's changes in real-time, as well as create different folders to organize all the documents properly.*
- **Student 6**: *I save everything both on my computer and a USB drive, as well as in the cloud.*
- **Student 7**: *I wrote my thesis on Overleaf and managed the versions independently. Usually, at the end of each writing session, I downloaded the PDF to have a sort of backup of the various versions.*
- **Student 8**: *I handled everything pretty autonomously. I wrote my thesis in Word due to the high number of graphs and tables. The company reviewed the content, and my academic supervisor reviewed the structure.*

Follow up: What would make this process more efficient for you?

- **Student 1**: *Uploading the file for the final thesis submission requires it to be in PDF/A format. It would be easier for everyone if these details were handled internally by the Politecnico's system, with any necessary conversions done automatically, instead of the user having to do it, especially when they might not know the difference between PDF, PDF/A, etc. (The student sees that it has to be PDF, uploads it, and that's it.)*
- **Student 2**: -

- **Student 3**: *Personally, I don't think any major improvements are needed in this area.*
- **Student 4**: -
- **Student 5**: -
- **Student 6**: *I think everyone can manage it independently.*
- **Student 7**: *I'm not sure; I was happy with my method.*
- **Student 8**: -

Summary

Students interviewed said they manage their documents, including drafts, final versions, and references, using different platforms and strategies. Some make use of a shared drive to keep track of various versions, while others use Overleaf for their thesis projects. Some students prefer Google Drive for collaborative work, which allows them to track changes in real-time, while others save documents to multiple devices, including USB sticks and the cloud. While some write the thesis in Word and handle it on their own, others manage versions by downloading PDFs for backup after every session.

In terms of efficiency, it was suggested that the system could automate the conversion to PDF/A for final thesis submission to reduce confusion. Nonetheless, the majority of students showed a high degree of comfort with their document management procedures since they were happy with the current approaches and did not suggest any notable changes.

---

8. How do you organize meetings with your supervisor?

- **Student 1**: *I kept sending emails until I got a response to schedule the meeting.*
- **Student 2**: *Through a Telegram group where the meetings are organized.*
- **Student 3**: *Via email.*
- **Student 4**: *We have set fixed weekly appointments to maintain constant and quick synchronization. I found Microsoft Teams very effective, but it is an external tool that needs to be agreed upon with the supervisor.*

- **Student 5**: *Since he was abroad at the time, we had a video call once a week through Google Meet. In each meeting, we discussed any doubts we had or he presented new ideas.*
- **Student 6**: *We communicated via email for quicker information exchanges and in person for revisions and organizing work.*
- **Student 7**: *Via email.*
- **Student 8**: *We mostly communicated via email. We didn't have many meetings because my supervisor mainly handled the "structure" of the document, as he wasn't familiar with the thesis topic.*

Follow up: Is there any feature or tool that could improve this process?

- **Student 1**: *Ideally, there would be a panel showing the supervisor's available time slots where I could book a meeting slot.*
- **Student 2**: *I'm not sure, I'm currently satisfied with this method.*
- **Student 3**: *A messaging function or a booking service.*
- **Student 4**: -
- **Student 5**: -
- **Student 6**: Refer to previous answers.
- **Student 7**: *My advisor has always been very responsive to emails, so I don't see this as an area that needs improvement.*
- **Student 8**: *I'm not sure; I didn't have problems with this process.*

Summary

According to the interviewees, they mostly used email to arrange meetings with their supervisors, but some also used Telegram groups. Some set weekly appointments to ensure constant communication, while others made video calls via Google Meet, especially if the supervisor was abroad. One student mentioned using Microsoft Teams, although this requires prior agreement with the supervisor.

One noteworthy aspect is that each supervisor uses their own method for scheduling meetings and, as a consequence, there is no standard method.

Some students suggest improving the process by implementing a panel showing the supervisor's available times to book meetings, or a messaging function, or a

booking service. However, many students say they are satisfied with the current methods and do not need significant changes.

---

9. How useful do you find the current content in the 'Thesis' section of the portal?

- **Student 1**: *It's useful because it's the only tool available to manage your thesis.*
- **Student 2**: *Aside from being able to see theses offers, I don't find any particularly useful information.*
- **Student 3**: *Not very useful.*
- **Student 4**: *So far, I have only seen the thesis request part; therefore, I cannot say what other content it contains.*
- **Student 5**: *Currently, the 'Thesis' section of the portal seems to be useful only for thesis application, but could be improved by adding more detailed organizational information.*
- **Student 6**: *There is currently no information in the thesis section, so it is a completely useless section.*
- **Student 7**: *The content is definitely useful, but the thesis search function is a bit confusing. It would probably be better if it were more categorized. Currently, when you open the Thesis page, 2000 proposals appear, of which 1700 are likely not relevant to you.*
- **Student 8**: *It's useful but a bit confusing. For example, when you enter the 'Thesis' tab, you have the option to access the Thesis page again, which contains the same information as before, plus the shared drive.*

Follow up: Are there resources or information that you think are missing?

- **Student 1**: *It would be useful, for example, to have various resources for writing the thesis (Politecnico logo, templates, etc.) directly in the Thesis section without having to search elsewhere.*
- **Student 2**: *A calendar with all the important deadlines easily identifiable, and perhaps some sort of guide on how to write the thesis (templates, etc).*
- **Student 3**: Refer to previous answers.

- **Student 4**: -
- **Student 5**: *It could be useful to add general organizational info, such as when the official emails for the thesis discussion and graduation will be sent, how many minutes we have to present the thesis, and perhaps deadlines could also be listed here.*
- ***Student 6**: Various deadlines to be respected and proposals could be directly included in the "thesis" section.*
- **Student 7**: *One thing I find missing is an indication of the time frame to complete the thesis. I understand that with research theses, this is often unknown, but companies usually give a deadline of around 6 months, while at Politecnico, there's no such indication. I think it would be helpful to know when the professor expects the thesis to be finished. For example, after about a month of work, my professor mentioned that he would like the work to be completed by the October session. It worked out well since that was roughly my goal too, but otherwise, it could have been an issue.*
- **Student 8**: *Maybe a template for writing the thesis. I had to search for it and found it in a remote corner of the Politecnico website.*

Summary

Regarding the value of the present content in the portal's "Thesis" section, the respondents' opinions were divided. While some students thought it was helpful as their main thesis management tool, others said it was unclear or lacked useful information. Specific feedback highlighted that the thesis search function could be more organized, as many irrelevant proposals appeared at once.

These are some of the suggested improvements:

- Adding resources for writing the thesis, such as templates and the Politecnico logo.
- Providing a calendar with important deadlines and a guide on how to write the thesis.
- Including general organizational information, such as when official emails regarding thesis discussions and graduations would be sent, as well as presentation time limits.
- Indicating expected time frames for thesis completion to help students manage their schedules more effectively.

Overall, there is a strong call for improved categorization and clarity in the content presented in the 'Thesis' section.

---

10. What features or functionalities do you consider most valuable in a thesis management system?

- **Student 1**: *I'd like to have a well-structured page where I don't have to struggle to find information.*
- **Student 2**: *The ability to view the entire thesis process (various versions, feedback, and deadlines).*
- **Student 3**: *Clear search and an updated list, with deadlines clearly stated in the section itself.*
- **Student 4**: *Similar to the previous answers.*
- **Student 5**: *A system that clearly highlights the thesis objective and the expected results. And for each result, it should show the steps needed to achieve it. This way, it provides a clear overview and allows for better work organization.*
- **Student 6**: *Defining the bureaucratic procedures to follow and the timelines.*
- **Student 7**: *One of the most valuable features, in my opinion, would be having a clear and updated overview of deadlines related to the thesis, specifically for the closest graduation session. This would help avoid confusion and delays. Additionally, a more accurate and categorized search function for thesis proposals would be useful to reduce the number of irrelevant proposals that appear and make it easier to choose. Lastly, having a general indication of when the advisor expects the work to be completed would prevent any conflicts or misunderstandings.*
- **Student 8**: *I think clear instructions and well-structured information are crucial, especially regarding the thesis conclusion request, the necessary documents, and the internship process. For example, it wasn't clear that I had to fill out a daily log of my internship activities, and that caused problems because it showed I still had a pending exam, even though I had completed my exams much earlier. Clearer guidelines on what the thesis committee requires, such as the thesis summary, would also be very helpful because right now it depends on the degree program or the supervisor, and there aren't any uniform guidelines.*

Follow up: Why do you think these features are important?

- **Student 1**: *To improve the user experience for students.*
- **Student 2**: *Because they give us students the opportunity to have a single point of reference for managing our thesis.*
- **Student 3**: *Without these features, you end up with a very long list of potential theses that can be overwhelming, and since it's not kept up to date, it's not reliable, which discourages students from using it.*
- **Student 4**: -
- **Student 5**: -
- **Student 6**: *Because if deadlines are not met or procedures are not followed correctly, there is a risk of compromising or at least delaying graduation.*
- **Student 7**: *Because they allow for a more organized and informed approach to managing the thesis. A better search function would reduce the time wasted on irrelevant proposals, while a clear overview of deadlines and the advisor's expectations would help students stay on track and plan their work better.*
- **Student 8**: *Because they help avoid confusion and potential delays. For example, in my case, I had to ask in Telegram groups or reach out to the coordinator because it wasn't clear how to fill out some documents or whether the thesis summary was necessary. Having everything clearly explained on the platform would have made things much easier, especially with tight deadlines.*

Summary

As regards the characteristics that students consider most important in a thesis management system, several key themes have emerged.

Firstly, students expressed a strong desire for a well-structured platform that would allow them to easily find and navigate information related to the thesis. Effective management requires visualizing the entire thesis process, from drafts to feedback and deadlines, which has been highlighted as essential. A clear and up-to-date list of deadlines was found useful by many students to significantly reduce confusion and avoid having to search for information in different places.

In addition, students stressed the need for a more accurate and categorized research function for thesis proposals. They were often overwhelmed by a long list of proposals, even those not relevant to their degree course, making it difficult to identify those most relevant to their interests.

It was also considered essential to have clear instructions on bureaucratic procedures, in particular regarding the thesis conclusion confirmation request and internship requirements. This clarity would help students understand what is expected of them and avoid unnecessary complications.

During the conversation about the importance of these features, students pointed out that they would greatly improve the overall user experience. A well-organized system would provide a single point of reference for the management of the thesis, facilitating monitoring of progress and deadlines. Clear guidelines and updates would also avoid feelings of oppression and uncertainty, allowing students to focus on research and writing.

---

11. Is there any functionality you would like to add to the 'Thesis' section of the portal?

- **Student 1**: *Having all the resources needed for writing a thesis (Politecnico logo, various templates, etc.) directly on the student's thesis page would be very useful.*
- **Student 2**: *The ones mentioned in the previous answers (calendar with deadlines, resources for writing the thesis, etc.).*
- **Student 3**: Refer to previous answers.
- **Student 4**: Refer to previous answers.
- **Student 5**: *It would be useful to have feedback from peers who have already done a thesis with a certain supervisor, to know how they found working with them.*
- **Student 6**: See answer 9.
- **Student 7**: *No, rather than adding features, I would focus on better structuring the existing functionalities.*
- **Student 8**: *I would definitely add templates, like one for the cover page, or more detailed guides on how to write the thesis. The information is there, but you have to search for it in remote areas of the site, sometimes through Google. I think it would be helpful to have everything centralized in a more accessible section. Additionally, the part of the portal dedicated to uploading the thesis is a bit confusing — you have to click through several sections before you find where to upload the final version in PDF/A format, and I think this could be simplified.*

How do you think this feature would enhance the user experience?

- **Student 1**: *It would allow people not to have to go crazy looking for information.*
- **Student 2**: *It would make some very useful information more accessible.*
- **Student 3**: Refer to previous answers.
- **Student 4**: -
- **Student 5**: *This helps understand if your personality fits in that research environment or if there's a risk of having a negative experience. This way, there are no surprises.*
- **Student 6**: See answer 9.
- **Student 7**: -
- **Student 8**: *It would make the experience much simpler and more straightforward. You wouldn't have to waste time searching for information or templates on different sites, and the thesis upload process would be smoother. Right now, if you're not used to navigating the portal, you risk getting confused or wasting time figuring out where to click.*

Summary

In discussing additional functionalities for the 'Thesis' section of the portal, students stressed the importance of centralizing resources and rearranging existing functionalities to improve their overall experience.

Many students suggested incorporating the essential materials for writing their thesis, such as templates, guides, and the university logo, directly into their thesis pages. Such accessibility would reduce the time and effort required to identify the information needed, often scattered across the university's website.

There is also a general agreement on the benefit that could have the introduction of deadlines for all steps to get to the degree directly within the section 'Thesis'.

Furthermore, students emphasized the need for a more advanced and categorized search functionality for thesis proposals, allowing them to more effectively filter through available topics and focus on those most relevant to their interests.

Some students consider that the addition of new functionalities is beneficial, while others have stressed the need for a better structure of existing functionalities. In

particular, there was also an observation on the navigation tree that currently includes several sections named in the same way ('Thesis') and that requires a few clicks before you can access important information or procedures to be able to submit your thesis or register for the final exam.

Overall, students feel that these improvements would simplify their interactions with the portal, making it more intuitive and efficient.

3.4 Conclusive evaluations

Through the interviews conducted, it was possible to extract the most important needs of the students along with the most recognized critical issues. The qualitative data collected were subsequently translated into user stories so that it was possible to specify what features needed to be implemented in the new thesis management system.

In fact, the user stories describe in a very pragmatic way how the user (in this case, the students) would interact with the system to be developed. Each user story contains the type of user, their need, and why they need it. In this way, through the needs made explicit by the users, it is possible to guide software development so as to provide functionality that directly satisfies the end users.

User stories follow the Role-Functionality-Benefit pattern, which is given below:

```
As a [actor type],
I want [to do something],
So that [some value is created].
```

In this way, it is possible to clearly express what the end user's expectations are and make explicit the value the system provides to them.

User stories must also comply with the **INVEST principle**, introduced by Bill Wake in a 2003 article [3]. This principle states that each user story should be:

1. **Independent**
2. **Negotiable**
3. **Valuable**

4. **Estimable**
5. **Small**
6. **Testable**

Another advantage of user stories is that they allow incremental development of features. In fact, it is a good practice to prioritize them by trying to develop the most critical ones first, i.e., those with the greatest impact or that meet the most urgent needs, and then the others. All user stories will be entered into the product backlog and, at each sprint, the development team can select those to be implemented based on the priorities established and the resources available. Agile development methods will be described later in chapter 6.1.

This is the list of the main extracted user stories ordered by the business value they provide:

1. As a student, I want to see all available thesis proposals so that I can review the full list of options.
2. As a student, I want to apply filters to the thesis proposals so that I can find relevant topics more efficiently.
3. As a student, I want to sort thesis proposals so that I can prioritize the most recent or soon-to-expire options.
4. As a student, I want to access degree-related deadlines in one place so that I can avoid missing key dates.
5. As a student, I want to see the full thesis process in one view so that I can track my progress easily.
6. As a student, I want to quickly access templates, guides, and the university logo from the 'Thesis' section to avoid searching through multiple sources.
7. As a student, I want to enable a dark mode in the portal so that I can reduce eye strain when working in low-light conditions.

Chapter 4

Prototyping

Prototyping marked the beginning of the design and development phase. While in the analysis phase, to have an integral vision of the process and interactions, the focus was on the three users who interact in the thesis management (students, teachers, and the secretariat), in this phase, it was decided to start focusing on a single interface, the students one, since the student teaching portal was the only one being redesigned and developed.

From this point on, the focus was on the implementation of a new interface for thesis management from the student's point of view; in particular, the main objective was to redesign the interface for consulting the thesis proposals.

There will be several references to the 'new' and the 'old' portal: the 'old' portal is the one analyzed so far while the 'new' one is the structure given as a basis for this second phase, i.e. a high-fidelity prototype made by some students previously who entirely redesigned the user experience of the student teaching portal [1].

4.1 Information Architecture analysis

To arrange the information correctly and intuitively, the information architecture of the 'old' portal was analyzed and documented. In particular, the general IA structure of the new teaching portal had already been reorganized in detail in the first navigation levels and, based on this rearrangement, an attempt was made to define the content organization for thesis management.

4.1.1 Information Architecture analysis (AS-IS)

The analysis of the old portal revealed several significant critical issues in the structure of the navigation tree of the portal. In particular, the following problems emerged:

- **Sections with ambiguous or duplicate names:** there are sections at different levels with the same name, such as 'Thesis'. This is not particularly user-friendly, as the user may struggle to determine which level of the navigation tree they are in.
- **Unintuitive positioning of information:** much information is placed in sections that do not meet the logical expectations of users, forcing them to browse for a long time before finding the content they are looking for. This problem negatively affects the overall user experience.
- **Presence of obsolete or unused sections:** several areas of the portal are completely unused or contain information that is now obsolete or even inaccurate (for example, the 'Contact' section or the 'Forum'). These sections were used in the past but now only contribute to the interface load, making it less functional.

4.1.2 Information Architecture TO-BE

To address the issues that emerged, a new arrangement of the information content concerning the management of the entire thesis process was drawn up. The decisions are as follows:

- **Removal of superfluous sections:** sections no longer needed were removed to simplify the interface.
- **Clear distinction between 'Thesis Proposal' and 'Thesis':** the decision was made to keep the 'Thesis Proposals' section separate from the section about the student's specific thesis. Within the 'Career' section, therefore, there will be two subsections: 'Thesis Proposals', which contains the list of filterable thesis proposals, and 'Graduation and Final Examination', which will contain the information on the eventual thesis assigned and the instructions for proceeding with the registration for the final examination.

