NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

LAURA GASPAR MARTINS

Master in Electrical and Computer Engineering

PORTAL FOR SUBMISSION AND REVIEW OF PROPOSALS FOR OPEN CALLS

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
July, 2022

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

PORTAL FOR SUBMISSION AND REVIEW OF PROPOSALS FOR OPEN CALLS

LAURA GASPAR MARTINS

Master in Electrical and Computer Engineering

Adviser: Ricardo Jardim Gonlçalves
Full Professor, NOVA University Lisbon

Co-adviser: José Alexandre Pires Ferreira
Senior Researcher, UNINOVA

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
July, 2022

Portal for submission and review of proposals for Open Calls

À minha "mãezinha" Catarina

Acknowledgements

I would like to thank my adviser, José Ferreira, for the guidance and patient during the whole process of creating this thesis. For my adviser, Ricardo Gonçalves, thank you for your work and for making possible for me to create this thesis in a area that interests me so much.

To my university, NOVA School of Science & technology, thank you for the knowledge, thank you for the opportunities and thank you for the community created here.

For Catarina, when this thesis felt impossible to do, you made things lighter and grounded me and I'm so thankful for your friendship during this time.

For my friends, João Lameiras, Pedro, Miguel, João Anastácio and Alexandre, you made this five years my best five years, and I will always cherish every moment we spent together studying, partying or just hanging out. A special thanks to Miguel and Alexandre for your willingness to help me at any day or any hour with this project, João Lameiras for the best 6 months and endless dad jokes and Pedro e João Anastácio for the laughs that made this year fifty times easier.

For last but not least, thank you to my family. Mom, dad I'm so grateful your support and love. Alice and Marta thank you for being who you are and inspiring me so much. To Inês, Francisco and Filipa, your sweetness means so much to me.

"See you later alligator." (Bill Haley)

Abstract

In this day and age there is a discrepancy in the knowledge accessible of data structures, analysis and optimization and the way proposals for calls are handled. It is still a very slow, inefficient and unorganized process for candidates and especially for evaluators of proposals for calls, where most tasks are manual and repetitive and everyone involved wastes a lot of time.

There is a need to create an online place where candidates can explore different open calls (ideally, it would propose calls more likely fitted for the candidate), check their requirements, make a proposal and be able to see the stage and status of the call. In this place creators of calls can have different templates to use as well as a listing of available evaluators that are likely fitted to the position. On the same platform, creators can hire the evaluators and see their progress. Besides that it would be benefit if the evaluator would have some assistance with verifying some requirements.

This dissertation aims to create a data structure to organize and hold a platform where users can create calls, candidates can submit proposals to said calls, and users can apply to become reviewers of calls. The administrator should also be able to choose the most fitted evaluators directly from the platform as well as handles an initial part of the evaluation. This is only possible if the information about every call, candidate and evaluator are connected in the same database. Taking that in consideration it is also a goal of this thesis to create a complete proposal portal, in a efficient, easy, user-friendly way, where all components of call proposals can be made in the same place.

Keywords: database, ontology, portal, calls, proposals, data structure, data analysis

Resumo

Nos dias de hoje existe uma discrepância no conhecimento acessível de estruturas de dados, análise e otimização das mesmas, e na forma como as propostas de concursos são realizados. Ainda é um processo muito lento, ineficiente e desorganizado para candidatos e principalmente para avaliadores de propostas de editais, onde a maioria das tarefas são manuais e repetitivas, onde todos os envolvidos perdem muito tempo.

É necessário criar um local online onde os candidatos possam explorar diferentes concursos públicos (o ideal seria o portal propor concursos mais adequados ao candidato), verificar os seus requisitos, fazer uma proposta e poder ver a fase e o estado do concurso . Neste local, os criadores de concursos podem ter diferentes templates para usar, bem como uma lista de avaliadores disponíveis que provavelmente se encaixam na posição. Na mesma plataforma, os criadores podem contratar os avaliadores e acompanhar seu progresso. Além disso, seria benéfico se o avaliador tivesse alguma ajuda na verificação de alguns requisitos.

Esta dissertação tem como objetivo criar uma base de dados para organizar e estruturar uma plataforma onde os utilizadores podem criar concursos, os candidatos podem apresentar propostas aos referidos concursos e os utilizadores podem candidatar-se a revisores de concursos. O administrador também deve poder escolher os avaliadores mais adequados diretamente da plataforma, bem como cuidar de uma parte inicial da avaliação. Isso só é possível se as informações de cada chamada, candidato e avaliador estiverem conectadas no mesmo banco de dados. Levando isso em consideração, é também objetivo desta tese criar um portal de propostas completo, de forma eficiente, fácil e user-friendly, onde todas as tarefas necessárias para criação de concursos e propostas possam ser feitos no mesmo local.

Palavras-chave: base de dados, ontologia, concursos, propostas, estrutura de dados, análise de dados

Contents

List of Figures

Acronyms

DAML DARPA Agent Markup Language 19
DARPA Defense Advanced Research Projects Agency 19
DL Description Logic 20

EC European Comission 7, 8

OIL Ontology Interchange Language 19
OWL Web Ontology Language 19

RDF Resource Description Framework 19

SEDIA Single Electronic Data Interchange Area 7

URI Uniform Resource Identifier 19

W3C World Wide Web Consortium 19

XML Extensible Markup Language 19

1

Introduction

The launch of calls for proposals and the submission of these has been a concept around for decades. There have always been the need to materialise the desire of receiving ideas of projects or innovations and consequently the need for people or entities to make a proposal whether that would be for an entity, company or even an organization. This process, in the beginning, used to be through conversations or meetings. To spread new calls it could be either through word to word or posters but this communication would always take a long time, and would not be very efficient. [1]

Later, when newspapers started to appear, it used to be a tool to spread this kind of contests. The entity which desired to create a call would pay for an advertisement in the newspaper, with a description of the purpose and goal of the call, usually with an address for the candidates to send their candidature through a letter. This process, as it is easy to imagine, was expensive, complicated and especially very time consuming. Another disadvantage for the candidates was that they would never be able to see the status or stage of the call and the communication in case of any questions would be extremely hard. For the reviewers, they would have to verify all the requirements for the proposal individually and manually which is also very time consuming.
After the digital revolution there was a transition from newspapers to websites and letters to emails. Both of these alterations were helpful in creating a more accessible process although it was still quite slow and the problem of the unknowing of status maintained as well as the validating individually each proposal. [2]

Although years have passed and technology has increased immensely, the world of submission for calls and proposals has been quite stagnant. Most calls still navigate through emails and all the requirements for the proposal have to be verified by a human.[3]

Nowadays, computers are used mostly to process information and data. [4] For this reason, these machine deal with massive quantities of information that, until recently, it was almost impossible to organize because it wasn't know how and given the amount it was becoming a problem. With this problem in mind, ontologies emerged (in a computer science manner). Ontologies are, in a simplified way, grouping sets of concepts and

stating the relationships between them in a language that machines can compute.

The goal of this master thesis is to create and validate a platform where it is possible to structure all the data from calls and proposals submitted and use that information in order to create an easier process for the administrator to be updated and make decisions about the reviewers and states of the call.

The platform, which is a Web Application, is pretended to have a ontology as a base, where is organized all the information about the candidates, reviewers, calls and proposals an the relationship between this data so that it can be used to connect reviewers and calls for the reviewers to evaluate.

Besides the referred before, this platform should be a user-friendly portal, where both users and reviewers can log in and submit their calls and proposals and apply for different contests and calls. This website must contain all the stages and sections needed for the communication between the users, reviewers and administrators. Thus, in the users point of view, the website should contain all the different open calls available, the information about them, the requirements needed to make a proposal and the forms for the candidature as well as updates from the calls the used made a proposal. In the reviewers point of view, first of all, the website must contain the information and process on how to became a reviewer. It should also have the calls proposed for the reviewer as well as a section where the reviewer can validate the proposals directly in the website. It is also proposed a new feature where the website can do some automatic validation. This can look like verifying the minimum and maximum size of documents and making sure that every participant meets the needed criteria. It is also important and is an objective to create a safe website where the user's information, given in the registration phase, is secure and stored in a private way.

1.1 Motivation

There is a lack of user-friendly portals online nowadays, with the main goal of submitting and evaluating proposals for calls, so there was a need to create one where both submission people and evaluators can easily access their needs and create an environment where the users save time and consequently be more productive. Besides that, with the evolution of technology, semantic web and in specific Ontologies, there is a big opportunity to evolve the submission portal process, making them more adaptable, helpful and automatic. Another main motivation for this database was to enable a creation of posterior optimizations, for instance algorithms that use the data saved on this database.

This dissertation was inspired by the necessity to create a smart platform for DIH4CPS, where is easily accessible a portal that makes the contact between administrators, reviewers and candidates fast, direct and not time consuming, as well as creating an effortless experience to each individual. With this system it is possible to track all the data from the proposals and creating a secure and robust backup for the creator of calls.

1.2 Research Questions

After understanding the background and motivation to the execution of this thesis, it is possible to raise the fundamental questions that must be answered in this dissertation:

> "Is it possible to represent the data collected in an open call into categories, properties and relationships between different concepts and entities to guide and simplify users tasks and decisions?"

In this chapter it is going to be described the plan for the execution of this thesis. In the first section, its presented the scientific strategy and methodology, where it analyses all the stages. The second section is where the hypothesis is proposed. For last in the third section it is planned the time schedule for the stages necessary to complete the thesis.

1.3 Work Methodology

In order to have a cohesive and successful thesis it is necessary to use the scientific method due to being impossible for researchers to remain impartial and not created an outcome that its preferred by them but not the most useful. So this scientific method prevents this from happening. [5]

The plan proposed next was based in the scientific method as well as in the guidelines given by the advisers of this thesis: [6] [5]

1. Problem
2. Context Observations
3. Formulate Hypothesis
4. Design Experiment
5. Implementation and test Hypothesis
6. Analyse Results
7. Thesis Writing

The first step, the problem, is to find and research the field where there is a problem or a need missing. Although the research problem must be clear, it should not be a declarative statement. For last, the study should be attainable and be able to be verified or denied. Some examples of this problem could look like not being user-friendly, not being as automated as possible (which wastes employers money and time).[6] This phase, in this thesis, will consist in studying the field of proposal portals and understanding what flaws these systems contain and what needs improvement, like having room to

improve the experience of the users and reviewers or having an opportunity to increase the efficiency with an element of taxonomy.

The context observations is the study of the public or area that will be impacted by the analysis and research about how the work has been done before[6]. In this case, the context observations contains the study of different platforms used to connect proposal creators and candidates, as well as the study of ontologies and different options to implement one in website development.

The third stage, Formulate Hypothesis, is to state a speculation, precise and well defined, so the goal of the work/research is clear: To validate or deny this speculation. It is possible to have more than one hypothesis[6]. In this dissertation the hypothesis must be around whether its possible to use an ontology to create a better proposal portal.

After formulating the hypothesis, the researchers know what they are looking for, and so enters the fourth step: Design Experiment. This is where the experimental part begins, so the researcher must delineate the steps necessary to validate the thesis. This often is creating a system architecture, programming or designing a prototype. It is important that the procedure is explicit and can be easily repeated, so its clear the context in which the results and conclusion were taken. This is also helpful in case a third party wants to find how to replicate this study [6]. Designing a mock up for the website and dissect the stages of creating an ontology would be the bulk of this part of this step.