The AS-IS and TO-BE information architecture scheme can be found at the following link: `https://www.figma.com/design/QdV54j1bA5NqikeInlTmQJ/In`

formation-Architecture?m=auto&t=SY4jJO57GeQ35JZC-1

4.2 Low-Fidelity prototyping

After defining the location of the information concerning the thesis proposals, two low-fidelity prototypes were produced to evaluate different options for presenting the information content. The two designed solutions differed in the way the data was displayed: one in tabular format and the other by card.

The low-fidelity prototypes, although not definitive, offered a clear representation of the possible solutions, facilitating the comparison of alternatives. The prototypes developed for the evaluation are shown below:

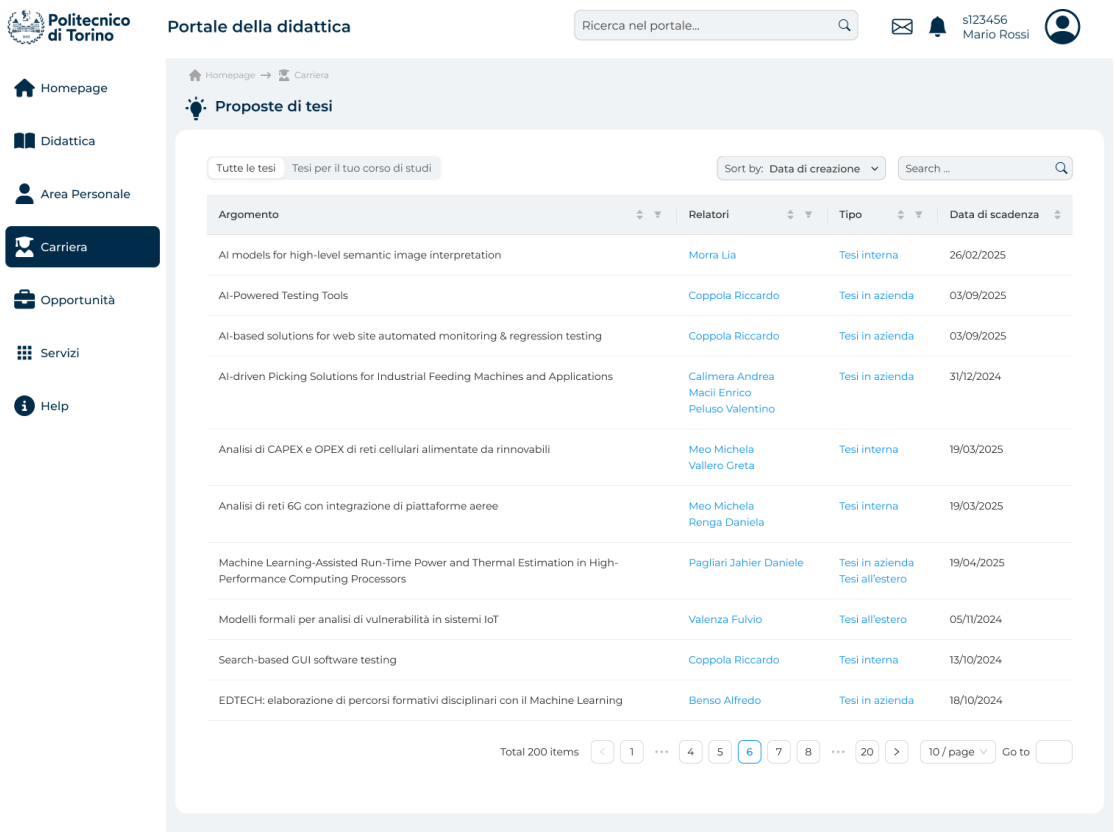


Figure 4.1: Thesis proposals - tabular view

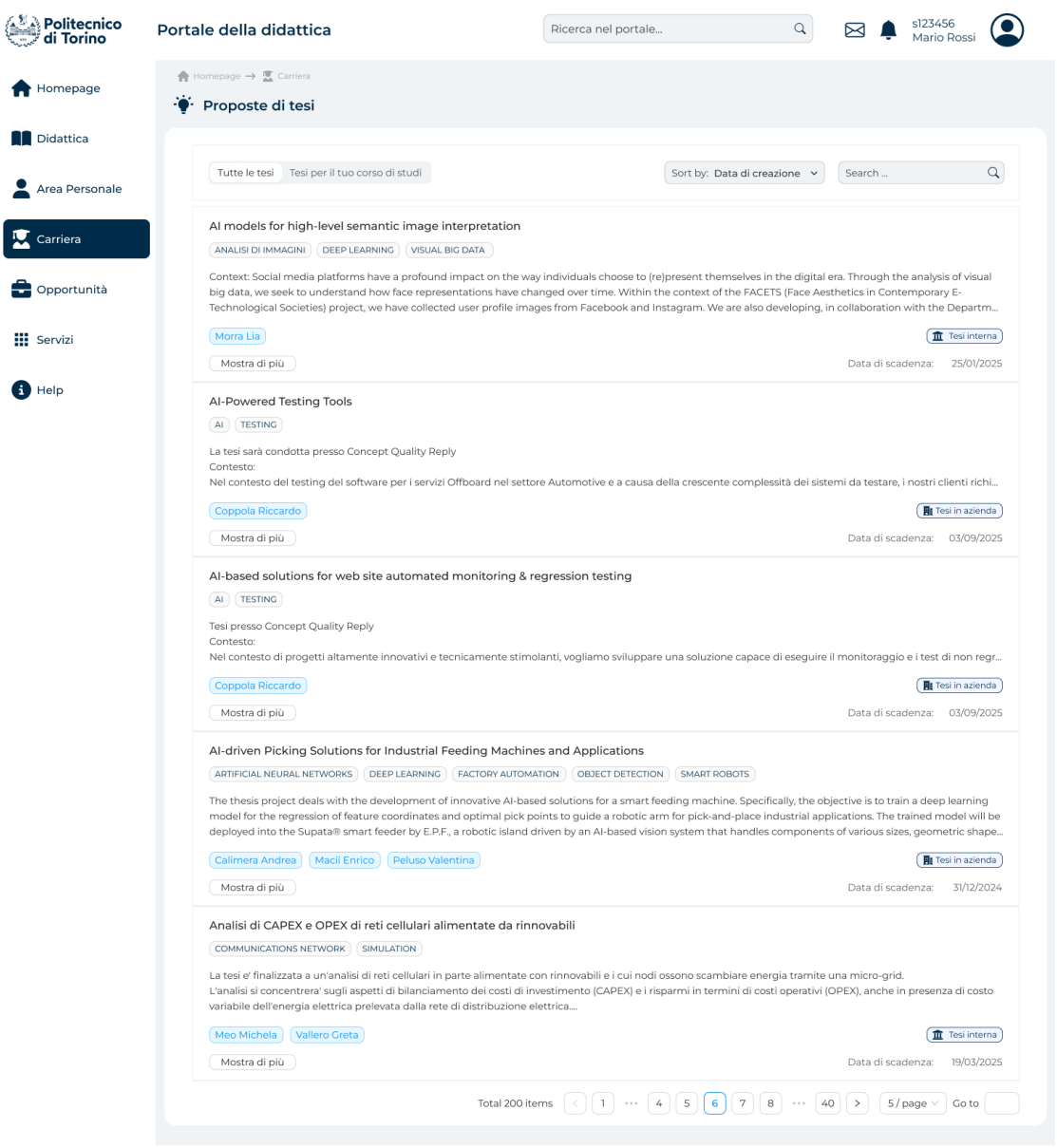


Figure 4.2: Thesis proposals - card list view

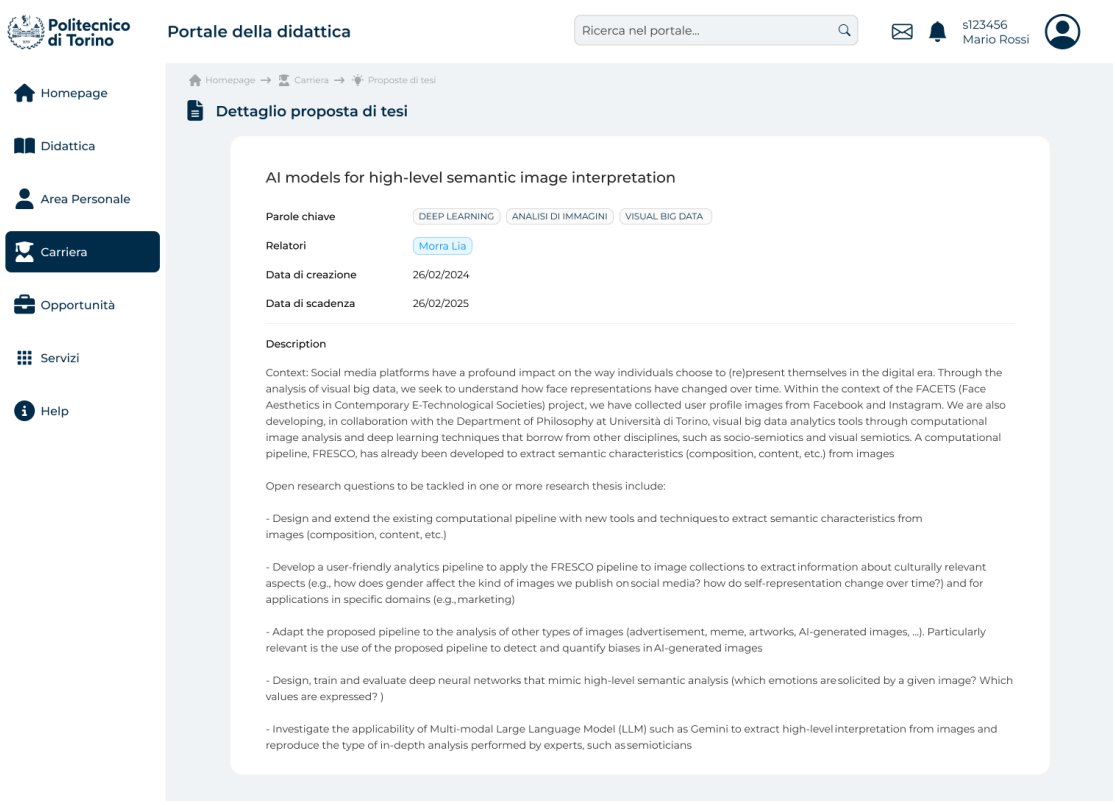


Figure 4.3: Thesis proposal detail view

Considering the amount of information associated with each thesis proposal, it was decided, also thanks to the opinion of the designers, that visualization using cards would be more effective. Although cards generally tend to occupy more space than a tabular representation, they allow a large amount of information to be organized in a more readable way, improving the usability of the interface. Furthermore, they provide greater flexibility in adapting to different devices, ensuring a better user experience on smaller screens and enhancing the responsiveness of the interface.

Since the development followed an iterative approach, as specified in chapter 6, the initial prototypes were progressively refined until a more mature version adhering to the design requirements was reached. The final design was the result of continuous improvements, which were also made possible through the integration of a design system that provided consistent guidelines for the definition of the interface.

The prototypes file can be found at the following link: `https://www.figma.com/design/dfCz596wWx4GiuJxmjAjvP/Prototypes?m=auto&t=SY4jJO57GeQ35JZC-1`

4.3 Components protoyping

The low-fidelity prototyping phase played a key role in defining the structure and functionality of the thesis proposal section. However, it was not immediately followed by a medium and high-fidelity prototyping phase, mainly due to the absence of a defined design system.

As mentioned earlier, the development of the new student portal was initiated from a thesis project that produced a high-fidelity prototype in which the information architecture was restructured but without an elaborate design system study. Therefore, when the work related to this project began, the design system of the new portal had not yet been defined and studied by the designers.

As a result, this project initially focused on implementing the essential functionalities identified. Only at a later stage, once the first versions of the design system were available, was it possible to proceed with the graphic and stylistic alignment of the interface. This phase was carried out in collaboration with design experts, who provided useful guidance to adjust the components within the established guidelines.

Components prototyping work involved the detailed study of all specific elements of the thesis proposal section. In particular, the following were designed and refined:

- **Information badges:** visual elements dedicated to the immediate identification of key information such as supervisors, keywords, thesis types, location (Italy or abroad), and environment (corporate or academic).

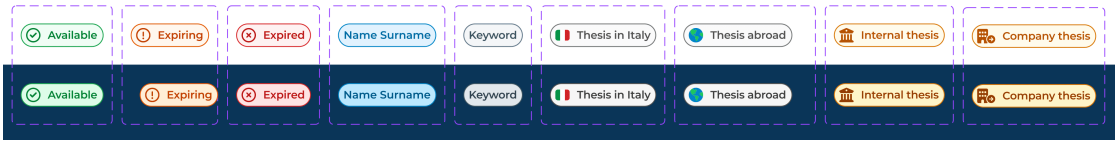


Figure 4.4: Thesis proposal badges

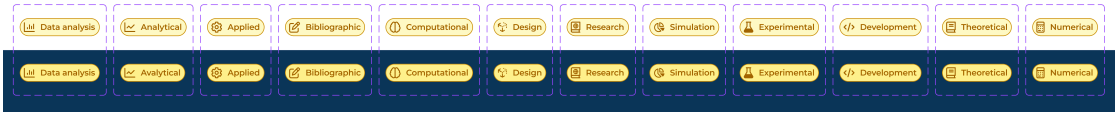


Figure 4.5: Thesis proposal types badges

- **Interactive buttons and controls:** designed to ensure consistency with

design system guidelines and to optimize the user experience in both visual modes.

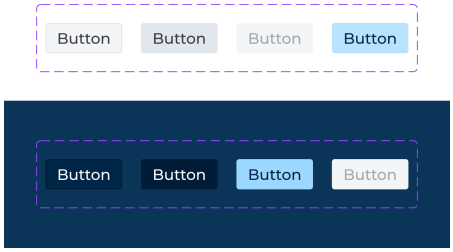


Figure 4.6: Default button

- **Other interface components:** including text elements, icons, and navigation sections, to provide a uniform and accessible user experience.

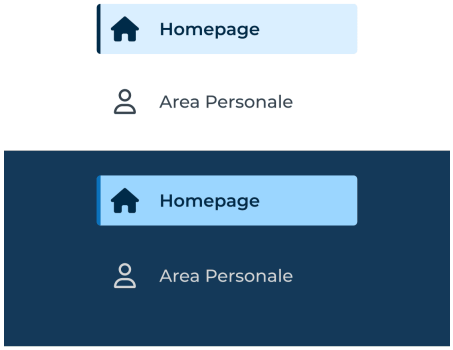


Figure 4.7: NavSide

Special attention was paid to the development of visual variants of the components, with the study and implementation of 'light' and 'dark' versions. This choice was necessary to ensure an optimal and comfortable visualization according to user preferences and different use conditions.

The design system file with components prototypes can be found at the following link: `https://www.figma.com/design/Mt4JM7E1upaFheroanNySF/Temp-Design-System?m=auto&t=cmKmhgRBW4zC568C-1`

Chapter 5

API Design

5.1 Overview of OpenAPI specifications

As was already done for the development of the student's app, given the non-uniformity and obsolescence of the current APIs, it was decided to design a new API specification based on the REST paradigm. This approach allows for improved interoperability, maintainability, and clarity of interfaces, bringing greater consistency to the entire system.

The **Representational State Transfer (REST)** paradigm, introduced by Roy Fielding in his 2000 dissertation [4], is based on the following architectural constraints:

1. **Uniform interface:** consistent communication interface between client and server.
2. **Client-server:** client and server applications must be able to evolve separately without depending on each other.
3. **Stateless:** every client request to the server must include all the information needed to process it, without relying on any server-side stored context. All session state is managed on the client side.
4. **Cacheable:** the data returned in a response is marked as either cacheable or non-cacheable (implicitly or explicitly). If a response is cacheable, a client cache has the right to reuse the response data for equivalent subsequent requests.

5. **Layered system:** REST allows the use of a layered system architecture, where, for example, APIs are deployed on server A, data is stored on server B, and requests are authenticated on server C. Typically, a client cannot know whether it is connected directly to the end server or to an intermediary along the way.

6. **Code on demand (optional):** servers have the ability to temporarily enhance or customize client functionality by sending executable code to the client.

The most important of these constraints, the one that distinguishes a REST architecture from other architectures, is the first one, that is, the emphasis on uniformity of interfaces between components. To achieve such uniformity, four constraints must be met:

1. **Resource identification:** a Uniform Resource Identifier (URI) is used to uniquely identify each resource involved in an interaction between components.

2. **Manipulation of resources through representation:** there is no direct access to resources, but a representation of them. This allows APIs to be used independently of the platform, making the system more accessible and allowing integration with other applications.

3. **Self-descriptive messages:** messages sent to the server contain all the information needed to understand the request. This means that messages do not have to depend on external state to be interpreted, simplifying communication and reducing the possibility of errors during information exchange.

4. **Hypermedia as the engine for managing application state:** hypermedia links are used to guide users through the available operations. In other words, a server response can include links to other possible actions, creating a system in which the client can dynamically navigate the API without having to know all the URLs or specific endpoints in advance.

In addition to deciding to follow the REST paradigm by designing RESTful APIs, the **OpenAPI Specification (OAS) 3.0** [5] standard was chosen because it allows for structured documentation that is easily understood by all developers involved.

OpenAPI is an API description format for REST APIs that allows endpoints, HTTP methods, parameters, response formats, and any error codes to be described in detail. Moreover, **Swagger** [6] provides a set of open source tools built around

the OpenAPI specification that can help design, build, document, and consume REST APIs as specified in section 5.3.

OpenAPI is used to help guarantee that API specifications are:

- **Consistent:** developers may engage with APIs consistently by standardizing the answer formats and endpoints.
- **Well-documented:** developers can rapidly find how to use the API thanks to the interactive and explicit documentation that is automatically created from the OpenAPI definition.
- **Easily testable:** the standard includes tools that reduce the amount of work needed for validations and troubleshooting by enabling testing of API endpoints directly from the documentation.

The adoption of a new API specification, based on the REST paradigm and defined through OpenAPI, leads to a more modular, maintainable, and well-documented system, which will not only improve the overall user experience but will ensure a safer and quicker evolution.

5.2 API structure and endpoints

To ensure secure and structured access to system resources, the following APIs have been defined (for students only):

- **Authentication:** API for students' login and logout, with the management of access tokens.
- **Thesis proposal management:** API for the display of thesis proposals with all associated attributes and various parameters for filtering, sorting, and pagination.
- **Thesis application management:** API to verify the student's eligibility to submit a thesis request and API to allow them to submit a thesis application.
- **Thesis management:** API to view the student's thesis.
- **Thesis conclusion confirmation request management:** API to verify the student's eligibility to send the thesis conclusion confirmation request, API for conversion of the thesis file in PDF-A format, and API for sending the request with all required attributes.

Authentication is implemented via **Bearer Token**, which guarantees that all requests to the API are protected and accessible only to authorized users. The APIs follow the **RESTful** paradigm, respecting the principle of separation between resources and operations. Therefore, the following rules were applied:

- Endpoints refer only to **system entities** and do not include the type of operation in the name (for example, /thesis-application and not /submit-thesis-application).
- Operations are defined exclusively through **HTTP methods**, such as:
  - **GET** for information retrieval
  - **POST** for the creation of new resources
  - **PUT** for change
  - **DELETE** for removal

This structure allows a consistent, scalable, and easily understandable API for developers. The OpenAPI file containing the complete specification is available in the following GitHub repository: `https://github.com/polito-ThesisManagement/api-spec`

5.2.1 Endpoint examples

This section presents examples of endpoints to manage thesis proposals. Each example includes the HTTP method, a description of the functionality, the query parameters, and the response schema.

1. **List of thesis proposals**

   Endpoint: /thesis-proposals

   Method: GET

   Description: Retrieves the list of all available thesis proposals, applying optional filters, sorting, and pagination based on query parameters.

   Query parameters:

   - lang *(optional)*: The language of the thesis proposals ('it' for Italian, 'en' for English, default: Italian).
   - page *(optional)*: The page number for pagination (default: 1).
   - limit *(optional)*: The number of results per page (default: 10).

- search *(optional)*: A search string to filter proposals by topic or description.
- isInternal *(optional)*: A boolean to filter by internal (true) or external (false) proposals.
- isAbroad *(optional)*: A boolean to filter by abroad (true) or Italian (false) proposals.
- teacherId *(optional)*: The ID of the teacher to filter proposals supervised by a specific teacher.
- keywordId *(optional)*: The ID of a keyword to filter proposals linked to specific topics.
- typeId *(optional)*: The ID of the proposal type (e.g., research, experimental) to filter proposals with a specific type.
- sortBy *(optional)*: The field to sort results by (topic, description, creation_date, expiration_date).
- orderBy *(optional)*: The sorting order, either ascending ('asc') or descending ('desc').

Response schema: A successful response returns an object containing the data property, which is an array of objects conforming to the ThesisProposalOverview schema.

2. **List of targeted thesis proposals**

   Endpoint: /thesis-proposals/targeted

   Method: GET

   Description: Retrieves the list of thesis proposals targeted to the authenticated student, based on their degree program. It supports the same query parameters as the previous endpoint for pagination, filtering, and sorting.

   Query parameters: The query parameters are the same as those described for the previous endpoint.

   Response schema: The response structure is identical to that of the /thesis-proposals endpoint, returning an array of ThesisProposalOverview objects.

3. **List of thesis proposal types**

   Endpoint: /thesis-proposals/types

   Method: GET

   Description: Retrieves all available thesis proposals types. It supports translation and filtering through query parameters.

   Query parameters:

- lang *(optional)*: The language of the thesis types ('it' for Italian, 'en' for English, default: Italian).
- search *(optional)*: A search string to filter types.

Response schema: Returns an array of objects conforming to the ThesisProposalType schema, which includes the type's identifier and description.

4. **List of thesis proposals keywords**

   Endpoint: /thesis-proposals/keywords

   Method: GET

   Description: Retrieves all keywords associated with thesis proposals. It supports translation and filtering through query parameters.

   Query parameters:

   - lang *(optional)*: The language of the keywords ('it' for Italian, 'en' for English, default: Italian).
   - search *(optional)*: A search string to filter keywords.

   Response schema: Returns an array of objects conforming to the ThesisProposalKeyword schema, which includes the keyword's ID and corresponding text.

5. **List of teachers**

   Endpoint: /thesis-proposals/teachers

   Method: GET

   Description: Retrieves the list of teachers associated with thesis proposals, allowing filtering through the query parameter.

   Query parameters:

   - search *(optional)*: A search string to filter teachers.

   Response schema: Returns an array of objects conforming to the TeacherOverview schema, containing essential teacher information (name, surname, and ID).

6. **Thesis proposal details**

   Endpoint: /thesis-proposals/:thesisProposalId

   Method: GET

   Description: Retrieves the details of a specific thesis proposal identified by the thesisProposalId parameter.

   Query parameters:

- lang *(optional)*: The language of the thesis proposal details ('it' for Italian, 'en' for English, default: Italian).
- thesisProposalId: The unique identifier of the thesis proposal.

Response schema: The response consists of one object conforming to the ThesisProposal schema, containing all attributes of a thesis proposal.

5.2.2 Schema Definitions

This section presents a description of the reference schemas mentioned in the endpoints examples. Each schema is illustrated with a table of fields details, and a JSON example.

ThesisProposalOverview

This schema summarizes the main information about a thesis proposal, including:

Field	Type	Description
id	integer	Proposal identifier
topic	string	Topic title
description	string	Brief description
creationDate	string (date)	Proposal creation date
expirationDate	string (date)	Proposal expiration date
isInternal	boolean	Flag indicating an internal proposal
isAbroad	boolean	Flag indicating an abroad proposal
keywords	array of ThesisProposalKeyword	Associated keywords
types	array of ThesisProposalType	Proposal types
supervisor	TeacherOverview	Supervising teacher information
internalCoSupervisors	array of TeacherOverview	Internal co-supervisors

Table 5.1: `ThesisProposalOverview` fields details

```
{
  "id": 1,
  "topic": "Gamification for test refactoring",
  "description": "This thesis proposes an in-depth
   investigation into the application of gamification in the
    context of test refactoring.",
  "creationDate": "2024-10-16T10:35:10.547Z",
  "expirationDate": "2025-10-16T23:59:59.999Z",
  "isInternal": true,
  "isAbroad": false,
  "keywords": [{ "id": 1, "keyword": "Testing" }],
  "types": [{ "id": 1, "type": "Research" }],
  "supervisor": {
    "id": 12345,
    "firstName": "Marco",
    "lastName": "Torchiano"
  },
  "internalCoSupervisors": [{
    "id": 23456,
    "firstName": "Riccardo",
    "lastName": "Coppola"
  }]
}
```

Listing 5.1: ThesisProposalOverview schema example

ThesisProposal

Field	Type	Description
supervisor	Teacher	Supervisor details
internalCoSupervisors	array of Teacher	List of internal co-supervisors
externalCoSupervisors	string	Information about external co-supervisors
attachment	Attachment	Any attached file, if present
link	string	External link related to the proposal
requiredSkills	string	Skills required to apply for the proposal
additionalNotes	string	Additional notes regarding the proposal

Table 5.2: `ThesisProposal` fields details

```
{
  "id": 1,
  "topic": "Gamification for test refactoring",
  "description": "This thesis proposes an in-depth
   investigation into the application of gamification in the
    context of test refactoring.",
  "creationDate": "2024-10-16T10:35:10.547Z",
  "expirationDate": "2025-10-16T23:59:59.999Z",
  "isInternal": true,
  "isAbroad": false,
  "keywords": [{ "id": 1, "keyword": "Testing" }],
  "types": [{ "id": 1, "type": "Research" }],
  "supervisor": {
    "id": 12345,
    "firstName": "Marco",
    "lastName": "Torchiano",
    "role": "Teacher",
    "email": "marco.torchiano@polito.it",
    "profileUrl": "",
    "profilePictureUrl": "",
    "facilityShortName": "DAUIN"
  },
  "internalCoSupervisors": [{
    "id": 23456,
    "firstName": "Riccardo",
    "lastName": "Coppola",
    "role": "Fixed-term researcher Law 240/10 art.24-b",
    "email": "riccardo.coppola@polito.it",
    "profileUrl": "",
    "profilePictureUrl": "",
    "facilityShortName": "DAUIN"
  }],
  "externalCoSupervisors": "",
  "attachment": {
    "id": 0,
    "filename": "requirements.pdf",
    "mimeType": "application/pdf",
    "sizeInKiloBytes": 1024
  },
  "link": "",
  "requiredSkills": "Development with OOP languages (
   preferably Java), Testing fundamentals",
  "additionalNotes": ""
}
```


Listing 5.2: ThesisProposal schema example

ThesisProposalKeyword

This schema describes thesis proposal keywords with the following fields:

Field	Type	Description
id	integer	Keyword identifier
keyword	string	Keyword text

Table 5.3: `ThesisProposalKeyword` fields details

```
{
  "id": 1,
  "keyword": "Testing"
}
```

Listing 5.3: ThesisProposalKeyword schema example

ThesisProposalType

This schema describes thesis proposal types with the following fields:

Field	Type	Description
id	integer	Type identifier
type	string	Type text

Table 5.4: `ThesisProposalType` fields details

```
{
  "id": 1,
  "type": "Research"
}
```

Listing 5.4: ThesisProposalType schema example

TeacherOverview

This schema provides essential information about a teacher:

Field	Type	Description
id	integer	Teacher identifier
firstName	string	Teacher first name
surname	string	Teacher last name

Table 5.5: `TeacherOverview` fields details

```
{
  "id": 12345,
  "firstName": "Marco",
  "lastName": "Torchiano"
}
```

Listing 5.5: TeacherOverview schema example

Teacher

This schema extends TeacherOverview with additional attributes:

Field	Type	Description
role	string	Teacher role
email	string	Teacher email
profileUrl	string	Teacher profile URL
profilePictureUrl	string	Teacher profile picture URL
facilityShortName	string	Name of the teacher's facility

Table 5.6: `Teacher` fields details

```
{
  "id": 12345,
  "firstName": "Marco",
  "lastName": "Torchiano",
  "role": "Docente",
  "email": "marco.torchiano@polito.it",
  "profileUrl": "https://www.dauin.polito.it/personale/
   scheda/(matricola)/003019",
  "profilePictureUrl": "https://www.dauin.polito.it/
   personale/scheda/(matricola)/003019/foto",
  "facilityShortName": "DAUIN"
}
```

Listing 5.6: Teacher schema example

Attachment

This schema describes the attached file with details such as filename, type, and size.

Field	Type	Description
id	integer	Attachment identifier
fileName	string	Name of the attached file
mimeType	string	File type
sizeInKiloBytes	integer	File size in Kilobytes

Table 5.7: `Attachment` fields details

```
{
  "id": 0,
  "filename": "requirements.pdf",
  "mimeType": "application/pdf",
  "sizeInKiloBytes": 1024
}
```

Listing 5.7: Attachment schema example

5.3 Documentation and Swagger UI integration

API documentation was generated using the **OpenAPI Specification (OAS) 3.0** standard, which allows for a clear and precise description of the endpoints, HTTP methods, parameters, responses, and error codes associated with each resource.

To make documentation easy to use, the **Swagger UI** has been integrated. This tool allows an interactive view of the APIs, allowing developers to execute requests directly from the browser and verify the responses in real time. In addition, the documentation can be viewed with a clear and easily navigable graphical interface, which facilitates testing and development work. To obtain a readable interface, you can choose between two options:

1. **Import the .yaml file** of the specification directly into Postman to test endpoints.
2. **Run Swagger UI locally** via Docker.

The documentation is automatically updated whenever the OpenAPI specification is changed, ensuring that developers always have access to the latest and most accurate version of the API. In addition, the ability to test endpoints directly from documentation greatly reduces development time and facilitates debugging.

Full documentation and integration with Swagger UI are available in the GitHub repository previously linked.

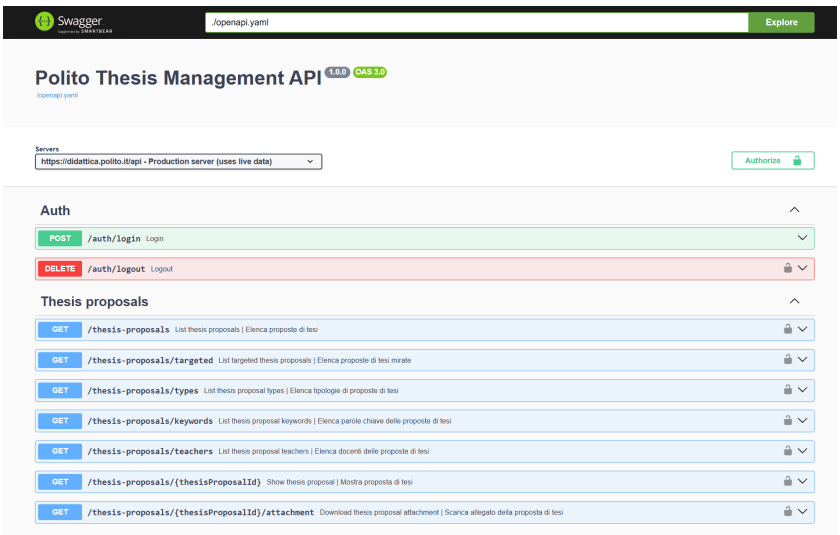


Figure 5.1: Swagger UI interface

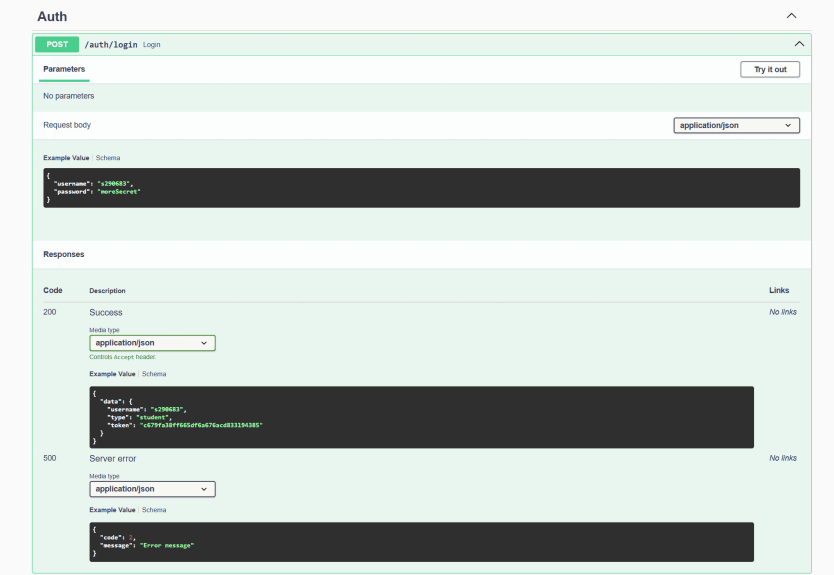


Figure 5.2: Swagger UI - API documentation

Chapter 6

Development

6.1 Project management

Software development can follow various methodologies, including the waterfall method, agile development, or iterative models. Due to this project's characteristics, agile development was chosen because work can be broken down into iterations, and feedback can be collected at the end of each sprint to guide future work in the right direction.

The main objective of the Agile method is to release high-quality solutions through a development process that adapts quickly to changes in requirements and sets the goal of continuous improvement of the final product. In the specifics of this thesis project, the Agile approach is particularly appropriate because direct student involvement and constant feedback are key aspects of redesigning and improving the thesis portal.

6.1.1 Agile theoretical background

Agile methodology encompasses a set of development techniques united by common principles that originated in the early 2000s. The core values of this family of methods are defined in the 'Manifesto for Agile Software Development' [7] and are as follows:

- **Individuals and interactions** over processes and tools

- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan.