When the experiment is designed, it is necessary to implement and test the proposed thesis through the design experiment. That is the fifth step and is where the investigator implements methods or prototypes, creates simulations and works in the field[6]. For the present dissertation, implementing an ontology, programming and creating the website will be the main tasks in this phase, as well as using different tests to understand if the work has improved communication between proposals creators and candidates, validating or not the hypothesis.

In the end, the last two steps are analysing results and thesis writing. Analysing results consists in examining the tests and drawing conclusions. This typically includes clustering data and can either look like a qualitative or quantitative analysis. Its imperative that the researchers have a specially critical way of thinking in this phase so the conclusions are as real as possible. It is also important to speculate and be aware of possible factors that could mislead a conclusion. Finally, the last step is to write the dissertation. It is creating a document to register all the steps mentioned before and the conclusions drawn regarding the hypothesis proposed. After writing the dissertation it can be published.

If in the sixth step, the conclusion is that the hypothesis is false or partially false, it is typical to return to step 3 and try to rebuild a new hypothesis and retest with a goal to validate it. In the Figure 1.1 its possible to see a visual representation of the scientific method.

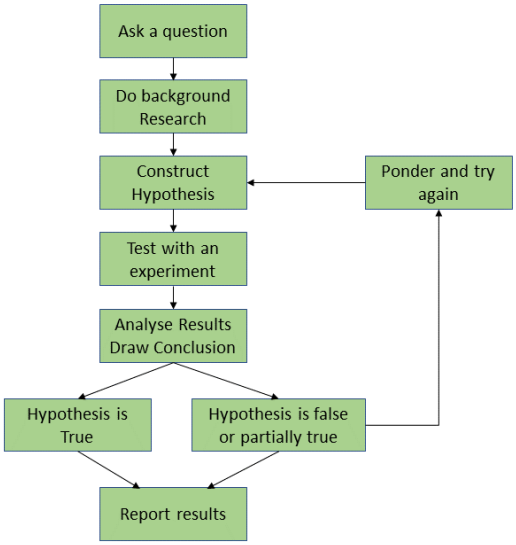


Figure 1.1: Steps of the scientific method (adapted from [6])

1.4 Hyphotesis and approach

According to the last section, once the problem and state of the art are complete, the third phase takes place.

Taking in consideration the research done in the previous chapters, it is possible to define one hypothesis that is going to be executed, tested and confirmed or not in this dissertation:

> "If it is possible to create a portal for submissions of proposals, then we are able to do it in a digital way, where both clients and reviewers have a fast, simple and easy experience navigating it by organizing and structuring the data base?"

2

State of art

The goal of this dissertation is to organize and structure the information of a submission portal into an ontology and create a platform where the whole process can be accessed through and that facilitates the assignment of the administrator to make decisions regarding the proposals. It is necessary to study what has been done so far. That includes the study of submission platforms and how have they evolved over the years, similar platforms with the same goal and for last the study of ontologies, what it is and how can it be helpful.

In this chapter there will be firstly a summary of what are portals of proposals, their goal and how have them worked trough the years; after that, there is an analysis of similar platforms, how they work and their design. The third section is directed to Ontologies, their purpose, different tools and languages of ontologies.

2.1 Portals of proposals

A portal of submissions is a platform where are listed different open calls for proposals in which someone can make a proposal and submit their project, candidature or proposal[1].

There has been always the need to share calls for proposals and projects. In the same context, there has been always the need to submit proposals. For a long time it was used newspapers as a tool to share this calls and the submission would be through letters. For instance, if a company wanted to open a call for their marketing strategies, it was necessary for the company to pay an advertisement in at least one newspaper, with a description and requirements of the call and an address for candidates to apply. Then it was necessary to wait a considerable amount of time for proposals, due to being letters and after that the reviewers would have to verify all the requirements individually. Only after that, the candidates would hear back from the call, most likely through letter with the outcome of the call. This process was highly slow, hardworking and inefficient. After that, when internet appears, there was an evolution from newspapers to websites or portals and letters to emails[1].

Nowadays, this process is most of the time done by portals although there is still a

lot of submissions trough email. There are some cases where it is possible to submit the proposal directly in the portal and after that the platform transfer the data from the proposal to a spreadsheet to help the reviewers and administrators to help analyze the proposals. Nevertheless this portals are still not automated at all which makes difficult a process that could have been evolved immensely in the later years with the evolution of artificial intelligence and data analysis[3].

2.2 Similar platforms

There are multiple websites that have similar working processes and functionalities. In this chapter, its going to be numbered some of these websites and compare their different performance and components.

The first step to building a framework like this is to research what already exists, similar to what has been proposed to do. It's necessary to compare related websites, identify their characteristics, and pinpoint which ones are helpful and necessary to the creation of this one.

With that in mind, it was decided to conduct research for similar websites, frameworks, and even papers that are related to website development.

The websites stated below were the ones that were found to be closer to the end goal of this thesis:

- European Commission - Funding & tender opportunities
- F6S
- MarTera
- EMS

2.2.1 European Commission - Funding & tender opportunities

The European Comission (EC) has a complex website which contains all the different areas where it operates and the work that is done by them. In particular, there is a large section of this website similar to the website proposed. It's the European Commission—Funding & Tender Opportunities, which is a (Single Electronic Data Interchange Area (SEDIA)). In other words, this platform function is to help people and entities find a funding program or tender as well as being able to apply for them on the very same website. [7]

The analysis of this platform will be more extensive due to the resemblance between both goals.

When the current website was developed, there were some guidelines that were followed in order to get the best outcome possible.

First, the website must have a constant layout design, so there is no doubt whether a certain page belongs and was developed by the EC (European Commission) or not.[8]

Second, all information on the website must be useful, appropriate, and aimed specifically at assisting the user with their task.

Third, the content must be concise and easy to understand regardless of the person behind the screen.

Regarding the third principle comes the fourth: it's essential that the website is accessible. As a result, being easily translated into other languages, developed in a way that is accessible to various types of devices (smartphones, tablets, computers), and having features to assist disabled users are imperative[8].

It's important to understand why EC made this decision. This website is used by all types of people, from all over the world with the most diversified jobs, that visit the website with a task in mind and would like to complete it as fast as possible. It makes sense that one of their bigger priority is to make the website accessible[8]. In the case of the thesis website, the user has a task in mind when enters the platform, so it must be a task oriented website, but the target user will be someone from an company or entity and will also most likely be using a computer, so although accessibility is a important factor, it is not the priority.

Lastly the EC website is structured in a architecture building which aims to create a logical website to organize all content and functionalities. For this, it was used two tests to verify the concepts as they evolve.

The first method, card sorting, doesn't give an answer on how to organize the website, but can be an interesting tool to understand and dictate if the content is understandable and relevant and it has 3 types, open, closed and hybrid. This method consists on creating individual cards, with different concepts and features that the website will have and ask target users to sort them into categories. This method can also be moderated (with a researcher present) or unmoderated (online, for example).

The second method, tree testing, is best suited to after the card sorting method and is used to understand where the user would first to go find a specific information. In summary the tree test consists in asking a group of people to what would they think that a certain category would contain.

Both of this tests are extremely helpful in order to have the best organization and structure possible and to get feedback from the users.

2.2.1.1 Layout

When the platform is started, it goes automatically to a home page where it's possible to search for calls through key words, to find news from the European Union, how to participate in the offers, some useful links, and in the end, some FAQ (Frequently asked questions). On all the pages of this website, there is the same bar on the top with several buttons, each of which has its own menu. This bar has the following buttons: home page, "search funding & tenders", "how to participate", "projects & results", "work as an

expert"and "support"as it can be observed in Figure 2.1. It is obvious that the website is well constructed, with easy access to every section and very user friendly.

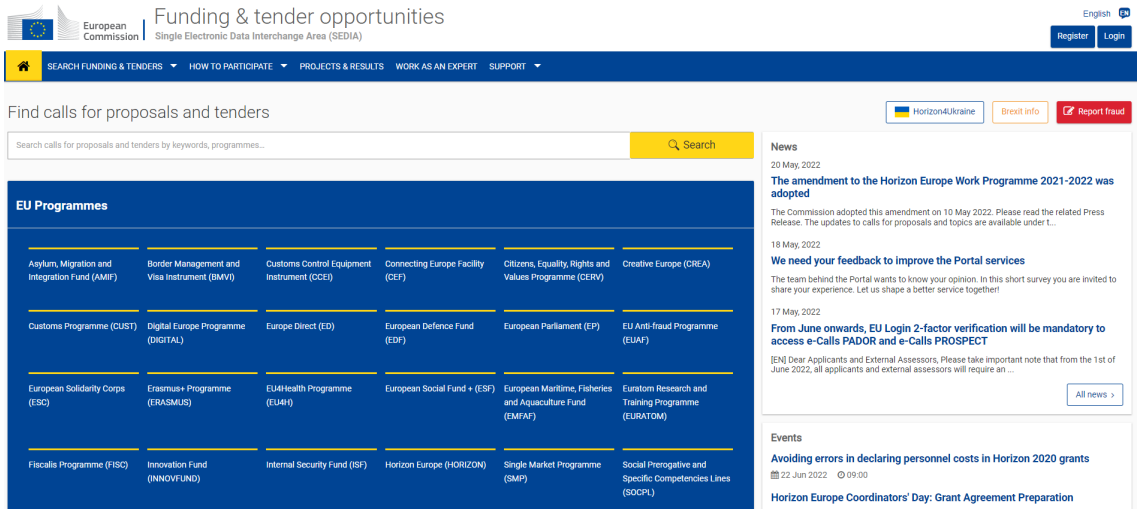


Figure 2.1: Europe Commission- Funding & tender opportunities- Home page before log in

For the registration process in this website, it's kept simple with being only necessary a name, email and for security is sent an email with a link to create a password. So the registration is made with few steps but its secure, which is an advantage. There is a small inconvenient with this method which is when the user sets his password, the platform doesn't redirect automatically to the home page, the user has to open it manually again.

When the login is made, there is a new menu bar on the left side called "Manage my area"where there is a "My Person Profile", "My Organization(s)", "My Formal Notification(s)". This change of layout can be observed in Figure 2.2.

On the "My Person Profile"is possible, as the name suggests to create a Person Profile, meaning a individual profile where the user need to upload their personal details like nationality, country, language, biography as well as their education, publications, employment history and, for last, important documents like for example recommendation letters. This functionality facilitates the application process to this funding programs.

On the second section, in "My Organization(s)"its listed all the organization associated to your personal account and the possibility to create one. When a new one is created, the European Commission gives a PIC number, which is a Participant Identification Code with nine digits. This code is also a requirement in the project for this thesis.

On the third and last section, the "My Formal Notification(s)"all the formal notifications for all the users proposals and projects are listed. This page has also the functionality to acknowledge the notification when needed.