Agile development focuses on 12 principles that establish the primary goals:

1. The highest priority is to satisfy the customer through frequent and continuous release of quality software.
2. Changes in requirements should be accommodated even if late in the development phase.
3. It is necessary to release working software frequently by trying to prefer shorter time intervals.
4. Developers and entrepreneurs should work together on a daily basis.
5. Projects should be carried out with motivated people. Therefore, they should be provided with the environment and support they need, and their ability to get the job done should be trusted.
6. The most efficient and effective way to communicate with and within a development team is face-to-face conversation.
7. The first measure of progress is working software.
8. Agile processes promote sustainable development. Sponsors, developers, and users should be able to maintain a steady pace indefinitely.
9. Continued focus on technical excellence and good design enhances agility.
10. Simplicity, understood as the art of maximizing the amount of work not done, is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team must reflect on how to become more effective and then adjust and adapt its behavior accordingly.

6.1.2 Project iterations and workflow

Although agile development methods are numerous, Scrum is the most widely used. In the context of this project, a Scrum-inspired approach was adopted. While not formally including the roles of Product Owner and Scrum Master, the work was organized into sprints, iterations of fixed duration that allowed for incremental and iterative planning, development, and review. Each sprint included:

- **Sprint planning:** Initial planning phase to define the user stories to be implemented and the related task breakdown.
- **Daily Stand-Ups:** A series of regular meetings to monitor progress.
- **Sprint Review:** Final review phase to gather feedback and adapt the following work.

This approach made it possible to maintain a steady pace, incorporate user feedback, and progressively improve the work, as established by the core principles of Agile.

Each sprint focused on a specific objective to ensure continuous improvement and important feedback collected from users. Below is an overview of the main iterations:

First sprint

The first iteration focused on building an initial version of the interface based on real data to analyze the distribution of values of a thesis proposal's attributes.

The work started by connecting a public API that allowed the download of thesis proposals related to a specific research group. Through this API, it was possible to create a first version of the interface, which, although not complete in terms of functionality, enabled the visualization of all the fields related to a thesis proposal while exploring the most efficient way to display them.

Because of this iteration and the possibility of seeing the actual data applied to the previously developed prototype, it was possible to notice some issues related to data structuring. In particular, the need to remove the concept of research groups emerged, as it did not represent essential information for users and risked unnecessarily complicating thesis proposals consultation. In addition, when analyzing the actual data from the thesis proposals, excessive variability was observed in the

'types' field, which was populated with very heterogeneous values, making clear and consistent categorization difficult.

With these insights, the sprint laid the foundation for the normalization of information and highlighted the need for a more structured back-end, which was then developed in the next iteration.

At the end of the sprint, the academic supervisor provided feedback on how to improve the clarity and organization of the information presented.

The code developed during this sprint is available at: `https://github.com/polito-ThesisManagement/Portale-PoliTO/releases/tag/v0.1.0-alpha`

Second sprint

In the second iteration, the main objective was to implement a test back-end that would allow for more structured and organized information management on thesis proposals. Implementing the back-end has proved fundamental to integrating pagination, filters, and sorting directly on the server side, which aligns with good software development practices.

Using the test back-end also allowed for validating and optimizing the APIs defined in the OpenAPI specification, ensuring greater consistency and control in data management and its flow through the system.

Another important phase of this sprint was normalizing thesis types. The heterogeneity of thesis proposal types, which emerged during the first sprint, was addressed by analyzing the actual data and defining a limited set of standardized types that could contain most of the values present. This approach reduced ambiguity and improved consistency in classifying thesis proposals.

The granularity of the display of thesis proposals has been improved to make the information more relevant to students. In particular, the system is designed to prioritize thesis proposals addressed to the course of study of the authenticated student, showing these in the foreground compared with the others. This allows students to quickly view the theses most relevant to their educational path without excluding access to proposals for other courses of study, which are nevertheless presented in the background.

To ensure effective implementation of this functionality, it has been established that a teacher can configure the granularity of proposals by selecting a Collegio

(such as Architecture and Design Collegio, Biomedical Engineering Collegio, ...) as the maximum level and a specific course of study (such as Computer Engineering, Mechatronic Engineering, ...) as the minimum level. In this way, the system can highlight the proposals of theses aimed at the student's academic path, ensuring a personalized and targeted experience. In addition, since courses with the same name but different IDs exist, mainly because the language of course delivery changes, containers have been designed to leave the teacher the possibility of choosing only between courses with different names. In this way, the teacher can select courses according to their name, while the thesis proposal will address all the courses with that name without needing to select them manually.

Furthermore, this sprint also focused on implementing all filtering and sorting functionalities on the back-end. These enhancements enable students to dynamically filter and sort thesis proposals based on various criteria, ensuring a more intuitive and efficient browsing experience.

At the end of the sprint, the designers were requested to provide feedback to assess the clarity of the interface and identify any improvements in the presentation of data.

The code developed during this sprint is available at: `https://github.com/polito-ThesisManagement/Portale-PoliTO/releases/tag/v0.2.0`

Third sprint

The third sprint's primary objective was to optimize the user experience and streamline the portal design based on feedback from previous iterations. This involved an in-depth review of the layout, ensuring that it was consistent and intuitive and that the information layout was clear and easily accessible to all users.

To achieve this, the input of design experts, who provided a detailed mock-up of the interface, was crucial. This mock-up was the basis for a complete review of all the graphic elements, which have been rearranged and improved to adhere to the design system developed in parallel with the new teaching portal. A uniform design system has ensured greater visual and functional consistency, improving app perception and facilitating a more fluid and satisfying user experience.

At the end of this iteration, the focus has shifted to the direct validation of the interface with students through a usability test phase. This step will allow any unidentified critical or improvement areas to be identified. Students will be presented with real-life use scenarios, and their interaction with the portal will be

assessed to identify potential difficulties and improve the interface further.

This iteration marks an important transition from design to practical validation. It ensures that the portal fully meets users' needs and provides an optimal experience at every process stage.

The code developed during this sprint is available at: `https://github.com/polito-ThesisManagement/Portale-PoliTO/releases/tag/v1.0.0-beta`

6.1.3 Project management tools

Since the project will be released under an open source license, it was decided to use **GitHub** as a hosting service and source code management. To ensure a transparent and collaborative work structure, a dedicated organization has been created on GitHub called '**polito-ThesisManagement**', which hosts the following repositories:

1. **api-spec** (at `https://github.com/polito-ThesisManagement/api-spec`): repository containing the openAPI specification and the necessary instructions for its understanding and use.
2. **Portale-PoliTO** (at `https://github.com/polito-ThesisManagement/Portale-PoliTO`): repository based on the existing work related to the redesign of the user experience of the teaching portal, which was then adapted and expanded to support the new project.

The organization is available at the following link: `https://github.com/polito-ThesisManagement`

GitHub Projects was used to manage and monitor the project's progress. This task management system allows users to organize their work through a board that is entirely customizable according to the development method adopted. Within the previously created organization, a board was created and divided into the following columns:

- **Product Backlog (User stories)** → contains all the user stories still to be implemented
- **Sprint Backlog (User stories)** → contains the selected user stories in the current sprint.

- **Todo (Tasks)** → contains all the tasks to be implemented, derived from user stories and defined in the Sprint Planning phase.
- **In Progress (Tasks)** -> contains the currently under-development tasks.
- **To Verify (Tasks)** → contains the tasks for which a pull request is active, that is, the developed functionalities ready to be revised before integration in the main branch.
- **Done (Tasks)** → contains completed tasks that have been merged into the main branch.
- **Done (User Stories)** → contains completed user stories, which are those whose tasks have been finished.

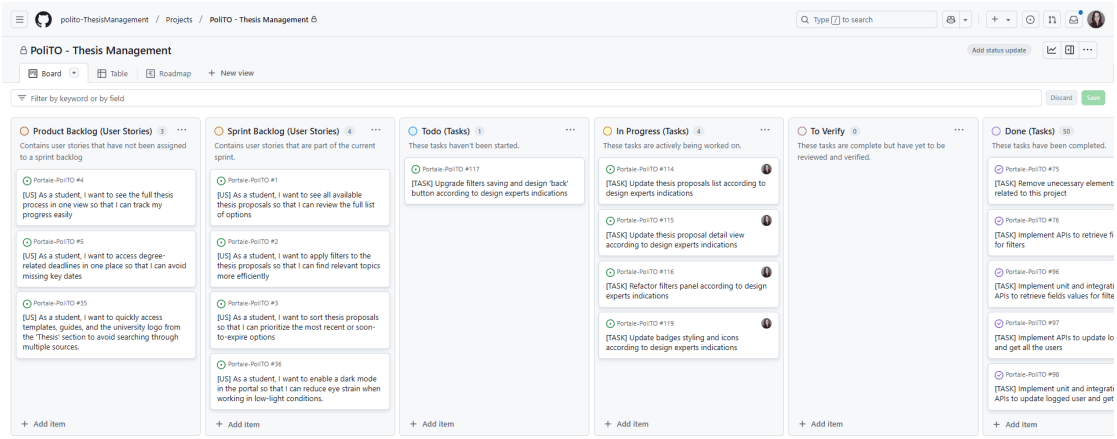


Figure 6.1: Project board view

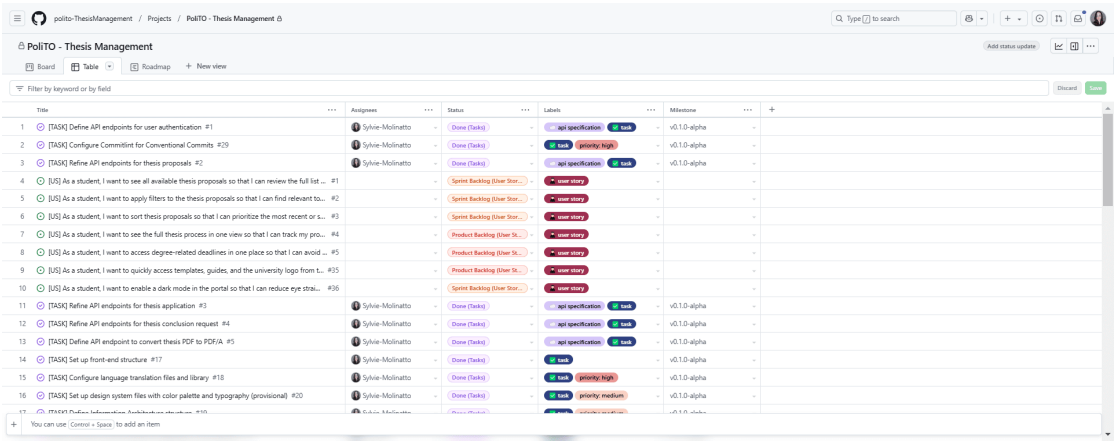


Figure 6.2: Project table view

Before the start of the first sprint, all user stories to be developed were uploaded in the Product Backlog as GitHub Issues of type 'user story'. This ensured a clear view of the required functionalities and facilitated the division into tasks during development.

To standardize issue creation and maintain consistency, two GitHub Issue templates were defined:

- **User Story template:** used when creating new user stories.

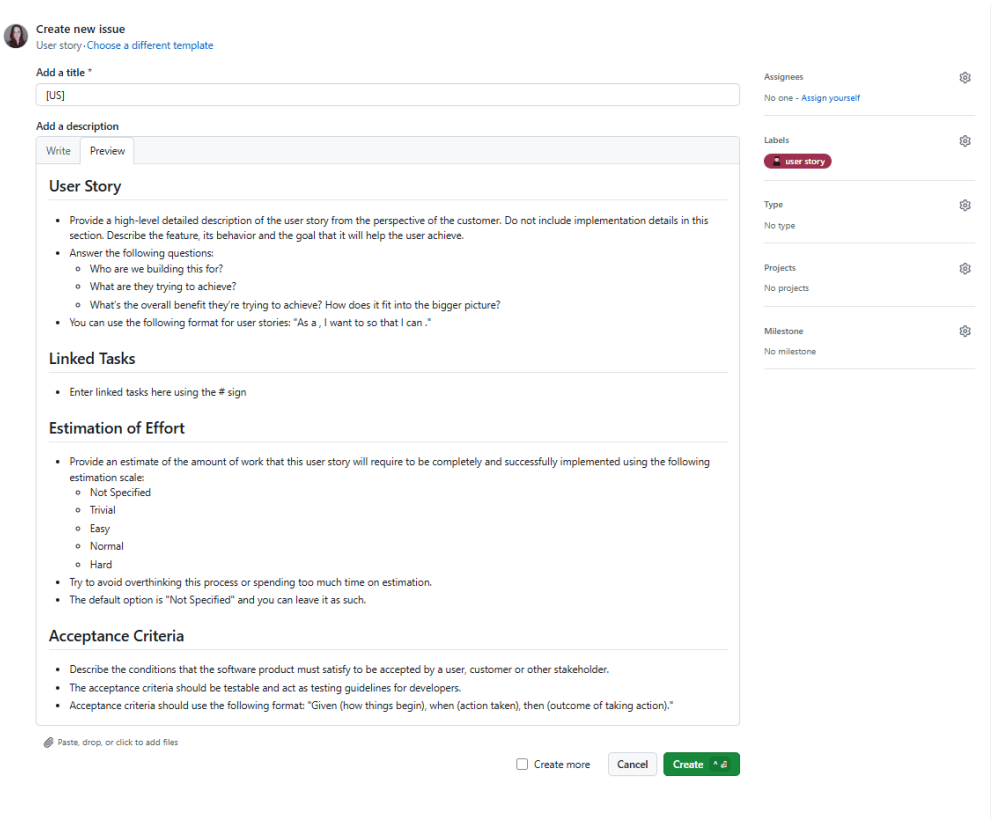


Figure 6.3: User Story template

- **Task template:** used when creating new tasks.

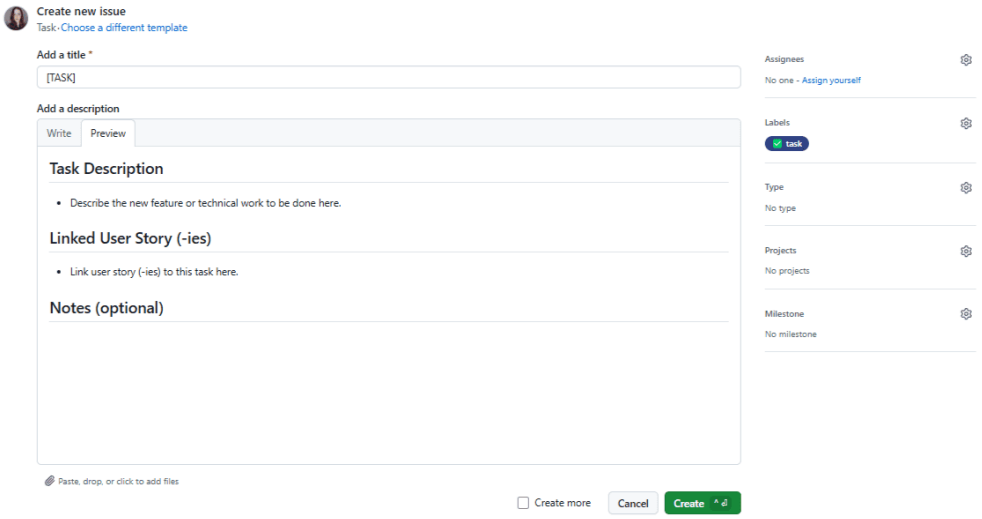


Figure 6.4: Task template

During the sprints, these were the steps for board management:

1. **Sprint planning:** At the start of each sprint, the team selected user stories to implement and moved them into the Sprint Backlog.
2. **Subdivision into tasks:** Each user story was divided into smaller tasks, created as GitHub Issues of type 'task'.
3. **Work Progress:** During development, the tasks were progressively moved to the appropriate columns (from To Do to In Progress, then To Verify, and finally Done)
4. **Completion of User Stories:** Once all the tasks in a user story were completed and verified, the user story was marked as completed and moved to the Done column (User Stories).

Using GitHub Projects allowed a clear view of the project status, facilitating monitoring activities, coordination among team members, and tracking work done.

The board is available at the following link: `https://github.com/orgs/polito-ThesisManagement/projects/5`

6.2 Project structure

The project follows a modular structure, dividing the codebase into distinct folders for the front-end, back-end, and database. Each module is organized to guarantee maintainability, scalability, and a clear separation of concerns.

Figure 6.5: Project structure

The project structure includes additional configuration files for CI/CD, code quality, and development workflows, such as:

- **.github/**: GitHub Actions for automated workflows.
- **.husky/**: Pre-commit hooks to enforce code standards.
- **sonar-project.properties**: Configuration for SonarQube code analysis.
- **Linting and formatting files**: .prettierrc, .eslintrc.mjs.

6.2.1 Front-end structure

The front-end is structured to manage the client side of the application.

Figure 6.6: Front-end structure

It includes:

- **cypress/**: End-to-end testing setup.
- **public/**: Static assets and publicly accessible files.
- **src/**: Main source directory containing:
  - **assets/**: Images, icons, and other static resources.
  - **components/**: Reusable UI components.

- **data/**: Local static data, in particular, the portal navigation structure.
- **pages/**: Main pages of the application.
- **styles/**: CSS and styling-related files.
- **utils/**: Utility functions for shared logic.
- **API.js**: Defines API calls and endpoints.
- **App.js**: Main application entry point.
- **i18n.js**: Internationalization settings.
- **index.js**: Main entry point rendering the React application.

6.2.2 Back-end structure

The back-end handles API requests, business logic, and interaction with the database.