Analysing the horizontal menu, the first button "search funding & tenders"extends to a drop down menu with 3 options: "search funding & tenders", "funding updates"and "archived funding". The first has the purpose of helping the user to find a favorable funding programme. There is load of different filters to help the user to search from

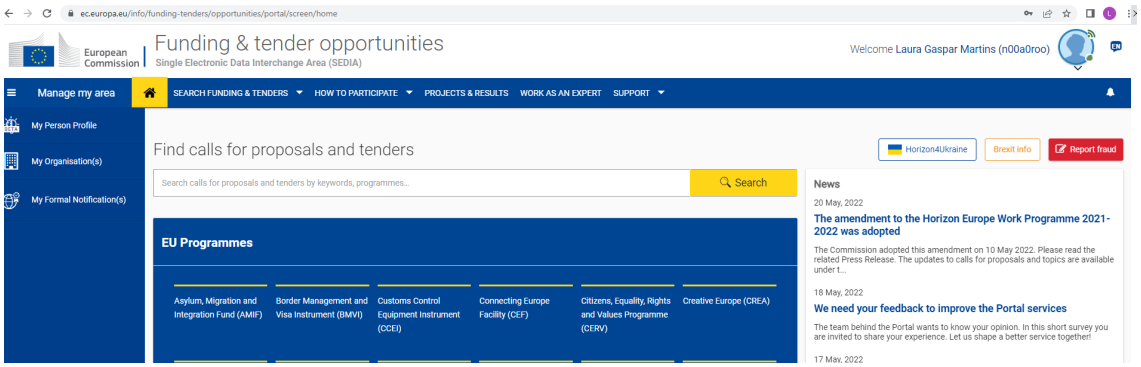


Figure 2.2: Europe Commission- Funding & tender opportunities - Home page after log in

keywords, submission status to filters by call that can be observed in Figure 2.3. This feature would also be helpful to have in the Portal of proposals.

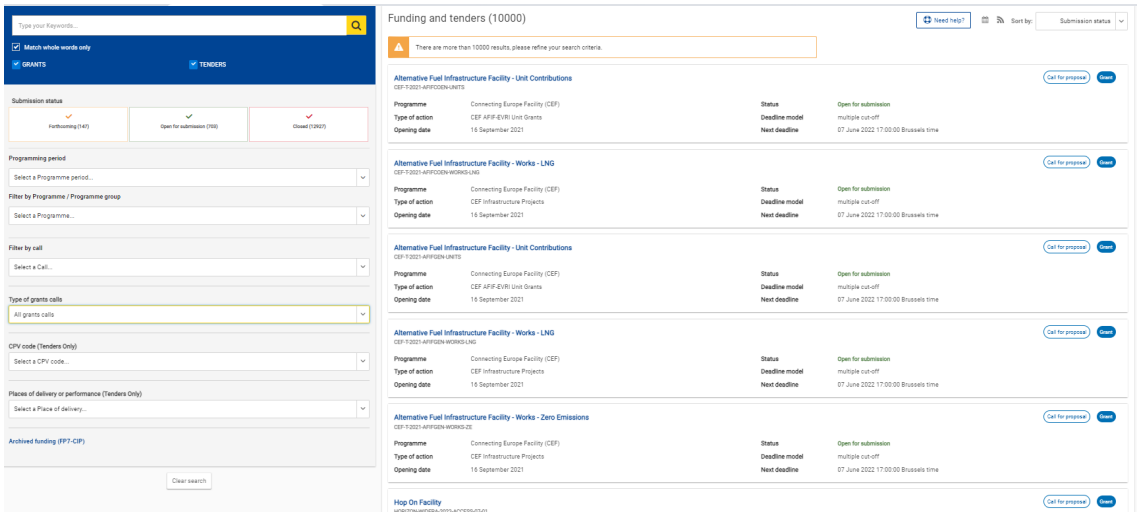


Figure 2.3: Europe Commission- search funding & tenders - filters available

The "Funding updates"opens a page with all the updates of the listings organized either by date or call. This page also has some filters available. The last option, "archived funding", directs the user to a new page of "Research & Innovation".

Although it's useful to have this differentiation between the different types of search for funding, it's not the most user-friendly, making it confusing which section serves which purpose.

On the next button, "How to participate"also has associated a drop down menu with the following sections: "key steps", "Reference Documents", "Participant Register"and "Partner Search".

On the first one, it is a simple page with info about the requirements to participate, with 2 subsections, one for grants and another one for tenders.

The second section, "Reference Documents,"is an easy way to know which documents are necessary for each type of programme. Once again, it is possible to search for a specific

programme.

The third button, "Participant Register", is a section where it's possible to search for a PIC if the user has an organization that is already registered and an option to register an organization in the case there is a need.

And last, the "Partner Search"is a functionality that lets users find other organizations and personal accounts also through filters.

The fourth button on the horizontal menu, is the "Projects & results"button, in which the user can get information about the work that has been done in each programme. Some programmes are more detailed than others, and some even present interactive data.

The last two buttons on the horizontal list are "Work as an expert"and "support".

In "Work as an expert"there is some information about how to become an expert, which means evaluating candidates for funding programmes and proposals and supervising actions, agreements, and contracts. This page also allows the user to register as an expert, which will redirect them to a completely different page, where it's necessary to fill a form. The remaining steps and overview of the website as a evaluator are not accessible to the remaining users so it was not possible to review that section. In the "support"button there is a drop down menu with 4 options. The first one is a general support page, the second one has some documents to help and guide users, the third is a frequently asked questions and for last it's a page with some support videos.
As a summary, there are a lot of similarities between the goals of the two websites. Both have the purpose of creating some type of contest (in the first case, a funding or grant contest and in this thesis, a project contest) and both have three types of users: an administrator, who creates contests; an evaluator, who evaluates the proposals made to each contest; and lastly, the applicants, who make proposals to a contest. It is not possible to see the way the website works for the administrator.

After this analysis, some aspects of the European Commission- Funding & Tender Opportunities that would be a good functionality for the website that is going to be built are the different types of account (before log in, after log in and evaluator) and also the many different times the feature filters are available.

When it comes to having an automatic system that evaluates partially the proposals it doesn't show evidence of that being implemented in this website.

2.2.2 F6S

Another website that has numerous similarities with the one proposed in this thesis is F6S which is a website that, as they state themselves, "Is the world's largest platform for founders", so it connects founders of companies, mostly start-ups, with anything they may need to become successful, from investors, accelerators, to any talent needed.[9]

The website works thru profiles. So to access the information you need to sign in first. Its possible so sign in either thru *Facebook* or *Linkedin*, or with the usual process or registering with an email. This is a nice feature because turns the registering process way faster and user friendly to users which is one of the goals of the website, to be a task-oriented, so the user doesn't need to waste time in unnecessary steps.[10]

After registering, the website has an horizontally bar menu, which is divided into four main areas:

- F6S (homepage);
- Apply;
- Events;
- Jobs;
- Deals;

There is a search bar, as well as a message icon and still in the horizontal bar there is three drop down menus. The first one opens to a side bar (vertically). The second drop down menu, called "Add your"is where the user is able to create different types of content. Lastly, there is a drop down with the profile user information.

The rest of the homepage is divided in 5 horizontal sections:

- "HERE'S HOW F6S HELPS YOU GROW"
- "TALENT MATCHMAKER- YOUR WEEKLY TOP 5";
- "POPULAR BENEFITS";
- "STARTUP PROGRAMS NEAR YOU";
- "CONTESTS NEAR YOU";

In Figure 2.4 its is possible to observe the general layout of the homepage.

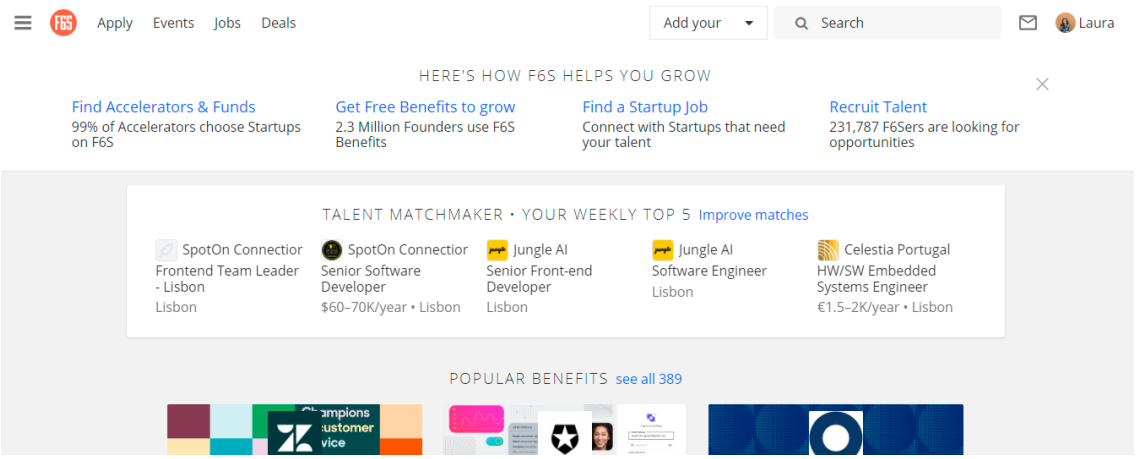


Figure 2.4: f6s website - homepage [10]

The first drop down menu has only two options, "Update your profile"and "Add a job". The "Update your profile"leads to an extensive profile (Figure 2.5) editor where basically you can insert all the features you would like in a curriculum from skills to education. Has a grading of your profile strength, with improvement recommendations and has a unique feature where it is possible to ask for a recommendation from another F6S user. The "Add a job"button where its possible to add information, in the users account, about the company the user is currently working at, so they can add a job opening. To add a new job there is a form to fill with various components from job description to salary.

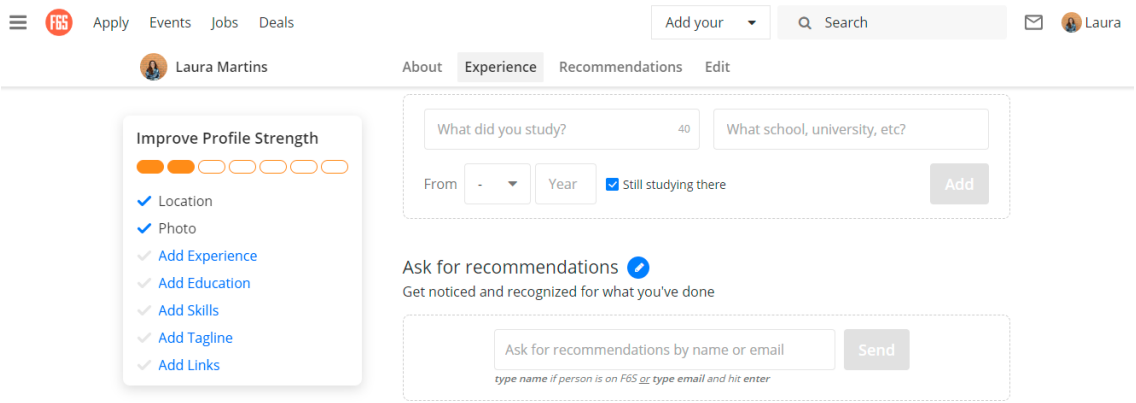


Figure 2.5: f6s website - homepage [10]

The "Apply"page is where it is possible to apply as a start-up for programs whether they are: events, investment funds, accelerators or events. There is a listing with all this options that can be ranked by date, popularity and geographically distance. Each offer has a small image, usually with the logo of the company, a title of the offer, location (city and country). Additionally can have a quantity of money given per team. This page has a feature of filters, such as the type of program desired, location, type of market or type of investment. For a user to utilize these filters there is an horizontal bar with each filter, that slides into a drop down menu with the options of each filter. Since this is an international

website these kind of features like filters are very important because otherwise it would be impossible to find a desired offer. Still in the "Apply"page, there is two extra sections. One called "close to you"which recommends some projects near the user and the other one is "list me here"which redirects the user to listing a company for an accelerator.