Figure 6.7: Back-end structure

It is structured as follows:

- **src/**: The main source directory containing:
  - **config/**: Contains configuration file for database connection with Sequelize.
  - **controllers/**: Defines request handling logic for different API endpoints.
  - **models/**: Represents the data models interacting with the database.
  - **routers/**: Manages API routes and endpoints.
  - **schemas/**: Defines validation schemas for request data.
  - **utils/**: Contains helper functions.
  - **app.js and index.js**: Main entry files to set up and start the server.
- **test/**: Contains unit and integration tests.
- **.env and .env.example**: Environment variable configuration files.

6.2.3 Database structure

The database directory contains SQL scripts for managing the database schema and test data.

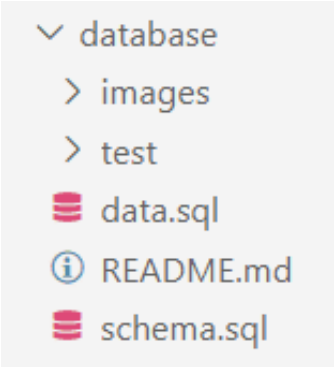


Figure 6.8: Database structure

It includes:

- **images/**: Includes database diagram.
- **test/**: Contains test database scripts:

- **data_test.sql and schema_test.sql** for testing purposes.
- **data.sql and schema.sql**: Define the main database schema and data insertion.
- **README.md**: Documentation for database setup and usage.

6.3 Technological choices

The project, being composed of front-end, back-end, and database, consists of several technologies. For each of the three main components, specific technologies have been adopted to ensure efficiency, scalability, and maintainability.

In addition, tools have been employed to enforce code quality, maintain a consistent style, and monitor the security and reliability of the software.

6.3.1 Code quality and conventions

To ensure a structured and maintainable development, the following tools were used:

- **Commitlint:** used to ensure that commit messages follow a predefined standard to improve the tracking of changes.
- **Eslint and Prettier:** used for code formatting, starting from the import order up to the choice of type of quotation marks used.
- **SonarQube:** used to perform code quality analysis and identify security, maintainability, and reliability issues. It also allows setting the Quality Gates, which are minimum quality criteria to avoid releasing defective code.

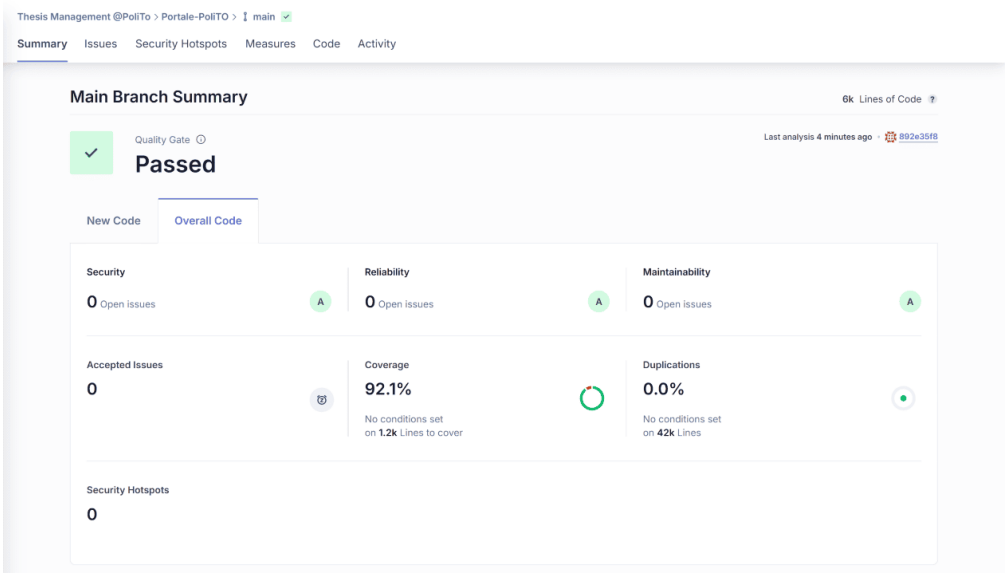


Figure 6.9: Sonar Quality Gates

6.3.2 Front-end technologies

The front-end was developed using **React**, a JavaScript library to build modular and responsive user interfaces, together with **Bootstrap**, a CSS framework used to create modern and responsive user interfaces.

Additional technologies adopted include:

- **Axios:** library for HTTP requests management, used for communication with the back-end.
- **FontAwesome:** vector icon library used in accordance with the design system.
- **i18next:** solution for localization management, which allows the interface to be offered in Italian and English.
- **Moment:** library for dates manipulation and formatting.
- **PropTypes:** library for property types validation in React components.

End-to-end tests were conducted using the **Cypress** framework, which enables the writing of tests that simulate user interactions with the application, as explained in the next chapter.

6.3.3 Back-end technologies

The back-end was developed with **Node.js**, using **Express.js** as a framework for API management.

The main technologies adopted include:

- **Express.js:** web framework for Node.js that simplifies the creation of RESTful APIs.
- **Body-Parser:** middleware for the analysis of HTTP request bodies, making data accessible through req.body.
- **CORS:** middleware to enable Cross-Origin Resource Sharing (CORS), allowing resources to be requested from different domains.
- **Dotenv:** library for environment variables management via .env file.
- **Zod:** Library for data validation and parsing. It is used to ensure that the data follows specific patterns.
- **Cross-Env:** library for setting environment variables in a cross-platform way. It is used to ensure that environment variables are set up correctly on different operating systems.

The following libraries were used for unit and integration tests:

- **Jest:** JavaScript testing framework. It is used to write and run unit and integration tests.
- **Jest HTML Reporters:** Jest reporter, which generates test reports in HTML format. Facilitates the display of test results.
- **Supertest:** library to test HTTP APIs. It is used to write end-to-end tests for the Express APIs.

6.3.4 Database technologies

The project database was built with **MySQL 8.4.3**, a relational database chosen for its reliability and performance.

For interaction with the database, the following technologies were used:

- **MySQL2:** client for MySQL that supports both promises and callbacks, improving the management of asynchronous queries.
- **Sequelize:** ORM (Object-Relational Mapping) to simplify model definition and database operations.

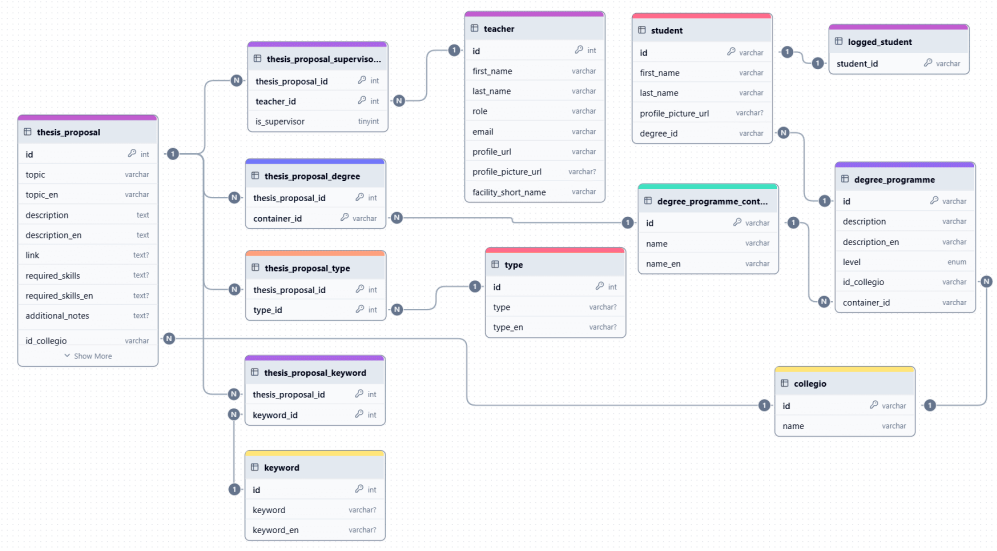


Figure 6.10: Database diagram

6.4 Output

The resulting interface is illustrated in the following screenshots and is structured as follows: upon accessing the *Career* section from the sidebar, a card provides access to the section dedicated to thesis proposals.

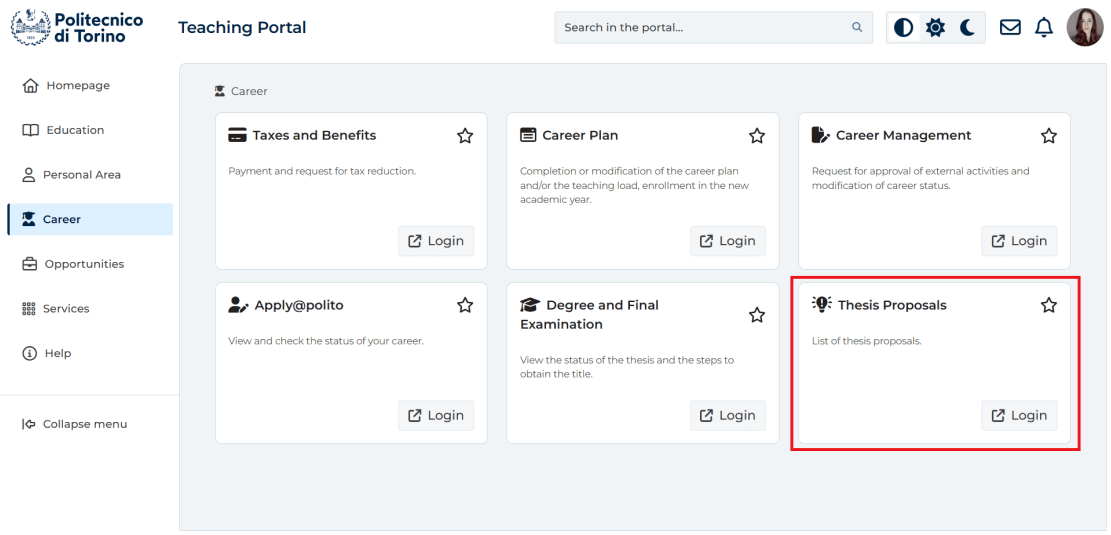


Figure 6.11: Accessing the thesis proposals section from the *Career* page

Within the proposals section, the first visible element is a **breadcrumb**, which helps the user orient themselves, as this page is on the third navigation level. Successively, there is the **filtering and sorting section**, which allows the user to view only the proposals of interest by applying specific filters and sorting them according to the preferred criteria. By default, the student sees the proposals targeted to his course of study, but can choose to view all the proposals available for other study courses if he wishes to explore additional options.

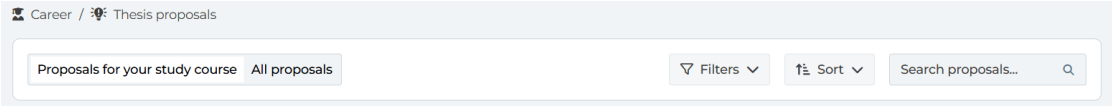


Figure 6.12: Breadcrumb and filtering/sorting section

Regarding filtering, one or more criteria can be selected from those represented by the badges by opening the filter dropdown menu and choosing the desired options for each filter type. Once filters have been applied, summary badges appear, allowing to quickly view active filters and remove them with a simple click. Proposals can be sorted by topic, description, creation date, and expiration date using the sort dropdown menu. It is also possible to perform a text search using the search bar.

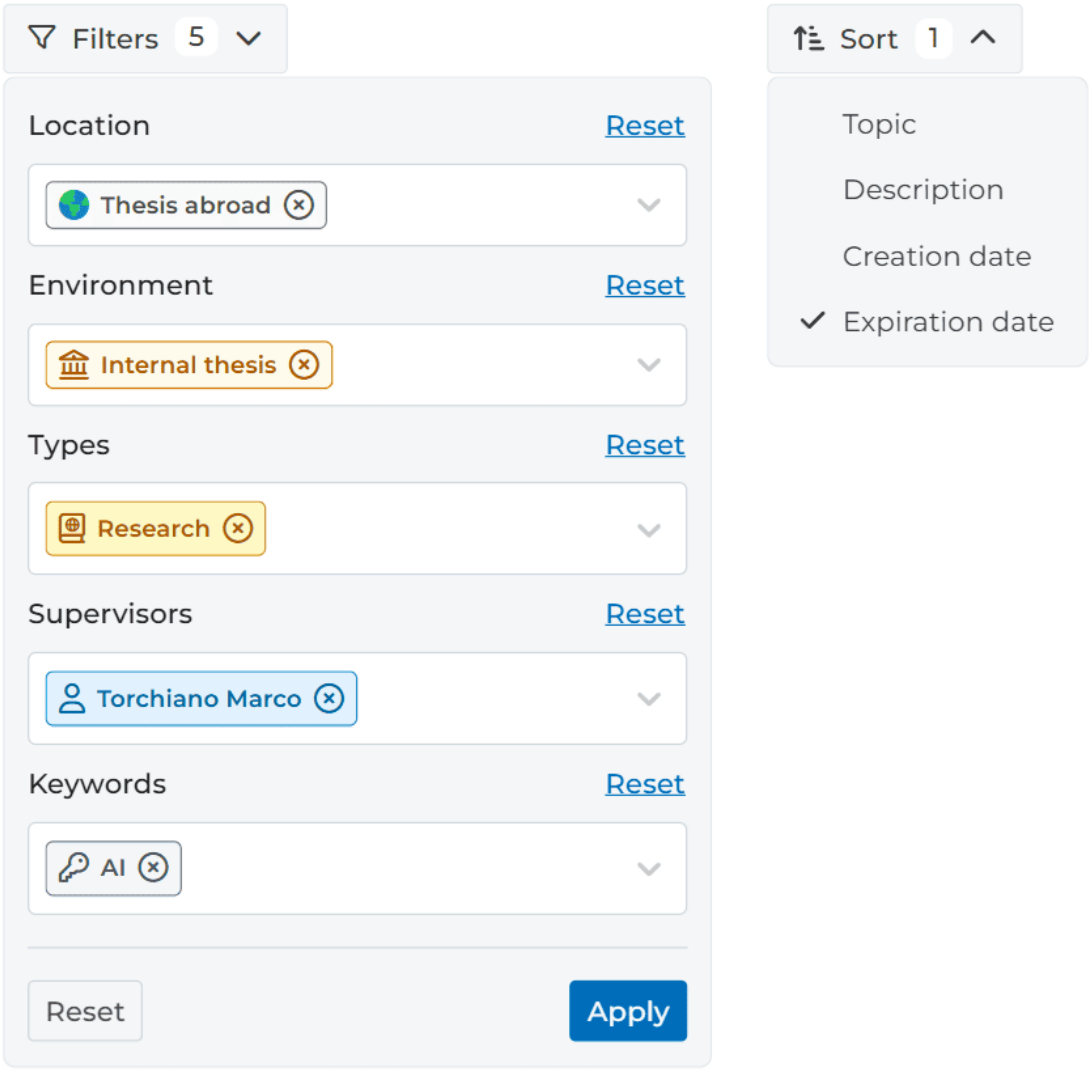


Figure 6.13: Filters and sort dropdowns

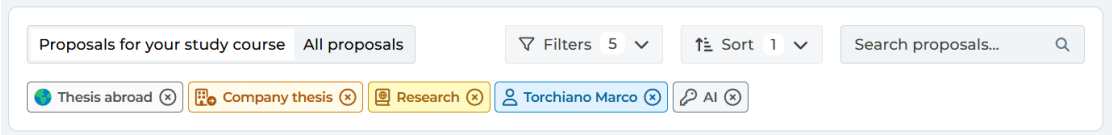


Figure 6.14: Filtering/sorting section with summary badges

After the research and refinement section, there is the **list of thesis proposals**, which is **paginated by default at ten items per page**. The student can change

the number of proposals displayed and navigate between pages using the commands at the bottom of the list.

Figure 6.15: Thesis proposals list pagination controls

The proposals are presented in the form of cards arranged in a grid, a design choice that avoids excessive space occupation and enhances readability.

The card header includes the following elements:

- **Topic** → Positioned in the first line, left-aligned.
- **Location icon** (Italian flag or world) → Indicates whether the thesis is to be completed in Italy or abroad; placed in the first line, right-aligned.

Below the header, several **informative badges** provide key details:

- **Proposal status** → *Available* or *Expiring* (if the proposal will expire within the next 14 days).
- **Thesis environment** → *Internal* (conducted at the university) or *In company* (external collaboration).
- **Thesis type** → Represented by zero or more badges from the following 12 categories: *Analytical*, *Applied*, *Bibliographic*, *Computational*, *Data Analysis*, *Design*, *Development*, *Experimental*, *Numerical*, *Research*, *Simulation*, *Theoretical*.
- **Supervisors** → Displayed as badges containing their full names.
- **Keywords** → Two keywords are visible by default, with additional ones shown when hovering over the *Others* badge.

Following the badges, the **description of the thesis** is displayed, limited to a maximum of six lines for readability. At the bottom of each card, there is a **button** that allows users to access the full details of the proposal.