The second button on the menu bar, "Events"is a page with a similar layout to the "Apply"page, with a list of different events, where each of them have a button to book it. Once again, the user can sort the events through popularity, location and time of proposal and use filters to adjust the search. Like the other page, there is the same "close to you"section but only with events and "list me here"but it redirects you to listing an event. When the user clicks on "book"an event, it redirects the user to a page about the event, with three sections, "View Event", "Discussion"and "Get ticket", it is also possible to contact the organizing team, which enhances the experience. Here it is possible to have all the information a client may want about the event.

The "Jobs"and "Deals"are similar to the "Events"but instead of listing events, lists jobs and deals correspondingly. It is possible to observe that the F6S website made an effort to keep the website simple and uniform, with the goal of simplifying the user experience when navigating though the different pages of the website.

When a client desires to run for a job or deal is directed to a form to complete. The components of the form vary from different jobs and deals although for multiple jobs applications the only requirement is a text about the reason for the candidate to want the job. For all of these areas, it is possible to apply on the website for each specific program/event/job with a form.

Still in the horizontal bar, there is a drop down menu called "Add your", where the user is also able to create his own job listing, startup, accelerator, benefit/deal, event, contest and fund. For each of these options, the website requires the user to indicate from which company or start up is, and, after that, it directs to a page with another form.

The last two buttons are a messages icon where it shows all the unread or pending messages of the user's account and a profile icon with a drop down with all the companies or startups registered with the used account, where it is possible to manage the company profile and create a listing. Other options in this menu is to "update your profile", "Help", "Settings"and "Log Out".

After reviewing the website it is possible to characterize the present website as well constructed and easy to navigate. Although, it is interesting to notice that there are multiple paths to execute the same task.

Another detail to enhance is that all the features in the first drop down menu are accessible through another path, being that all of them are more likely to be used by the client than that drop down menu. So it would be interesting to study if this menu is necessary or even helpful.

When it comes to the tools used to create this website, it was used jQuery, a javaScript Library that makes the process of using in HTML easier; it was also used the programming

language PHP as well as the web server Apache; lastly it was used CloudFare and Google Cloud Platform as clouds.

2.2.3 MarTera

MarTERA is an ERA-NET Cofund scheme of Horizon 2020 of the European Commission that funds multiple research projects that impact different application fields. To create calls for the funding projects, MarTera uses an online submission tool where all projects are submitted to be evaluated. It is possible to observe in Figure 2.6 that the submission tool has four types of profiles: coordinator, partner, funder and evaluator.

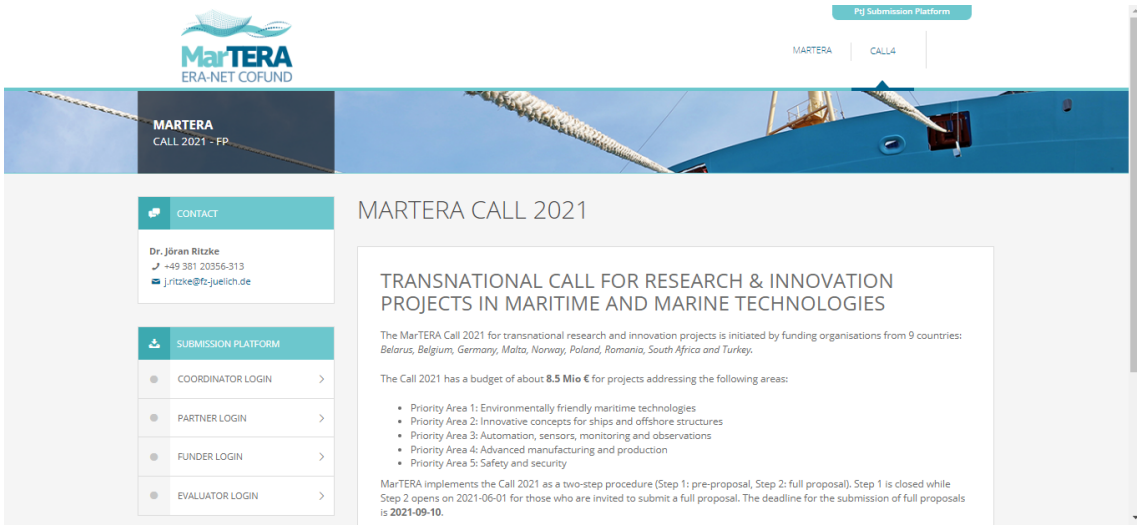


Figure 2.6: MarTera Website - Submission Platform[11]

Due to being a private platform, it was not possible to further the study of how the website works for either of this profiles although is noticeable that the goal of the platform is very similar to the web portal proposed in this thesis.

2.2.4 EMS

EMS is a tool created by CARSA, that has the purpose to support public and private clients at evaluating and managing open calls for proposals. According to CARSA [12], this tool supplies a completely traceable administration, including being secure, clear and monitoring of projects in all actions executed.

Although there isn't a lot of information about this product, given that it is a charged product, it was the only tool found in this search that has the component of an structure that supports the evaluators and so, very relevant to this thesis.

2.2.5 Comparison

All of these websites and tools have different functionalities and it is important to have them in consideration when creating a new platform with the same goal.

Most of them have the same structure, which makes possible to conclude that it is probably the best way to conduct this portal. So features like menu bars, drop down menus and searching bars should be included in the goal for this dissertations.

Another important tool that was present in all of the websites are filters. It is vital that the users have the ability to decrease the number of calls in order to find the desired one faster. Besides this, having different types of profiles seems to be the best way to organize this genre of platforms, since all of them have used it.

The European Commission guidelines as well as the tests are very helpful tools and are planned to be used in this thesis as a helpful tool to have the best outcome possible.

For last, the main disadvantage that all similar platforms revealed was the lack of support and automation in order to help the administrator in making decisions about the proposals. The only platform that show kind of this concept was EMS, but derived from being private it was not possible to study the repercussions of this tool.

Taking this need into account, it is necessary to investigate more about data structures and ways to organize information. That is the purpose of the next chapter.

2.3 Ontology

In order to create a system with an automatic component that helps the administrator make decisions, for instance to chose the most qualified reviewer for a proposal, it's necessary to access all the data about the proposal and the data about the reviewers, organize it in categories and run them through an algorithm that finds the best candidate for this job. In this thesis is planned to create this structure of information so that later an algorithm can use it.

There are different ways to structure and order data or facts. One of the most used is ontologies. Ontologies have the ability to categorize varied objects and define relationships between them, being an important tool to organize data.

2.3.1 What is an ontology?

The concept of Ontology started in a philosophic manner and the term is thought to have arisen with greek philosophers , around 2,400 years ago [13] and refers to the study of being. Essentially, studies how is it possible to categorize concepts, relate one to another and how. Defining the world as a set of objects, and grouping these objects in categories with similar properties is one of the concepts associated with Ontology.[14]

Humans create ontologies in a unconscious way in their minds through sentences and statements and use them to navigate every day life situations. Due to machines inability to do this naturally, there was the necessity to translate there relationships between concepts in a machine language.[15]

The first definition of Ontology in technology was in 1993 by Gruber[13]

> "an ontology is an explicit specification of a conceptualization."

This definition, although quite ambiguous expresses the main goal of an ontology, to define and conceptualize an object. Later, as Ontology's started to appear in logical technology and computer science, Neches described an ontology (in a technological sense) as [16]

> "an ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary"

This definition has a little more of practical component, where it offers a general definition on how to create an ontology. It is important to notice that this definition already includes the description of the relations between objects and terms. [16]

After some years, there is a consensual description of an ontology as a limited group of objects of a domain and the respective relationships between this objects. [16] These relationships can include hierarchies or not.

Ontologies can be described in four categories [17]: Upper Ontology, Domain Ontology, Interface Ontology and Process Ontology.

Upper Ontology is a description of a general concept that can be used with different domains. Domain ontologies refer to the collection of concepts and notions of a specific domain. On the other hand, Interface Ontology is a clear declaration of patterns that offer useful information modern techniques. The last type, Process Ontology describes the group of related terminology in a process.

The most common components in ontologies are [18][19]:

- Individuals - The base of any ontology, it can be real objects like people, beds, cars or evens mars, but it can include subjective concepts from numbers to adjectives, to words. Although there is a disagreement on whether these subjective concepts are individuals or classes.;

- Classes - Categories or ways to describe an individual. Some examples of classes are person, car, thing. There is two main ways to describe a class, extensional or intentional. Extensional is when classes are described as if when there is two classes with the same members, the classes are identical. There is a problem with this definition because for example if the classes are "creatures with a stomach"and "creatures with a heart", there his high probabilities that this classes have the same member, but they are not identical. An intentional approach defines that to belong to a class there are conditions correspondent. [14];

- Attributes - Properties of an objects, commonly aspects. Attributes can be considered individuals or classes. For instance, some attributes of a person may be: Laura (name), Martins (surname), 22 (age), engineer (job), Lisbon (address);

- Relations - Connections between objects. Name, surname, age, job (the description of attributes, in parenthesis) are all considered relations in the example above. The class and subclass taxonomy is also considered an relationship. [20];
- Functions - Individuals created with relations in a elaborate structure;
- Restrictions - Definitions that must be validated to an object or relation;
- Rules -Structured in a if-then way statements that describe the relationships between individuals in a particular form;
- Events - Modifications in the ontology of attributes or relationships;

Ygor Cardoso uses an example of an ontology, in his thesis [16], of a hierarchy of a university. In this case (that can be observed in (Figure 2.7)) the domain is the university people where there is two types, staff and students. Staff people can be categorized in three sections, academic, administration and technical. For last, there are two classes of academic staff: research and visiting. As for the students, the university can have undergraduates and postgraduates.

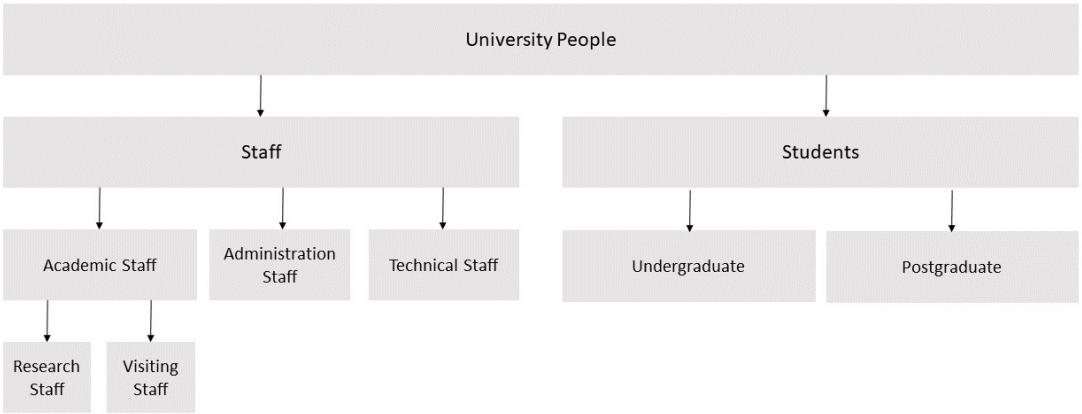


Figure 2.7: Example of an ontology, adapted from [16]

This is an example of an ontology with an hierarchy, where every layer belongs to the layer above, with a class system. Ontology's can have other types of relationships and even the one pictured in (Figure 2.7) could be only a part of a bigger and more complex ontology. For instance, "University people"could be an object of "University"along with "Courses", "Faculties", where the classes related to each, would later have other relations with other.

2.3.2 Ontology development tools

In order to create and use an ontology, it's important to know different tools and languages that are usually used in this field.

The first tool relevant to mention, that is consistently used in Ontologies is Extensible Markup Language (XML) that it is, as the name implies, a markup language. A markup language is notes added to text in a document, such as codes or tags, in order to modify the content (aspect or meaning). [21] [22] The main use of XML is to link data between applications, so it is a generic way to easily create a format that machines can read as well as humans. File documents like Office Open XML, OpenDocument, SVG XHTML are created in a XML syntax.[21] In the field of ontologies, some examples that use XML are Resource Description Framework (RDF) and RDF Schema, Defense Advanced Research Projects Agency (DARPA), DARPA Agent Markup Language (DAML)+Ontology Interchange Language (OIL), Web Ontology Language (OWL) and OWL2.[22]

2.3.2.1 Ontology languages

After understating what an ontology is, is important to define an ontology language. An ontology language can be characterized by a formal language used for coding an ontology. [23] There are multiples ontology languages, in this thesis there are going to be described the most used ones.

One of the first languages that was created is RDF, that stands for Resource Description Framework, is a data model and one of the reasons why it gained so much popularity was being one of the language recommendations by World Wide Web Consortium (W3C).

RDF is a language that uses XML and is used mostly to define objects through statements. This statements are compost by three components. One component is a resource, which uses an Uniform Resource Identifier (URI) to be characterized. The other two components are property and property value, where a property is the resource attribute and property value is the value of the attribute (the value can be another RDF resource, as long as an URI is used). [14]

RDF Schema can be considered a type of RDF for a simple system and is also a W3C recommendation. It supplies a system with classes of resources and enables specific properties. [24]

OWL (Web Ontology Language), was created after RDF, but it gained a lot of popularity in the last years. It was built according to the standard RDF (Resource Description Framework) and therefor to the W3C. [23]. The intention when created OWL was to conceptualize ontologies in a better way for applications to process information and to work through different machines.

Axioms are the basic construction of an OWL language. A collection of axioms, referred to as "classes,"and the kinds of interactions that may exist between them are defined by an OWL ontology. These axioms give meaning by enabling systems to deduce extra information from the expressly presented facts. It contains a strong vocabulary and very formal semantics which makes possible, for instance, classes having equal relationships, restricting class properties or have more complex properties. [14]

As time has passed, there have been created multiple variations of the OWL language.

In summary, OWL or OWL 1 is referring to the specifications created in 2004 and OWL 2 is referring to the 2009 version. Besides that, in OWL 1 there is three sublanguages: OWL Full, OWL DL and OWL Lite. OWL 2 has another three sublanguages: OWL 2 EL, OWL 2 QL and OWL 2 RL. [25]

OWL Full uses the OWL language bases which makes possible to use the RDF syntax more freely as well as allowing blending RDF with RDF Schema which creates its better quality: to be completely compatible in terms of syntax and semantic with RDF. OWL Description Logic (DL) is a language based in OWL Full but with some restrictions like being necessary a dissociation between components such as classes, instances, properties and data values. For last, OWL Lite is even more restricted than OWL DL for example doesn't allow classes with numbers or random cardinality.[14]

2.3.2.2 RDF vs RDF schema

The best way to compare RDF and RDF Schema is to observe Figure 2.8, where it is possible to observe that both have class, data, object, domain, range and individual properties. Their principal differences are RDF Schema has better inference, indexing and searching properties and has three more properties than RDF: Inverse, SPARQL Query and DL Query. Overall RDF Schema is a more complete language than RDF.

Properties	RDFS	RDF
Class	Yes	Yes
Data	Yes	Yes
Object	Yes	Yes
Domain And Range	Yes	Yes
Annotation	Yes	No
Individual	Yes	Yes
Graph	RDFS	RDF
Inverse	Yes	No
Inference	Good	Poor
Indexing	Good	Poor
Searching	Good	Poor
SPARQL Query	Yes	No
DL Query	Yes	No
Prefix	RDFS	RDF

Figure 2.8: Comparison of properties between RDF and RDF Schema [17]

2.3.2.3 OWL vs RDF Schema

One of OWL biggest differences from XML or bots RDFs is that OWL works mainly through axioms and it is easier work with multiple machines. In Figure 2.9 is possible to

observe a comparison between both RDF(s) and OWL 1 and OWL 2. In terms of concepts RDS Schema lacks all of the enumerated in the table except having formal semantics. When comparing OWL 1 and OWL 2, OWL 2 has the advantage of having property chains, disjoint properties and qualified cardinality restrictions, which OWL 1 doesn't.

Concepts	*RDF(s)*	*OWL 1*	*OWL 2*
Formal semantics	✓	✓	✓
Equivalence	×	✓	✓
Class definitions	×	✓	✓
Constraints	×	✓	✓
Enumerations	×	✓	✓
Cardinality constraints	×	✓	✓
Inference	×	✓	✓
Property chains	×	×	✓
Disjoint properties	×	×	✓
Qualified cardinality restrictions	×	×	✓

Figure 2.9: Comparison between RDF(s) , OWL 1 and OWL 2 [25]

After this comparison, it's possible to conclude that OWL is able to create ontologies more complex than both RDFs (RDF and RDF Schema) due to having a bigger vocabulary as well as a better syntax, which makes OWL an overall better language, specially OWL 2. [25]

2.3.3 Ontology modelling editors

In this section it's going to be discussed different applications to create ontologies.

Protégé is one of the most used ontology editor, created by Stanford Medical Informatics with the goal of a framework of Java and recently has become available a web version. [26] It contains two modelling possibilities, Protégé-Frames and Protégé-OWL and its advantages is being free and having a Java library available. [20]

Another application to create ontologies is NeON Toolkit, which is used in Eclipse, an IDE (Integrated Development Environment). It is an application similar to Protégé and it allows to create OWL and RDF ontologies.[26]

OntoEdit is a more modern editor, provides elementary and user-friendly support as well as allows reversibility and transparency. Uses F-logic as an base language, saves the ontology in a file format and has an ontology library. [27]

All of this editors have a feature that analyses and finds conflicts, although none of them can inform the user where is the source of the conflict. As a general consensus, it is established that Protégé is the best editor for beginners due to its simplicity although all of them are fairly similar and simple.

Below, in Figure 2.10 it is possible to see a comparison of multiple Ontology modelling editors on various categories. In this thesis it was only mentioned the editors that are more plausible to be used therefore some of the editors listed weren't mentioned.

	Release Date	Base Language	Imort/Exports from/to Languages	Exception Handling	Ontology storage	Availability	Ontology library
DUET	17/03/2002	UML	DAML+OIL	Yes	No	Free	No
UBOT	09/2002	UML	**Inp:**UML , XML, **Exp:**DAML, Slang	Yes	No	Free	Yes
OntoEdit	04/03/2004	F-Logic	RDFS, F-Logic, DAML+OIL; RDB, schemas	No	Files	Free	No
Protégé	22/06/2004	OKBC+ CLOS based meta-model	RDF, RDFS, DAML+OIL; XML, OWL, Clips; UML	No	Files &DBMs (JDBC)	Free	Yes
OiLED	31/10/2003	DAML+OIL	RDF URI's; limited XML Schema, export : HTML.	No	Files	Free	Yes
Ontolingua	11/2001	Ontolingua	**Imp/Exp:** KIF, OKBC,Loom,Prolog, Ontolingua, CLIPS **import only:** Ocelot, classic, Protégé	No	Files	Free	Yes
OntoSaurus	03/2002	Files	LOOM, IDL, ONTO, KIF C++	No	Files	Open source	No
WebODE	03/2002	HTML forms and Java applets	**Imp/exp:** XML, RDF(S), XCARIN, OWL **Exp:** OIL DAML + OIL FLogic, Prolog Jess, Java	No	DBMS (JDBC)	Free	No
WebOnto	05/2001	OCML	**Imp/exp:** OCML **Exp:** Ontolingua GXL, RDF(S), OIL	No	Files	Free	Yes
Swoop	08/2007	OWL	**Imp:** OWL, XML, RDF and text **exp:** RDF(S), OIL and DAML	No	HTML models	Free	No
Dogma Studio Workbench	Not reported		OWL, RDF, DAML+OIL			Not free: OnDemand	No
TopBraid Composer	2011/06/04	RDFS/OWL	**Imp:** RDFa, OWL, RDF(s) ,XHTML, Microdata, and RDFa, Data sources, SPIN, **Exp:** Merge /Convert RDF Graphs, RDF(S), OWL	Yes	Files	License	Yes

Figure 2.10: Comparison between Ontology modelling editors [28]

3

Concept & Architecture

This chapter evidences, in detail, the objects, their characteristics and the relationship with each other. This process was done with the trial and error method, multiple different databases were drafted in order to create one structure that holds all of the necessities of the portal.

Once our goals are defined it is necessary to implement and create the structure that will support the concept desired. As it was chosen to use an ontology as core organization of the database, the first step into the implementation is to determine our objects. It was made a draft of how the main concepts are connected that can be observed on Figure 3.1.

It is described an overview of each type of user of the portal, and which actions each is responsible for.

The users are every individual that uses the portal. It can be someone who wants to create calls, to summit proposals to a call or maybe someone which wants to be a reviewer. All of this users are important and the portal needs to save data about the user and his goal on the portal. As there are three main types of users that use the portal, it is important to define the different actions made by each type of user, what information is accessible to each user and which decisions can and should be made by who.

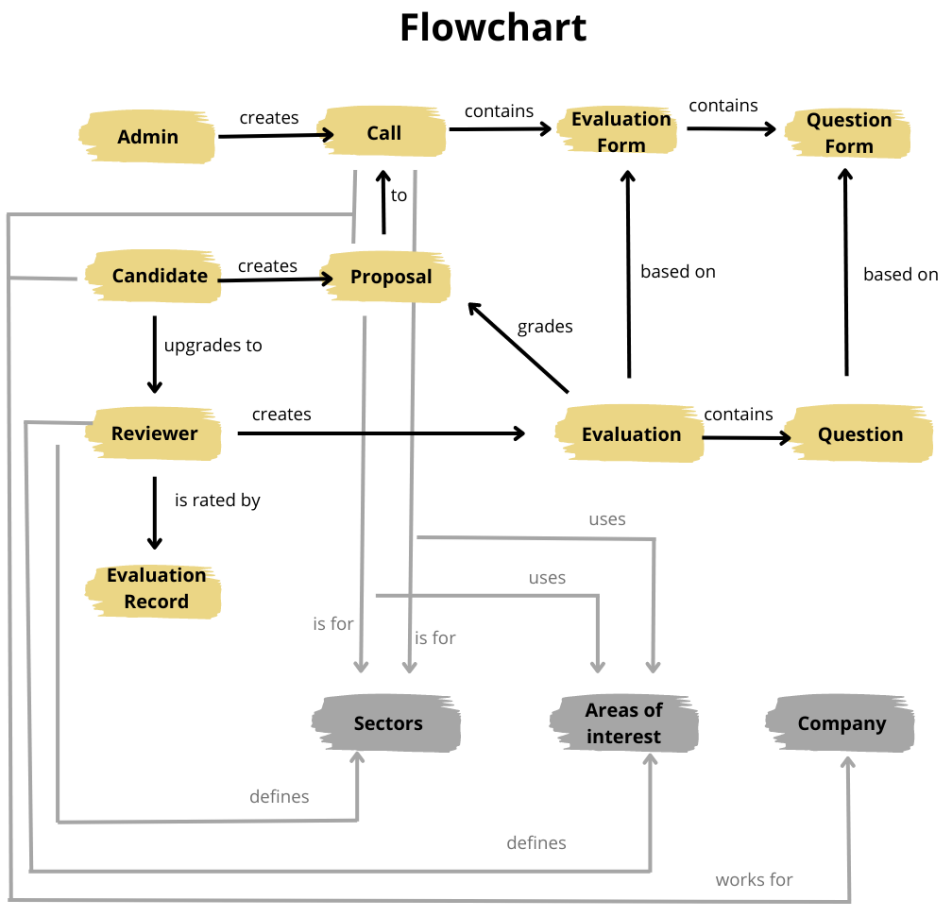


Figure 3.1: Portal Stakeholders Roles and their Interactions.