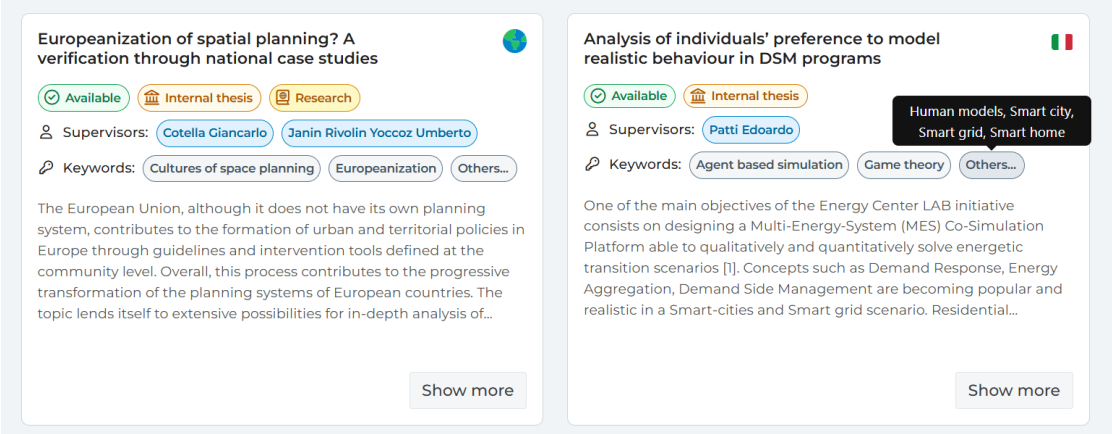


Figure 6.16: Thesis proposals cards

In the following pages, complete screenshots of both the light and dark versions of the thesis proposals section are provided, along with images of the thesis proposal detail page. This page follows the same structure as the external card but includes additional details such as external co-supervisors (if any), the full description, creation and expiration dates, and optional elements including notes, required skills, links, and attachments.

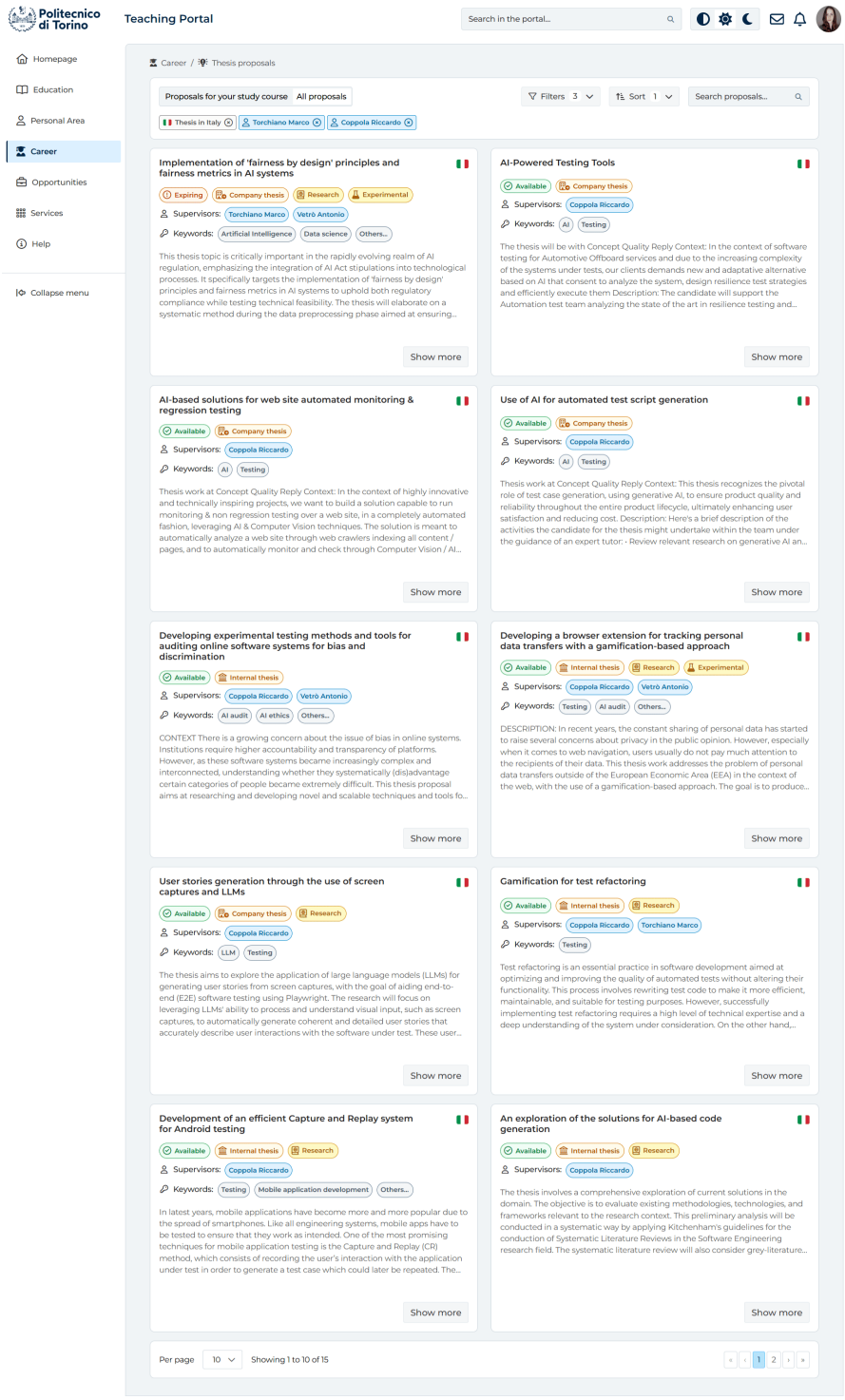


Figure 6.17: Thesis proposals interface - light version

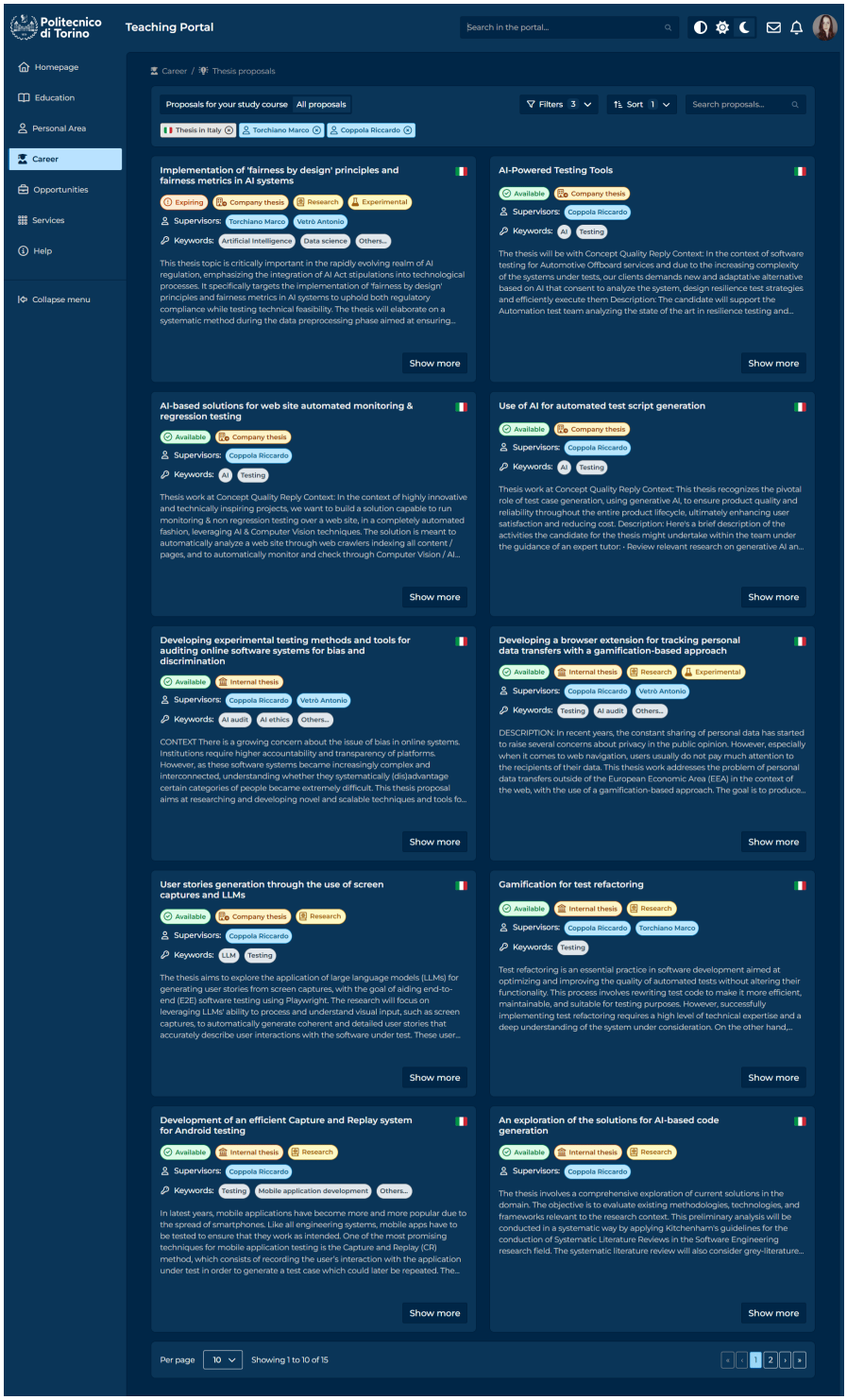


Figure 6.18: Thesis proposals interface - dark version

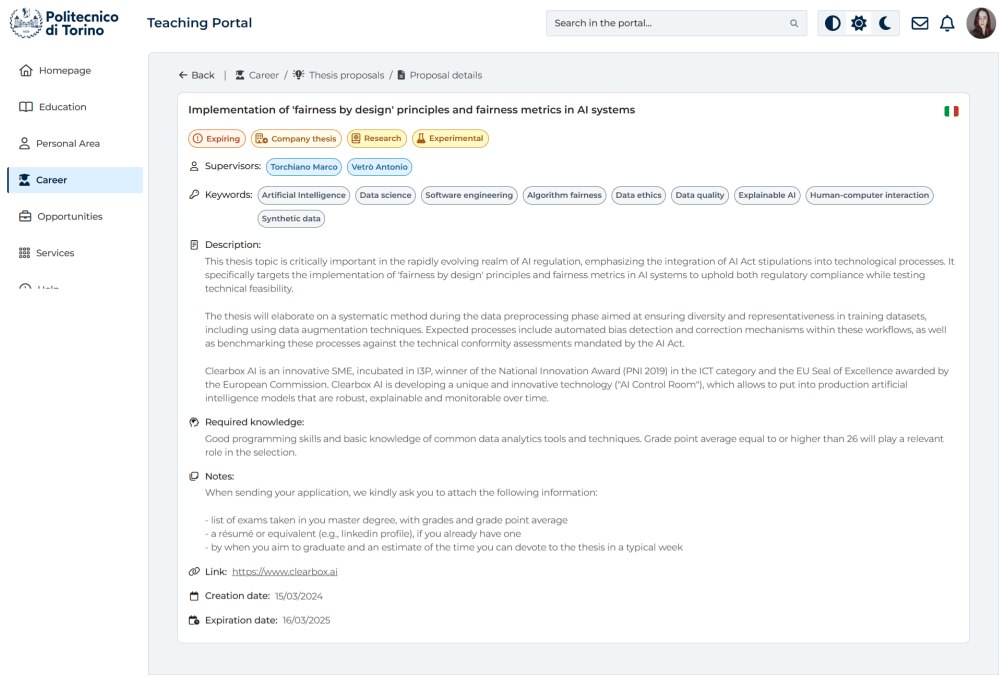


Figure 6.19: Thesis proposal details interface - light version

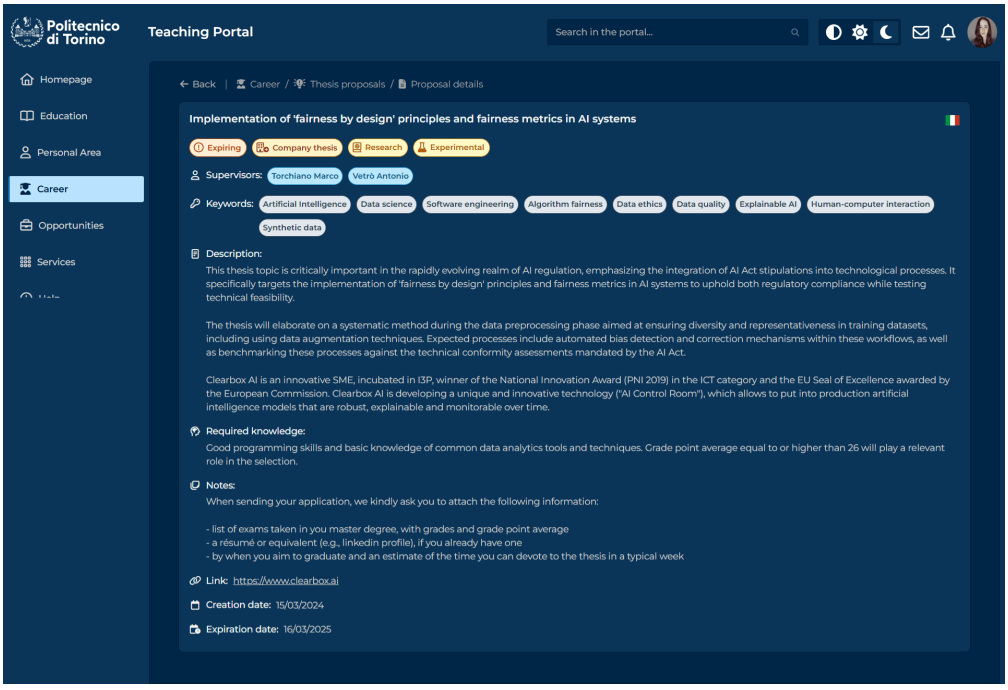


Figure 6.20: Thesis proposal details interface - dark version

Chapter 7

Verification and Validation

Software engineering relies on **verification and validation (V&V)** processes to ensure the quality, accuracy, and correctness of a system. These processes aim to answer two fundamental questions:

- **Verification:** *"Was the system built correctly?"* This step ensures that the implementation aligns with the defined requirements.
- **Validation:** *"Was the right system built?"* This step assesses whether the system meets user needs and expectations.

The key concepts for understanding the potentiality of V&V are:

- **Failure:** an event in which the software does not behave as expected, always resulting from one or more underlying faults.
- **Fault:** a feature of the system that causes the failure. A single fault may cause multiple failures or none at all.
- **Defect:** a general term that includes both faults and failures.

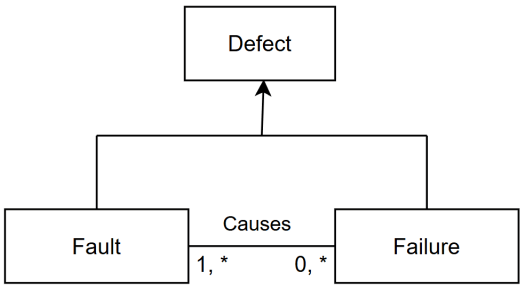


Figure 7.1: Relationship between fault, failure, and defect

While the best software would have no faults, a reliable one is software with no failures. Identifying and addressing defects early in development is essential, as fixing them becomes more costly as the project progresses. Consequently, the main objectives of V&V are:

1. Minimizing the number of defects introduced during development.
2. Maximizing the number of defects detected and removed.
3. Minimizing the time between defect introduction and resolution.

V&V techniques can be classified into:

1. **Static techniques** (e.g., inspections, source code analysis)
2. **Dynamic techniques** (e.g., testing at different levels: unit, integration, end-to-end, usability tests)

In this project, both static and dynamic testing techniques were applied. Static code analysis was performed using Sonar, allowing for early detection of code quality issues, security vulnerabilities, and maintainability concerns. In addition, various dynamic testing techniques were employed to evaluate the system's behaviour through automated and manual tests.

The following sections describe the testing strategy, methodologies, and results.

7.1 Verification

Verification ensures that the system is built correctly according to requirements and specifications. The testing strategy adopted in this project included several

types of tests, each with specific objectives:

1. **Static code analysis** → Detect code quality issues, security vulnerabilities, and maintainability concerns early.
2. **Unit Testing** → Verify that individual software components work as expected.
3. **Integration Testing** → Ensure proper interaction between components.
4. **End-to-End Testing** → Evaluate the system from an end-user perspective, simulating real scenarios.

The project adopted a combination of automated and manual testing, where automated tests provided efficiency and repeatability, while manual testing captured qualitative usability aspects.

7.1.1 Static code analysis

Sonar has been integrated into the development flow to improve code quality and maintainability, as well as to manage technical debt. Indeed, Sonar static code analysis is able to identify potential issues such as:

- Code smells and maintainability issues.
- Vulnerabilities and security risks.
- Duplicate code and complexity hotspots.

The adoption of SonarQube has ensured compliance with code writing standards, reduced technical debt, and enforced best practices.

This process has been fully automated within the CI/CD pipeline, ensuring continuous code improvement during development.

Specifically, for each Pull Request on GitHub, an automated pipeline was configured to run all tests (both back-end and front-end) and then start the analysis with Sonar.

In this way, in addition to performing code analysis, Sonar was also configured to receive test coverage data, ensuring that quality gates were always met before a PR was merged into the main branch.