The 3 types of users have the following definitions:

Candidates - This is the user which has the most base level of all users. When a user when creates an account automatically becomes a candidate. This type can see all the calls available and can create proposals to each call. This will be the majority of the accounts.

Reviewers - A candidate has to make a request to became a reviewer which involves creating a small *resumé* with past experiences on different sectors and areas of interest. A reviewer can be offered to be a reviewer of proposals, in which will review different proposals for a call. Besides that the reviewer still has the possibility to perform every action of the candidate as long as it doesn't interfere with his own tasks.

Administrator - The last category of user is the administrator. The website can have multiple administrators and they are responsible for creating calls, choosing the reviewers and reviewer form for each call as well as upgrading candidates to reviewers.

With the definitions above, is observed that there is a hierarchy on users, which is represented on Figure 3.2. The base level is candidate where you can only do the simple tasks as seeing the available calls and submit proposals to them. The next level,

reviewer, can do the same as candidate but can also review other proposals. Lastly, the administrators are on the top of the hierarchy, having the possibility to do the same actions as candidates and reviewers, but can also create new calls and choose reviewers.

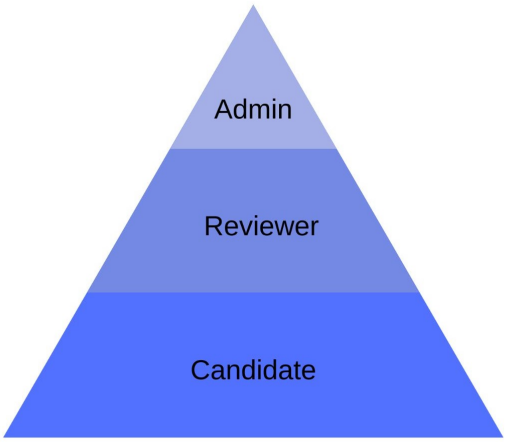


Figure 3.2: Types of users

Each user has information that is mandatory to store on the database such as an unique identifier, like a ID; a variable that differentiates between the types of users; an email (which also has to be unique, for login purposes); a password (for security purposes) and the name of the user. Additionally should be saved the occupation of the user (to be visible on the proposals submitted); the company in which the worker is from and the country where it is resident from.

The next sections will describe the usual process that user will follow on the portal.

3.1 Candidate

When a new account is created, it is automatically considered to be a candidate profile, as this is the most basic level of user. On Figure 3.3 is the follow-up of the tasks which a candidate can perform and how they are connected to each other.

When a candidate enters the portal can see all open calls or calls that will be open in the future. This call must clarify the goal and rules of the call, what is expected, the company in which the call is based as well the important dates that proposals have to meet. When a candidate finds a calls that they are interested in submitting a proposal, they can create a proposal. The candidate can save draft versions of the proposals and when the candidate considers the proposal finished, submits it. The database is able to structure the same candidate create more than one proposal for the same call and can be on the process of creating multiple proposals at the same time or can restrict one proposal per user.

Once the proposal is submitted the candidate can only see the state of the proposal. It can be on reviewing, approved or denied. Both approved and denied states trigger

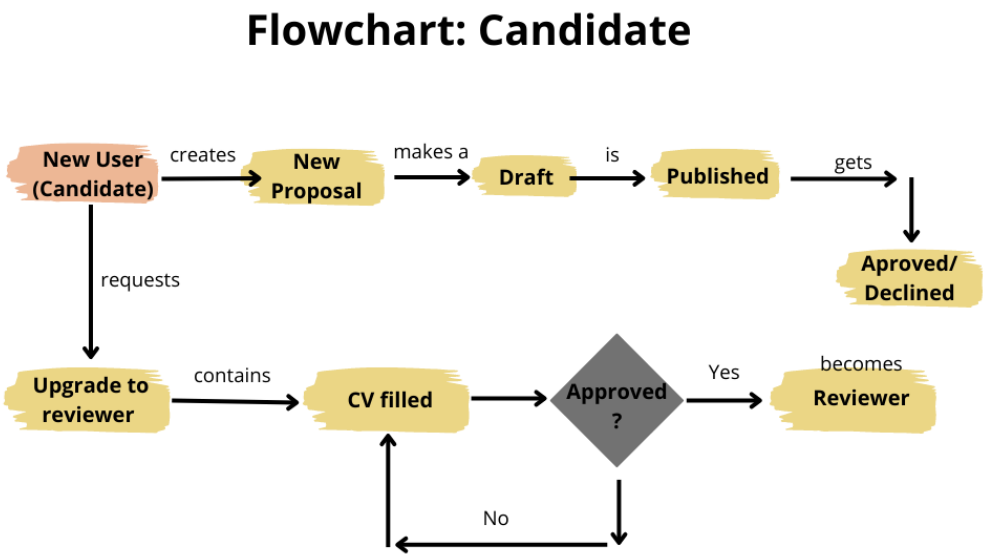


Figure 3.3: Candidate Skateholders and their interactions

an email as notification on the results of the call. The candidates cannot see the other proposals for the same call or the winners of the call.

The second action the candidate has access to is to upgrade the account to a reviewer. The process to became a evaluator is to first manifest the interest by choosing that feature on their account, second fill a curriculum with previous work done in sectors or areas of interest. This should be filled for past work only. Once the curriculum is submitted, the candidate will know the application is on evaluation and when the administrator has made a decision and then they will receive a notification. If it is accepted, the candidate becomes a reviewer automatically. If it is declined the candidate stays as candidates although they have the possibility to make a new request to became a reviewer whenever they want. For last, if the candidate is also a reviewer, the candidate will not be able to review the call in which they made a proposal for.

3.1.1 Proposal

Proposals are the "answers" for the calls. Candidates can create proposals for a call when they have a project valuable to the call that can offer a solution to the problem proposed. They must propose a solution for the problem requested and explain in a detailed form how would they achieve the proposed. Every call can have multiples proposals associated and the same candidate can submit more than one proposal to the same call.

The proposal is always identifiable with, for example, a unique number associated with it. The title and the abstract are also mandatory information about the call as well as the time duration needed to implement the solution proposed.

It is important that is clear which call is the proposal for and, of course, which candidate created the proposal. The company that is proposing to implement the proposal is the company of the candidate. One of the most important features that a proposal needs is the option to upload files. Although the portal must have the basic information of the proposal, the detailed information on "what" and "how" should be submitted through files.

For last, it is important to be clear in which phase the proposal is in, as each state defines who can see the proposal, if it is editable and if is ready to be reviewed. Each time the candidate decides to create a proposal there is a structure that every proposal follows as it flows through each state. The candidate always has access to which state the proposal is (Figure 3.4).

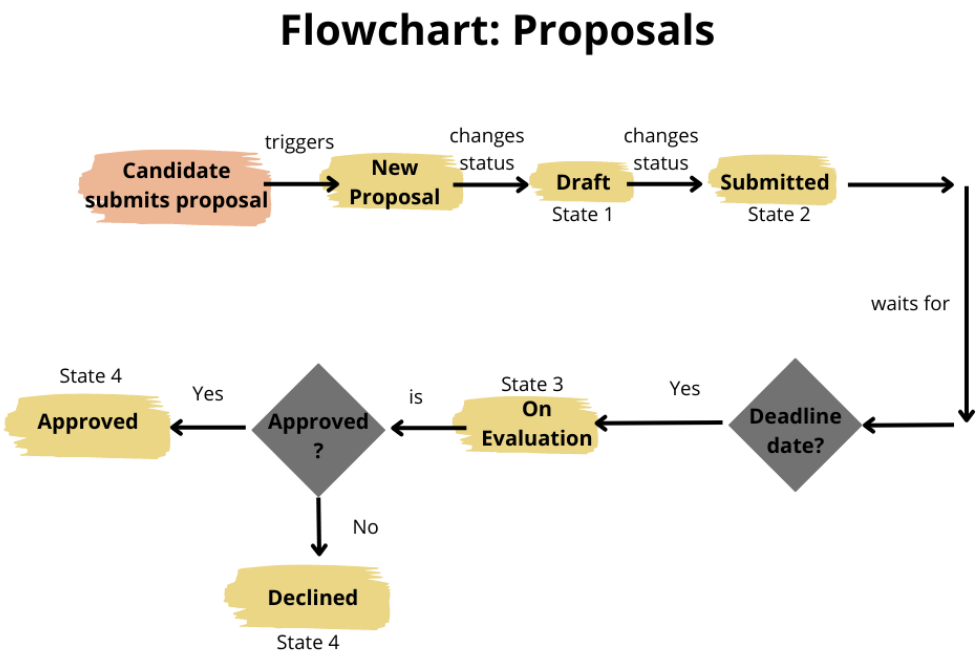


Figure 3.4: Flow of proposals

1. **State 1 - Draft:** The first state when a proposal is created. The candidate can edit every item, like title and add or delete files. It is possible to save a draft and comeback later (it must have a "save"button for each time the candidate wants to leave the page) to change or add components. The only mandatory information is the ID of the candidate that created the ID of the call which the proposal is for. This phase is over when the candidate has finished completely the proposal and presses "Submit"(the button only is available when all the verifications mentioned before are made). At this stage the proposal is not public and the only person who can have access to it is the creator.

2. **State 2 - Submitted:** When the button "Submit"is pressed, the proposal becomes not editable and all the data is saved. At this stage the administrator of the call can see the proposal but cannot edit it. It will not be visible to other candidates or administrators.

3. **State 3 - On evaluation:** This stage starts when the call which the proposal is for closes. In this stage the reviewers will read the proposal, create a report and grade the proposal.

4. **State 4 - Approved:** If the proposal is the winner of the Call, it will enter the stage 4, where the candidate will be notified. The work after the winner is chosen will happen outside of the portal, directly between the administrator (creator of the call) and the candidate (creator of the proposal). A call can have multiple proposals accepted and the evaluation necessary to be accepted is defined by the creator of the call.

5. **State 5 - Declined:** If the proposal is does not win the call, it will enter the state 5 and be declined. The candidate will be notified and the proposal will be archived.

3.2 Administrator

The administrator is the most powerful user. This kind of user resembles a manager as they have the power to manager all reviewers (upgrade candidates to reviewers and downgrade reviewers to candidates), see all calls, whether they are "set", "published", "closed"or "archived"(the exception is the calls on the state "draft") and can create them. For the same reasoning as calls, the administrator can see all evaluation forms and can create and edit the ones created by them. In order to have a overview of the progress and success of the portal, the user has the ability to see all the proposals submitted to each call and see which reviewers are allocated to each proposal.

The only possibility for a user to become a administrator is if one current administrator upgrades a user to that position. This process doesn't involve a request or acceptance as it is assumed that these actions are handled outside of the portal.

When a administrator logs in to the portal there are mainly two actions that can be made: Approve or decline reviewers and manage calls.

When it comes to calls, the administrator creates a call when is faced with a problem in a specific area and there is a desire to find one or multiple solutions. With this in mind, the first task is to create a call. Each time a user creates a call, has to create automatically a evaluation form. The evaluation form will be used by reviewers as a guide on which parameters the admin wants the proposals to be evaluated by. Once the the call is open, the administrator can see when a proposal is submitted and once the call is closed, when it is clear how many proposals were submitted and consequentially how many reviewers are needed, the administrator must choose the reviewers of the call.

This process is assisted by the portal as the platform suggests the best reviewers for a call based on their past experience on sectors and areas of interest. The administrator invites the reviewers to work on the call and the reviewers can accept or not the request. Either way the administrator will receive that information and if necessary will invite new reviewers. The administrator has also the possibility to create a pre-evaluation filter that discards some proposals automatically. This filter can check conditions like having the necessary documents or some areas of interest or sectors that are not allowed for that specific call. Each time a reviewer submits a evaluation, the creator of the call in which the proposal of the evaluation is notified as requested to validate the evaluation. In this phase the administrator can request changes and make notes for the reviewer change on the evaluation. If no alterations are needed, the evaluation is submitted.

Once all the evaluations are submitted the portal will notify the administrator with the results and declare the winner or winners.

The second responsibility of the administrator is accepting and declining upgrades to reviewer requests. This is a simple process as the portal notifies the administrators when a new request enters the platform and every admin is able to see the curriculum of the reviewer and evaluate if candidate as the capacity to become a reviewer or not. The flow of actions of the administrator can be observed on Figure 3.5.

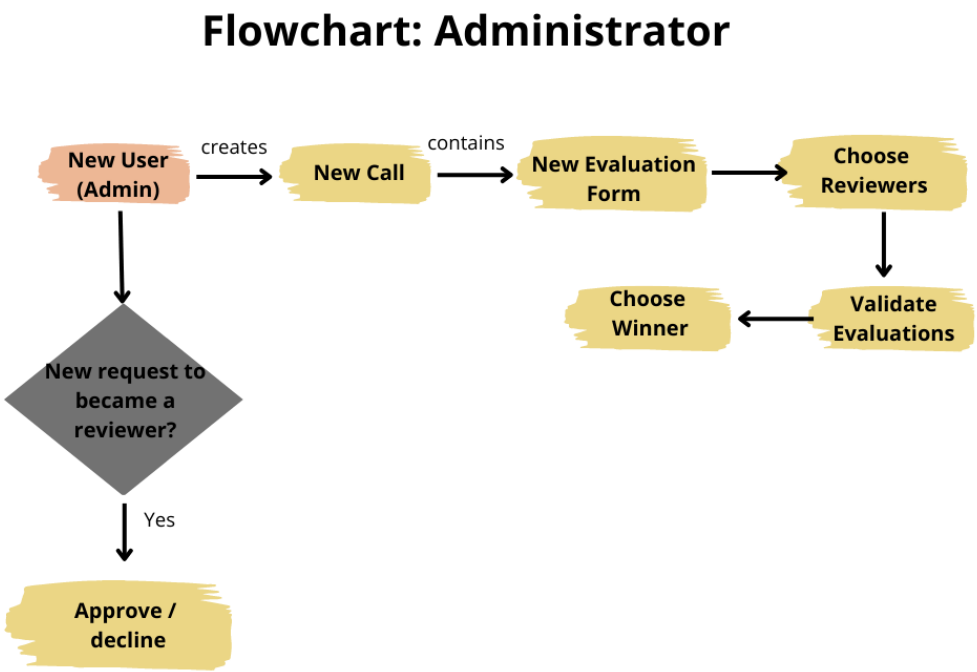


Figure 3.5: Administrator Stakeholders Roles and their interactions

3.2.1 Call

A Call is a contest created by the administrator when a company needs some kind of technological solution for a problem and needs to outsource the creation and implementation of it. This object on the database must have all the information about the requirements and goals of the call, which every proposals must solve. The information that the database must have about every call is:

1. Title of the project;
2. Acronym;
3. Company which created the call;
4. User who created the call;
5. Date when the call is open for proposal submissions;
6. Date when the call closes for proposal submissions;
7. Current state of the call (will be described below);
8. Sectors the call is inserted in;
9. Areas of interest the call most likely will require (tools or areas of technology)

From the creation of the call to its closing, there are several phases that the call must go through. This states are described below:

1. **State 1 - Draft:** The first state when a call is created. The administrator can edit every item, like title, acronym, the deadline, the opening date and add or delete files. It is possible to save a draft and comeback later (it must have a "save"button for each time the administrator wants to leave the page) to change or add components. The only mandatory information is the ID of the administrator that created the call and consequently the company which the company is for. This phase must trigger automatically a evaluation form and is over when the administrator has finished completely the call and the evaluation form and presses "Publish"(the button only is available when all the verifications mentioned before are made). At this stage the call is not public and the only person who can have access to it is the creator.
2. **State 2 - Set:** When the button "Publish"is pressed, the call becomes not editable and all the data is saved.
3. **State 3 - Published:** When the opening date arrives, the state Published starts. The call is public and all users of the portal have access to view the call. It's appears on the call list and candidates can create a proposal and submit it in this stage.

4. **State 4 - Closed:** This state is starts when the deadline date has passed and this is the state where proposals are no longer being admitted and the administrator must choose the reviewers to evaluate the proposals. The platform informs how many reviewers will be needed based on the number of proposals and will suggest the reviewers more appropriate to the call. After the administrator chooses, the reviewers will receive an email or notification that they were invited to review a number of proposals for a call. The reviewer can accept or deny the offer and the administrator will be notified. Once all necessary reviewers have been recruited the state 5 will start.

5. **State 5 - On Evaluation:** This state is when the reviewers are creating evaluations for each proposal. The reviewer must read the proposal, create a report based on the evaluation form and grade the proposal. When the reviewer finishes the evaluation, the admin will validate it. In case the administrator is not in agreement with the reviewer, they will ask to make the adjustments necessary. After all the evaluations are created the winner will be revealed and notified.

6. **State 6 - Archived:** When the winner is defined the call will enter the last stage, archived. In this stage the call will not be visible for anyone except the creator.

It is possible to observe a visual representation of this states on Figure 3.6.

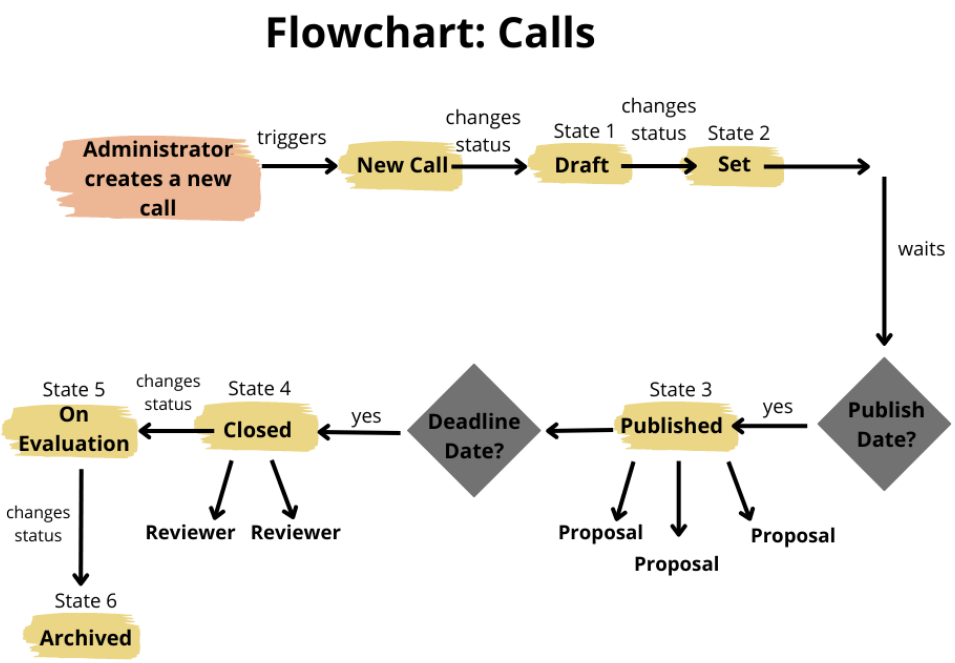


Figure 3.6: Evaluation Stakeholders Roles and their interactions

3.2.2 Evaluation Form

Every reviewer must have access to a template made by the call creator where there is all the questions the reviewer needs to answer regarding every proposal. This is one of the most unique needs for this platform, the possibility to have different layouts for the evaluation form, with specific questions individually chosen for each call and a dynamic number of questions per evaluation form.

The evaluation form serves the purpose mentioned above as it is a template created by the creator of the call, with clear guidelines of the most valuable parameters for the administrator and in which criteria the proposals must be evaluated. The evaluation form will contain the questionnaire that each reviewer must answer about each proposal. Additionally the administrator must define the weight of each question for the total grade of the proposal.

Figure 3.7 is the visual interpretation of the flow every evaluation form goes through.

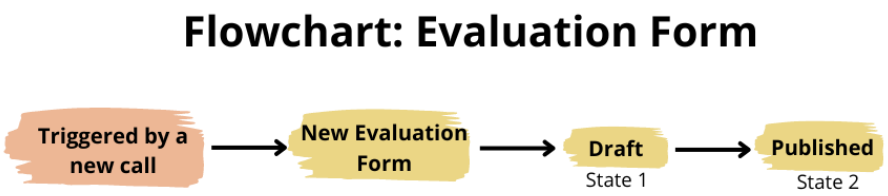


Figure 3.7: Flow of Evaluation form

When considering the evaluation form as an object of an ontology, the parameters that need to be clear are the call for which the evaluation form was created and the list of questions that are contained in it. Each question must have a quotation of the weight of the question for the total grade.

A creation of an evaluation form is triggered by the creation of a call and is composed by two states:

1. **State 1 - Draft:** The first state when the evaluation form is created. Is triggered by the creation of a call. Only the creator of the call can access this form and can edit all components as well as add as many questions as necessary. For each question it will have the filled to write the question as well as the cotation that the question must have. The evaluation form changes stage, when "Published"is pressed, if the evaluation requirements are filled and the sum of the quotations of all questions of the form are 100. Otherwise the button will not appear visible.

2. **State 2 - Submitted:** When is published there are no alterations that can be made, including by the administrator. Additionally the evaluation will be available to the reviewers and administrators.

3.3 Reviewer

The reviewer has the responsibility of grading each proposal that is submitted on the portal. The process to became a reviewer is described on the section 3.1 and it involves a candidate requesting an upgrade and an administrator accepting it based on prior work. It is important to mention that the profile reviewer can still submit proposals for calls. The only restriction is that if the reviewer has submitted a proposal for a call, cannot be the reviewer of any proposal of that call, due to conflict of interests.