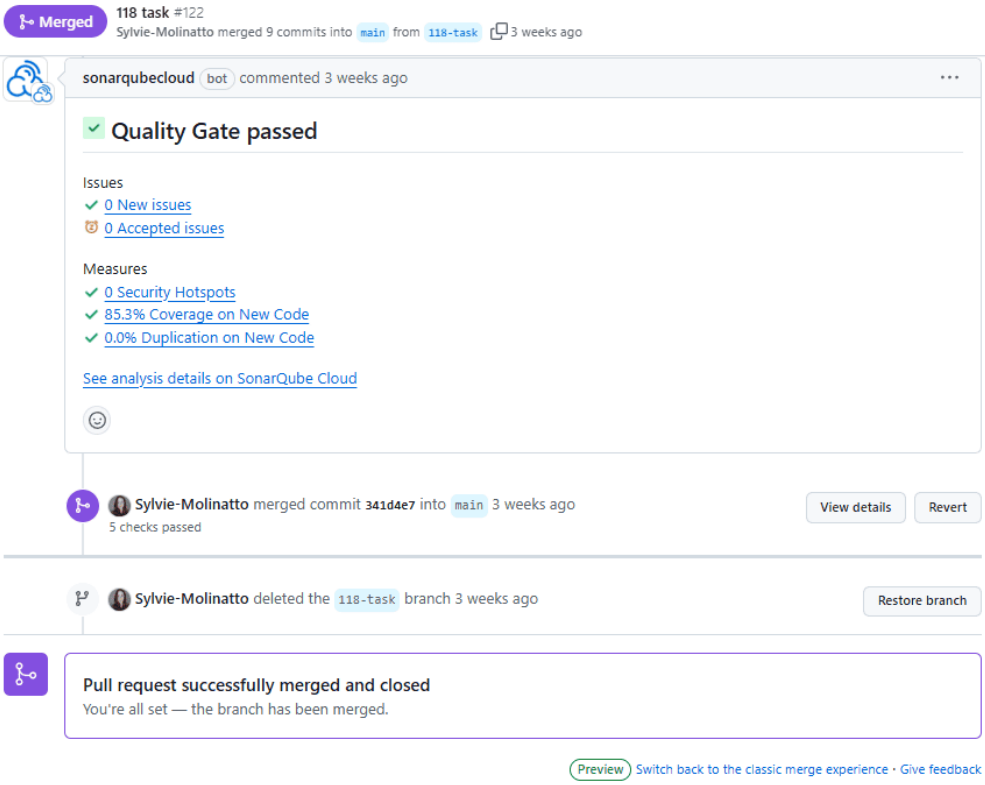
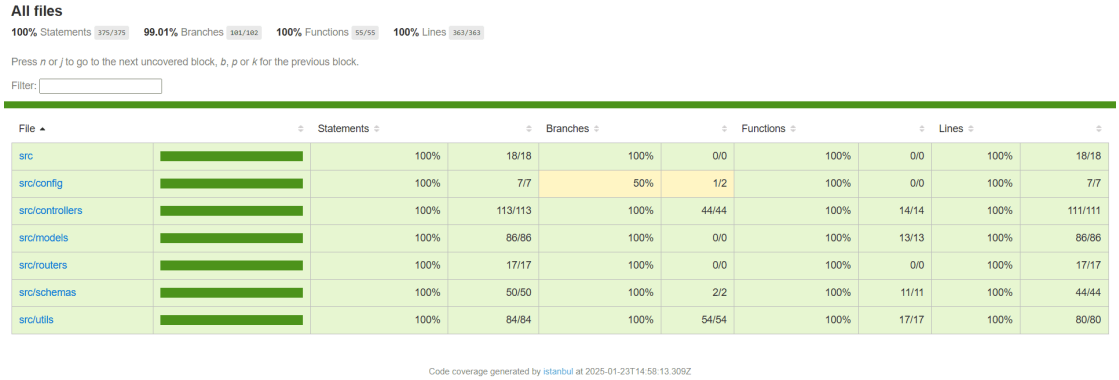


Figure 7.2: Sonar Quality Gates on a Github Pull Request

7.1.2 Back-end testing

The back-end of the application has been tested both with unit and integration tests.

The following image shows the code coverage achieved with both types of tests:



All files

100% Statements 375/375 **99.01%** Branches 101/102 **100%** Functions 55/55 **100%** Lines 363/363

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements		Branches		Functions		Lines	
src	100%	18/18	100%	0/0	100%	0/0	100%	18/18
src/config	100%	7/7	50%	1/2	100%	0/0	100%	7/7
src/controllers	100%	113/113	100%	44/44	100%	14/14	100%	111/111
src/models	100%	86/86	100%	0/0	100%	13/13	100%	86/86
src/routers	100%	17/17	100%	0/0	100%	0/0	100%	17/17
src/schemas	100%	50/50	100%	2/2	100%	11/11	100%	44/44
src/utils	100%	84/84	100%	54/54	100%	17/17	100%	80/80

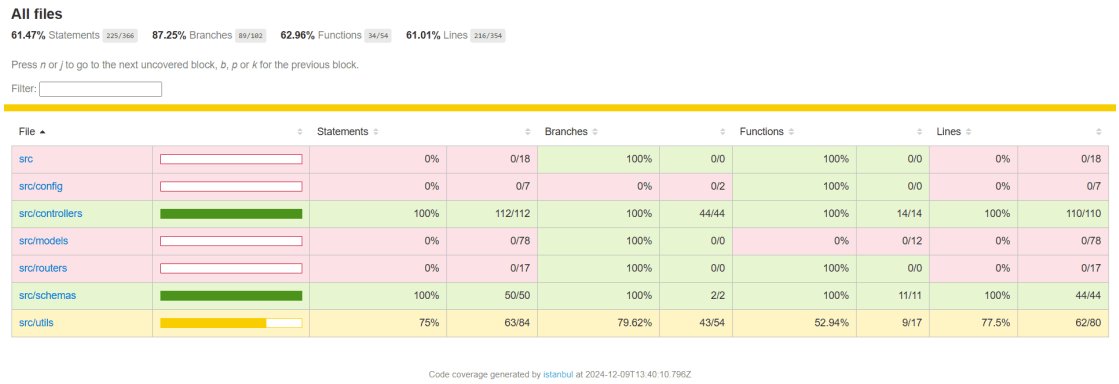
Code coverage generated by istanbul at 2025-01-23T14:58:13.309Z

Figure 7.3: Overall back-end tests code coverage

Unit testing

Unit tests represent the first level of automated testing and are intended to verify that individual units of code work properly in isolation. For this project, they were implemented using **Jest**, a framework widely used to test JavaScript and Node.js applications.

The following image shows the code coverage achieved through unit tests:



All files

61.47% Statements 225/366 **87.25%** Branches 89/102 **62.96%** Functions 34/54 **61.01%** Lines 216/354

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements		Branches		Functions		Lines	
src	0%	0/18	100%	0/0	100%	0/0	0%	0/18
src/config	0%	0/7	0%	0/2	100%	0/0	0%	0/7
src/controllers	100%	112/112	100%	44/44	100%	14/14	100%	110/110
src/models	0%	0/78	100%	0/0	0%	0/12	0%	0/78
src/routers	0%	0/17	100%	0/0	100%	0/0	0%	0/17
src/schemas	100%	50/50	100%	2/2	100%	11/11	100%	44/44
src/utils	75%	63/84	79.62%	43/54	52.94%	9/17	77.5%	62/80

Code coverage generated by istanbul at 2024-12-09T13:40:10.796Z

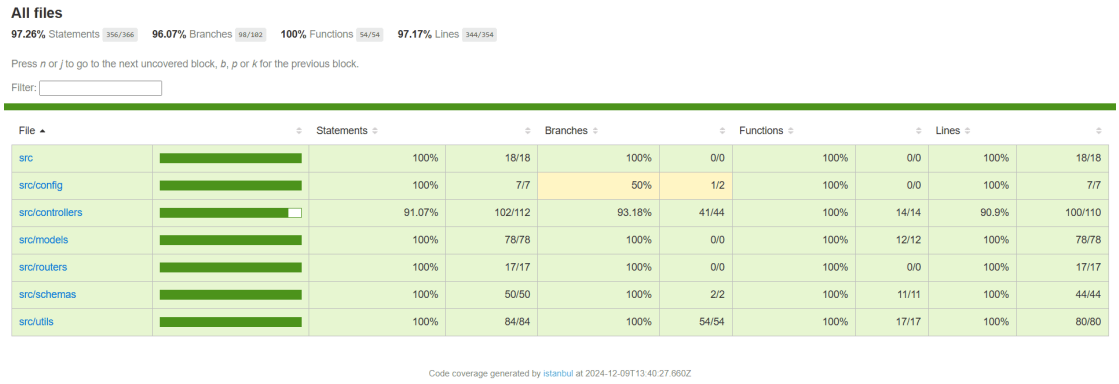
Figure 7.4: Unit tests code coverage

Integration testing

After verifying the proper functioning of the individual components, the next step was to test their interaction using integration tests. These tests were implemented using **Jest** and **Supertest**, a library to test HTTP APIs in the Node.js environment.

To avoid compromising the actual data, a separate test database was configured, which allowed the tests to run in a controlled environment.

The following image shows the code coverage achieved through integration tests:



All files

97.26% Statements 356/366 **96.07%** Branches 98/102 **100%** Functions 54/54 **97.17%** Lines 344/354

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements		Branches		Functions		Lines	
src	100%	18/18	100%	0/0	100%	0/0	100%	18/18
src/config	100%	7/7	50%	1/2	100%	0/0	100%	7/7
src/controllers	91.07%	102/112	93.18%	41/44	100%	14/14	90.9%	100/110
src/models	100%	78/78	100%	0/0	100%	12/12	100%	78/78
src/routers	100%	17/17	100%	0/0	100%	0/0	100%	17/17
src/schemas	100%	50/50	100%	2/2	100%	11/11	100%	44/44
src/utils	100%	84/84	100%	54/54	100%	17/17	100%	80/80

Code coverage generated by istanbul at 2024-12-09T13:40:27.660Z

Figure 7.5: Integration tests code coverage

7.1.3 Front-end testing

Front-end testing ensures a stable user experience and mitigates unexpected issues caused by code modifications. Even minor code changes can have unforeseen impacts, which makes front-end testing essential.

End-to-End testing

End-to-end (E2E) tests simulate user interactions to verify system functionality.

Cypress was selected as the E2E testing framework due to its ability to automate tests across multiple browsers and devices.

The test suite has been implemented to cover the most frequently used components, trying to simulate the behaviour of a potential end user.

Thanks to these tests, it was possible to notice that some interface components were hiding others, compromising the usability (for example, the floating button to return to the top of the page was overlaid with pagination buttons, in some cases).

The following image shows the code coverage achieved through E2E tests:

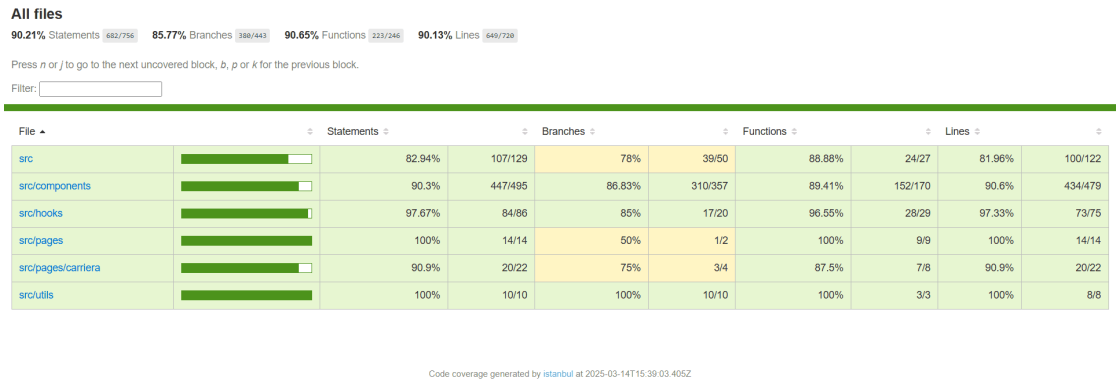


Figure 7.6: End-to-end tests code coverage

7.2 Validation

As briefly explained before, validation ensures that the right system has been built by assessing whether it meets user needs and expectations.

In the context of this project, the validation step was made up of usability testing and post-test questionnaires with students.

7.2.1 Usability testing

Usability tests were conducted to identify interface design problems and correct them as soon as possible. This testing phase allowed to learn more about students' behaviour in researching thesis proposals.

A total of 10 students from different master's degree programs at Politecnico di Torino were selected, and the tests were conducted in person. Each testing session required the presence of at least two members of the development team: one played the role of facilitator (i.e., the one who guides the participant through the test process), while the other played the role of observer (the one who focuses on

observing the participants, taking notes, without interacting with them), to ensure that the tests were conducted properly and to collect detailed observations on user behaviour.

Below are graphs showing the distribution of study courses and the gender composition of participants:

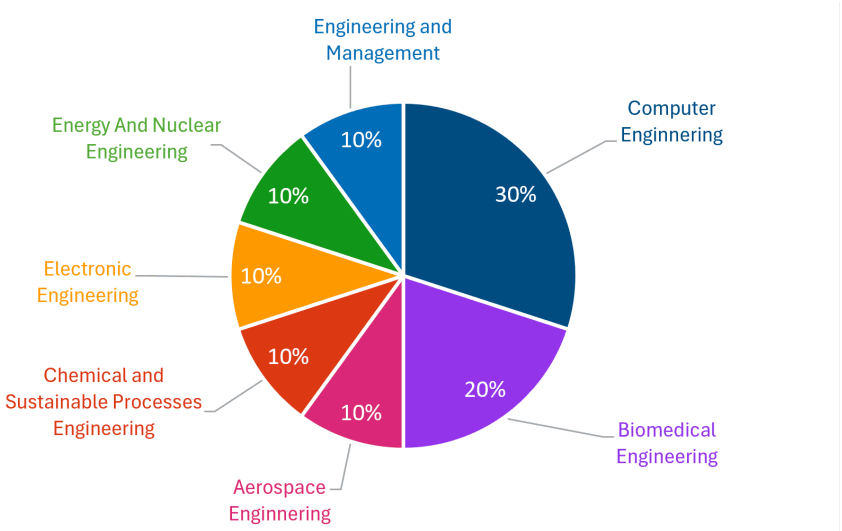


Figure 7.7: Course of study breakdown

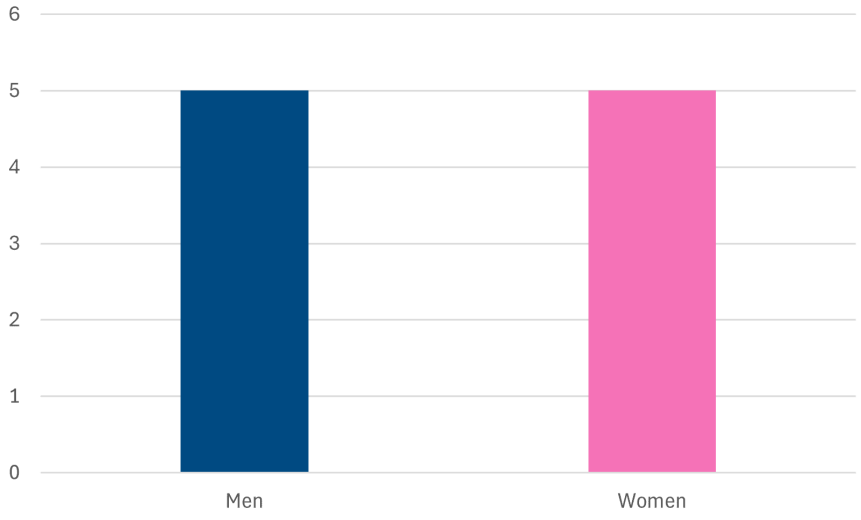


Figure 7.8: Gender breakdown

Students who participated in this phase had to complete six tasks regarding the main features of the new interface so that qualitative feedback could be collected on its usability.

Below is a description of each task, along with the objective and associated evaluation metrics. For each task, the methodology used was thinking aloud so that the tester's doubts and/or hesitations could be captured.

Tasks

1. **Navigation and general understanding of the interface**

   Goal: Assess whether students can easily navigate the thesis proposal section and interpret displayed information.

   Task:

   - Considering you are already logged in, locate the thesis proposal section without receiving explicit instructions.
   - Explore the page and describe what you understand by looking at the interface: Which information is immediately clear? Which are unintuitive? If you wanted to find a thesis of interest to you, which action would you take first?
   - Change the application theme from 'auto' to 'dark'.
   - Select a thesis proposal and open its details. Identify the following elements: title, supervisors, types of thesis, location, mode, and deadline.
   - Go back to the thesis proposal list.

   Evaluation metrics: Time taken to find the thesis proposal section, clarity of the information presented, difficulties encountered in understanding the interface, locating information, and/or changing the theme.

2. **Pagination and proposals display**

   Goal: Assess the usability of pagination and how many proposals per page are optimal.

   Task:

   - Open the list of all thesis proposals.
   - Scroll through and check how many proposals are displayed on one page.
   - Navigate to the next page and back to the previous one.

- Change the number of displayed proposals per page and select your preferred one.

Evaluation metrics: Percentage of users who correctly navigate between pages, ease of finding and using pagination controls, user preference for number of proposals per page.

3. **Thesis proposals' filtering**

   Goal: Test how easily users can apply and combine multiple filters.

   Task:

   - Go to the list of all thesis proposals.
   - Filter thesis proposals that are both internal, supervised by the teacher 'Edoardo Patti', and have the keyword 'machine learning'.
   - Remove only the filter on the supervisor and filter only theses that have the type 'Simulation'.
   - Clear all filters and return to the full list.

   Evaluation metrics: Percentage of users who successfully apply multiple filters, ease of combining filters, difficulties in clearing filters.

4. **Understanding active filters**

   Goal: Evaluate whether users understand which filters are currently active and how to modify them.

   Task:

   - Some filters have already been applied: location = 'thesis abroad' and keyword = 'Cultures of space planning'.
   - Identify which filters are currently active.
   - Change the keyword filter to 'Modern architecture history'.
   - Remove only the thesis location filter and check if the correct proposals remain.
   - Clear all filters and return to the full list.

   Evaluation metrics: Percentage of users who correctly identify active filters, ability to modify and remove filters without confusion.

5. **Handling No Results in Filtering**

   Goal: Test user reactions when no results are found due to applied filters.

   Task:

   - Apply a filter for company theses with type = 'numerical'.

- Observe what happens when no results match.
- Find a way to reset the filters and get back to the full list.

Evaluation metrics: Time taken to realize that no results are available, user frustration level, effectiveness of empty-state messages.

6. **Thesis proposals' sorting**

   Goal: Test the correctness and usability of sorting

   Task:

   - Go to the list of proposals targeted to your course of study.
   - Sort proposals by expiration date to see the ones that will expire soon.
   - Change the sorting order from ascending to descending.
   - Reset sorting.

   Evaluation metrics: Percentage of users who successfully apply sorting, change the sorting order, and reset sorting.

7.2.2 Post-test questionnaire

After completing all tasks, participants filled out a **System Usability Scale (SUS) questionnaire** to provide an overall assessment of their experience.

The System Usability Scale, developed by John Brooke [8] in 1986, is a rapid and effective way to evaluate the usability of an application by measuring the perceived usability of the system, which is composed of three main aspects:

1. **Efficacy** → Are users able to achieve their objectives successfully?
2. **Efficiency** → how much effort and resources are being used to accomplish these objectives?
3. **Satisfaction** → Was the experience satisfying?

In practice, the SUS consists of 10 Likert-scale questions, each with five response options ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

Consequently, the output, after applying specific mathematical calculations, is a grade from 0 to 100, but it should not be interpreted as a percentage but rather as a standardized usability measure. The average SUS score is 68, meaning that a lower value may indicate usability issues to be found and solved.

The SUS questionnaire is available at Appendix A.1

7.2.3 Results and improvements

This subsection presents the results and observations from the usability tests, broken down by task and related steps. In addition, the score obtained through the SUS questionnaires is reported.

Task 1: Navigation and general understanding of the interface

Step 1: Finding the thesis proposals' section	
Average time	24.6 s
Success rate	60% (calculated considering as success the case where the section was found with no more than two attempts)
Common issue	Many testers went to the 'Opportunities' section first.

Table 7.1: Usability test results for Step 1 of Task 1

Step 2: Locating unintuitive information	
Issue identified	**Suggested improvement**
Sorting by 'topic' is unclear	Change the label from 'topic' to 'title'.
The Italian flag badge is interpreted as the thesis language instead of location	To avoid confusion, replace it with a different icon that clearly indicates the thesis location, as the flag icon is already used elsewhere in the teaching portal to represent language.
'Available' and 'Expiring' badges cause confusion	Show the due date in the overview for clarity.

Table 7.2: Usability test results for Step 2 of Task 1

Step 3: Spontaneous actions to search for a thesis	
User strategy	**Number of participants**
Search via the search bar	5
Apply some filter in the appropriate panel	4
Explore all options and possibly mark his favourites to consult them later	1

Table 7.3: Usability test results for Step 3 of Task 1

Step 4: Changing the application theme	
Average time	3.1 s
Success rate	100% (all participants were able to apply the change)
Issue identified	80% could not understand the automatic theme icon.
Suggested improvement	Provide only light/dark theme selection and use system theme as default.

Table 7.4: Usability test results for Step 4 of Task 1

Task 2: Pagination and proposals display

- **Step 1: Navigating pages** - All participants were able to navigate the pages correctly.
- **Step 2: Preferred number of proposals per page** - Preferences varied, with most testers favouring fewer than 50 proposals per page. However, keeping the option customizable ensures flexibility for different user habits.

Task 3: Thesis proposals' filtering

Step 1: Applying filters (*environment = 'Internal thesis'*, *supervisor = 'Edoardo Patti'* and *keyword = 'machine learning'*)	
Average time	44.5 s
Success rate	90%
Notes	- One participant incorrectly applied the 'Italy thesis' filter, assuming an internal thesis could not be abroad. - Some hesitations were observed, especially regarding the filters 'Location' and 'Environment', which led to minor errors and corrections. - Some participants tried to use the keyboard to apply some filters or scroll through the lists of items they could filter on, but they fell into error as the use of the keyboard was not managed. - Some participants did not immediately realize that they could type to search for supervisors, keywords, or types.

Table 7.5: Usability test results for Step 1 of Task 3

Step 2: Removing the filter on the supervisor and applying the filter for *'Simulative'* type	
Average time	16.8 s
Success rate	100%
Notes	One problem, which emerged with only one tester, was forgetting to apply the new filter after selection, which required reopening the filter panel (an action performed independently as the participant realized that the new filter was not applied).

Table 7.6: Usability test results for Step 2 of Task 3

Step 3: Resetting all filters	
Average time	3.9 s
Success rate	100% (all participants were able to reset filters correctly, even though they used different methods)
Notes	Time to complete the subtask varies significantly depending on the method used: • Most participants clicked directly on the summary badges (fast method). • Two testers used the global reset button within the filter panel (fast method). • The others clicked the badges within the filter panel or the reset button for each filter type and then applied the changes (slower method).

Table 7.7: Usability test results for Step 3 of Task 3

Task 4: Understanding active filters

Step	Average time	Success rate	Notes
1. Identifying active filters (*location = 'Abroad thesis'*, *keyword = 'Cultures of space planning'*)	13.1 s	100%	Although some users took a longer time, all participants were able to recognize the active filters.
2. Changing the keyword filter to *'Modern architecture history'*	12.3 s	100%	-
3. Removing the thesis location filter	2.67 s	100%	-
4. Verifying visually the presence of the keyword for which proposals were filtered (*'Modern Architecture history'*)	N/A	100%	Participants correctly navigated to the "Others" badge to verify that the thesis proposals contained the required keyword

Table 7.8: Usability test results for Task 4

Task 5: Handling no results in filtering

Step	Average time	Success rate	Notes
1. Applying filters (*environment = 'company thesis'*, *type = 'numerical'*)	N/A	100%	All participants applied the filters correctly.
2. Recognizing the 'no result' situation	2 s	100%	All participants understood that the lack of results was due to the applied filters.
3. Resetting all filters	<1.3 s	100%	All participants reset the filters using the appropriate button.

Table 7.9: Usability test results for Task 5

Task 6: Thesis proposals' sorting

Step	Average time	Success rate	Notes
1. Sorting proposals by expiration date	3.9 s	100%	All participants were able to apply the sorting correctly.
2. Changing the sorting order from ascending to descending	12.4 s	50%	Some participants didn't immediately recognize the sorting icon's function and clicked multiple times or lingered without clicking due to lack of feedback.
3. Resetting the sorting	3.1 s	100%	-

Table 7.10: Usability test results for Task 6

Users feedback

During the usability tests, both implicit and explicit feedback were collected from the participants. These insights highlight improvement areas and potential refinements for the system. The main suggestions provided by the participants are listed below:

- The theme selection component should only include light and dark modes, with the system default assigned by default.
- The position of the thesis proposals section is not so intuitive for students, as they expect proposals to be in the same section where there are job offers, which is 'Opportunities'
- The search bar should search across all fields of a thesis proposal.
- In the 'All proposals' section, it should be indicated somewhere the course of study to which the proposal is addressed, and maybe it should be possible to filter proposals by course of study as well.
- The expiration date of thesis proposals should be displayed in the overview to facilitate control, especially when sorting proposals.
- Since there is only one sorting level, the sorting mechanism should apply changes immediately upon selection, without requiring an explicit 'Apply' button.
- A summary badge for sorting would be useful to allow users to reset sorting quickly.
- Teachers should be listed alphabetically by last name instead of first name.
- The application should support full keyboard navigation, including tab-based focus transitions, keyboard shortcuts for key actions, and accessible controls for all interactive elements.
- The 'no results' page icon is not intuitive; someone perceived it as a closing action rather than an indication of an empty state (even if the message is correctly understood).
- One tester expected that clicking on teacher badges would allow direct communication with them.

System Usability Scale (SUS) evaluation

Thanks to the questionnaires compiled by the 10 participants in the usability test, it was possible to calculate the SUS score. As mentioned in section 7.2.2, this score is obtained by performing some predefined calculations on the answers, rated from 1 to 5, given by users.

The SUS score is calculated using the following formula:

$$\text{SUS Score} = 2.5 \times \left(\sum_{i=1}^{10} (X_i - \text{adjustment}) \right) \tag{7.1}$$

where:

- For odd-numbered questions, the adjustment is 1 (i.e., $X_i - 1$).
- For even-numbered questions, the adjustment is 5 (i.e., $5 - X_i$).
- The total is then multiplied by 2.5 to obtain a 0-100 score.

The average SUS score across all participants is computed as:

$$\text{SUS}_{\text{avg}} = \frac{\sum_{i=1}^{n} \text{SUS Score}_i}{n} \tag{7.2}$$

where n is the total number of participants.

For this project, the **computed SUS score** is **90.5** , which is considered excellent based on the interpretation scale reported in Appendix A.2.

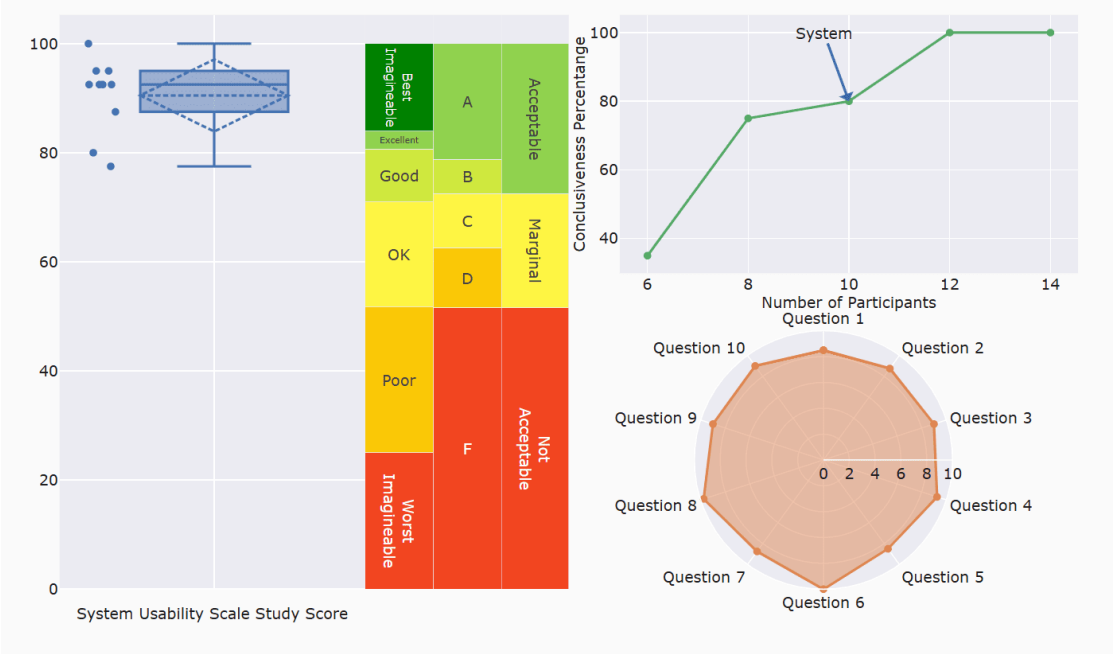


Figure 7.9: System Usability Scale (SUS) evaluation results

Figure 7.9 illustrates the results of the SUS evaluation using multiple visualizations:

- A **box plot** representing the distribution of SUS scores across participants.
- A **grading scale** that helps interpret the SUS score in terms of usability categories.
- A **line graph** showing the conclusiveness percentage based on the number of participants.
- A **radar chart** depicting the responses to individual SUS questions.

A score of 90.5 indicates a highly usable and well-received system, confirming that participants found the interface intuitive and efficient. This result aligns with the high success rates observed during usability testing and the generally positive verbal feedback provided by testers.

Chapter 8

Conclusions

8.1 Results summary

This project started with the objective of analyzing, designing, and developing a new thesis management procedure integrated into the Teaching Portal of the Politecnico di Torino. The primary goals were to improve usability, enhance user experience, and ensure a more intuitive management of the thesis process.

Following the initial phase of analysis and identification of student needs, it emerged that priority should be given to the management of thesis proposals, because this initial phase of the thesis life cycle has been found to be characterized by many critical issues for users.

Therefore, a user-centered design process was then adopted, using agile methodologies to iteratively develop and refine the system based on continuous feedback from the designers, the supervisor, and the students themselves.

As a result, a React web application has been created, parallel to the new Teaching Portal, containing a new interface for displaying thesis proposals. Additionally, RESTful APIs were defined and implemented following the OpenAPI 3.0 standard to ensure interoperability and scalability of the system. A dedicated back-end was also developed to support API testing and validation, containing normalized data to enable more structured thesis proposals management.

At the end of development, the interface was tested and validated through specific usability tests, conducted directly with students. The results of the validation phase

have shown a better usability of the information content and an interface more aligned to the new portal, bringing overall significant improvements in usability and user experience.

8.2 Limitations and future work

Despite the positive results, some potential areas for improvement have emerged. These include the enhancement of accessibility through the use of keyboard commands, a review of the positioning of the proposal section (potentially placing it in the "Opportunities" section), and the refinement of the search bar.

In addition, some icons could be made more intuitive, such as those related to the selection component of the application theme, the badge identifying the theses to be carried out in Italy, and the icon that appears when no proposals are corresponding to the applied filters.

Future functionalities could include the introduction of the option to select preferred proposals and the addition of a button to contact the teacher directly for further information.

A more detailed list of enhancements and proposed features, gathered from students, is available in the 'Users feedback' section. Some of these improvements have already been implemented, while others can be evaluated by colleagues who will continue the work.

In the future, it is planned to design and implement interfaces for the application of a thesis proposal, for the management of your thesis, and for the completion of the steps of registration for the final exam, including confirmation of the thesis conclusion, registration for the examination, and final upload of the document.

The OpenAPI specification provides a solid foundation for the expansion of these features, ensuring consistent documentation and facilitating future integration between front-end and back-end systems.

As the project is released in open source mode, there is ample opportunity to continue the work and progressively integrate the new interfaces into the Teaching Portal.

Appendix A

Appendix

A.1 System Usability Scale (SUS) questionnaire

For each statement, please check the box that best represents your immediate response.

Question	**Rating**				
	Strongly Disagree				**Strongly Agree**
1. I think that I would like to use this system frequently.	1	2	3	4	5
2. I found the system unnecessarily complex.	1	2	3	4	5
3. I thought the system was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to use this system.	1	2	3	4	5
5. I found the various functions in this system well-integrated.	1	2	3	4	5

Question	Rating				
	Strongly Disagree				**Strongly Agree**
6. I thought there was too much inconsistency in this system.	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly.	1	2	3	4	5
8. I found the system very cumbersome to use.	1	2	3	4	5
9. I felt very confident using the system.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system.	1	2	3	4	5

Table A.1: System Usability Scale (SUS) questionnaire

A.2 System Usability Scale (SUS) interpretation

SUS score range	Usability interpretation
0-50	Poor
51-70	Ok (Marginal usability)
71-85	Good
85-100	Excellent

Table A.2: System Usability Scale (SUS) interpretation

Bibliography

[1] Gianmarco De Paolis and Giacomo Maggio. «Riprogettare l'esperienza utente (UX) del Portale della Didattica». Master thesis. Politecnico di Torino, Dec. 2023. URL: https://webthesis.biblio.polito.it/30106/ (cit. on pp. 4, 48).

[2] Interaction Design Foundation - IxDF. *What is User Centered Design (UCD)?* June 2016. URL: https://www.interaction-design.org/literature/topics/user-centered-design (cit. on p. 20).

[3] Bill Wake. *INVEST in Good Stories, and SMART Tasks*. 2003. URL: https://xp123.com/invest-in-good-stories-and-smart-tasks/ (cit. on p. 46).

[4] Roy Thomas Fielding. «Architectural Styles and the Design of Network-based Software Architectures». PhD thesis. University of California, Irvine, 2000. URL: https://ics.uci.edu/~fielding/pubs/dissertation/top.htm (cit. on p. 55).

[5] OpenAPI Initiative. *OpenAPI Specification v3.0.0*. 2017. URL: https://spec.openapis.org/oas/v3.0.0.html (cit. on p. 56).

[6] Swagger. *What Is OpenAPI?* URL: https://swagger.io/docs/specification/v3_0/about/ (cit. on p. 56).

[7] *Manifesto for Agile Software Development*. 2001. URL: https://agilemanifesto.org/iso/it/manifesto.html (cit. on p. 69).

[8] John Brooke. *SUS – a quick and dirty usability scale*. 1996. URL: https://www.researchgate.net/publication/319394819_SUS_--_a_quick_and_dirty_usability_scale (cit. on p. 103).