The Figure 3.8 demonstrates the actions made by a reviewer. When a call is closed, the administrator will select the best reviewers to evaluate the proposals. When an administrator chooses a reviewer, they receive a request to evaluate proposals of that call and have the choice to accept or decline. If the reviewer declines, they will go back to wait for another request, if the reviewer accepts the request, the portal will automatically select a number of proposals that the reviewer has to evaluate and grade. Additionally they will have access to the evaluation form, which purpose is described on subsection 3.2.2. When an evaluation form is submitted, still has to be validated by the administrator.

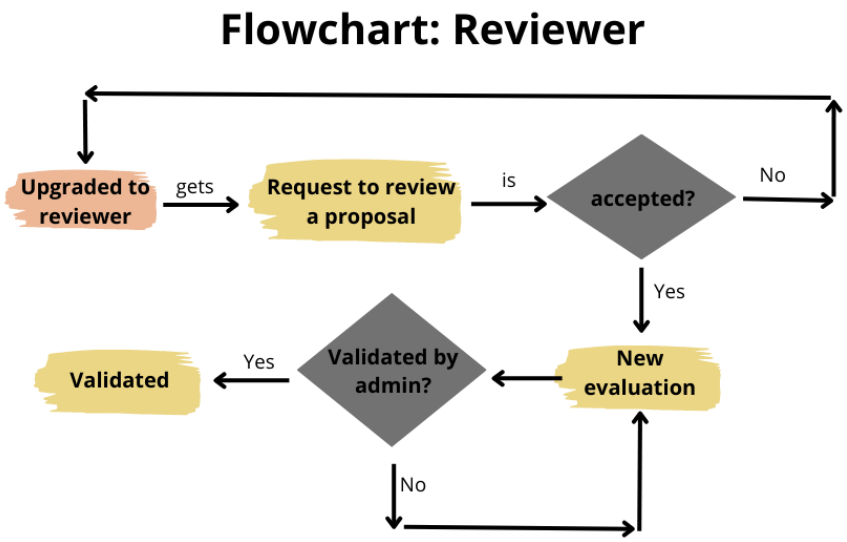


Figure 3.8: Reviewer Stakeholders Roles and their interactions

As the reviewer reviews proposals it is added to their history and is kept in count automatically when the algorithm suggests reviewers.

3.3.1 Evaluation

Each time the reviewer has to evaluate a proposal, they will create an evaluation. This is based on the evaluation form of the call and contains the grade for each question that is

on the evaluation form, as well as the justification of the reviewer for the grade.

An evaluation will contain:

1. The proposal which the evaluation is evaluating
2. The reviewer that created the evaluation
3. The evaluation form which the evaluation is based on
4. The date which the evaluation was created
5. The grade and justifications for each question of the evaluation form
6. A field for extra observations about that proposal that are not included on the evaluation form
7. Final grade of the proposal
8. Current state of the evaluation (described below)

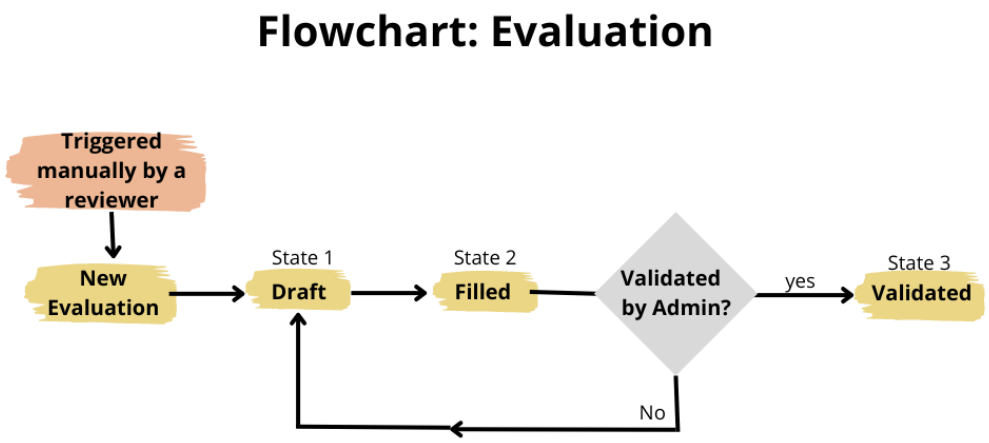


Figure 3.9: Flow of Evaluation

The Figure 3.9 has a visual illustration of the states of a evaluation. There are three states and each has the following definition:

1. **State 1 - Draft:** The first state when the evaluation is created. It is triggered by the reviewer when they want to grade a proposal. The reviewer is able to take multiple days to finish the report and save the answers that are filled from one day to the other. Only the creator of the evaluation can access and can edit the components.
2. **State 2 - Filled:** When the reviewer finishes a evaluation should press the button "send"and the evaluation will enter the state 2 where the creator of the call will

receive a notification to validate or not the evaluation. In case the administrator accepts it, the evaluation will go to the stage 3, otherwise will go back to the state 1.

3. **State 3 - Validated:** The final stage of the evaluation is when triggered by the administrator when the evaluation is validated. In this stage no more alterations are possible, including the grades for each on.

3.4 Auxiliary Objects

There is the need to create some auxiliary objects to create a robust database that includes all necessary information to have a user-friendly platform that is efficient and capable of assisting the choices needed by every user.

3.4.1 Sectors & Areas of interest

How will the reviewers be chosen to review a proposal? It is necessary to structure the database with the necessary information for the administrator to chooses the best reviewers for a call. This decision will be based in two important parameters: Sectors and Areas of Interest. Sectors are the working area in which the solution will be applied. For instance medical, transport or education. Areas of interest refers to different technologies that the proposal contains, for example IoT, big data, AI, etc. Both sectors and areas of interest are objects that can be related to most of the other objects. Calls, which will refer in which sector the call is for and which technology is most likely needed for the request.

Every call can have zero or more than one sector and area of interest. For instance, a call can be a contest to optimize the mobility on the medical sector but don't have an area of interest in which the implementation must be based on (2 sectors and 0 areas of interest). On the other hand, one call can be a solution for the fashion industry but have multiple ares of interest where the solution can be involved. Sectors connected to proposals represents all the areas in which the proposal can be applied.

The areas of interest of the proposal, are the ones which are used to implement it. Both of this don't need to be the same of the call. Users also must be associated with both of this tables, as the information about the background of the reviewer is important in order the algorithm be able to choose the most capable reviewers. There should be known for each reviewer how long and when has the reviewer worked with a sector and area of interest.

3.4.2 Company

The database has to save every company any user is representing. Every proposal must be connected to a user, which will be connected to a company. More than one user can represent the same company but a user can only represent one company. Additionally,

every call has to be connected to a company, and this company doesn't need to necessarily be the same company that the administrator, that is creating the call, is connected to.

It is important to store and relate every user, that is representing the same company, in order to understand if there are any conflicts of interests between candidates, administrators and reviewers. For example it should not be possible for a reviewer to be from the same company as a candidate, because it would not be an impartial evaluation.

A company must be stored on the database with the following information:

1. Name of the company or entity
2. PIC (number given by the European Comission that proves their reliability)
3. Hyperlink of the website of the entity (although this information is not mandatory)

3.5 Technological Specifications

This chapter has the goal of presenting the tools used to implement the desired concept. As the implementation is the creation of a portal, in a website form and has to have a dynamic database where the data about calls and proposals can be stored.

It is necessary to define development tools to create the database, to retrieve and add data from the data base, as known as *backend*, to develop the *frontend* and the interface the user visualizes and interacts.

3.5.1 MySQL

The most crucial tool used in this dissertation is MySQL. The database is 100% developed on MySQL, which supports the whole structure of the portal. It was made a decision to use the Ontology concept on a database and after analysing different options to, it was chosen MySQL for the three main reasons. First it is a relational database management system which means the data is organized in tables and rows creating relationships between them. This enables a flexibility and integrity and manipulation of data, which is an advantage. Another advantage is that MySQL allows large scales of data to be managed in a fast and efficient way. Lastly it is a very user-friendly tool, with a small learning curve which allows to focus more on the good implementation of this dissertation than on acquiring proficiency with the tool.

MySQL is based on queries, so the communication between the client and the database is always trough queries where is possible to add, insert, updated, delete, among others. The database is composed with tables and each table contains columns and rows. When a new record is created is stored in a row and has a field for each column which contains a specific information about each record.

This is the tool that supports fully the database and consequently the whole website. Contains 100% of the data inserted into the website and the connections between the data.

3.5.2 Express

As *backend* it was decided to use the Express tool, a framework from JavaScript. It is a server-side environment that facilitates the creation of APIs (stands for Application Programming Interfaces) which connects the database and the client (*frontend* in this case). The choice to use this tool was based on its simplicity, flexibility and community (has a large amount of developers who use this tool which enable access to more interesting solutions). Overall is a tool capable of creating a robust and reliable web applications.

In this dissertation the tool Express is used to create an API with routes to connect MySQL with JavaScript.

3.5.3 React

React is a widely known JavaScript Library specialized in interface developing. It highly flexible, can easily incorporate other libraries and frameworks and it uses virtual Document Object Model (DOM), which enables an efficient rendering.

This is the tool chosen as *frontend* for the platform, meaning the whole interface with the user is developed on React.

3.5.4 Next

The last tool used is Next.js, which is a framework of React used to optimize the development of web applications. It is considered on of the best tools for websites with dynamic content as it uses an rendering on the server, only sending to the browser an HTML file.

4

Implementation & Validation

4.1 Database

In order to create this website successfully, is crucial to have storage with all the information inserted. The database has that goal and is one of the main objectives of this dissertation.

The database was implemented with the tool MySQL and consisted of first defining our objects (that became our tables), the necessary information about each object (columns) and the relationships between them.

4.1.1 Objects

The objects are the main structures of a MySQL database and in that sense, defining them, must the the first step into creating a database.

There are three main concepts that would have to be objects, under any circumstance: users, calls and proposals. It was decided to create only one table to all types of users, as the majority of the information is mutual for each type. It was created a "calls" and a "proposal" table to store all the information about this two concepts.

The evaluation is one of the most complex structures of this database as it is highly dynamic and editable for each call. It was created an "evaluation_form" table to store the parameters the creator of the call values and wants the proposals to be evaluated on. It was created an auxiliary table, "question_form", due to each evaluation form has the possibility to have a different number of questions, with different quotations. Finally, it was created the "evaluation" and "question"tables for the evaluation of each proposal be stored. As the "question_form"table, the "question" table was created to accommodate different numbers of questions.

In order to connect users, calls and proposals of the same company, it was decided to create a table for companies so the data would be centralized and each time a user need to introduce data about a company that would already be stored.

To structure all the options for areas of interest and sector, it was chosen to save them in a simple table where you have an id and the name of the sector and area of interest.

The idea is that this tables are almost static, as the objects must be constant and the same the majority of the time and must be mostly used to connect with another tables like calls, proposals and users.

4.1.2 Variables of objects

Each table has a list of information that needs to be stored about it. This are the columns of a table. When creating the columns there were some precautions to be taken.

First, every table need an unique parameter to identify the row of the table. For this reason it was created an ID column for each table. Besides that, columns can have different characteristics, that can be a helpful tool to create a more robust database. For instance, a column can have a default value, that was used for situations like the state of an object, where the first state when a row is created is always zero. A column can be mandatory or not which made possible the feature drafts, as some columns can be editable or filled after the creation. The auto-increment tools, allows a column of a table to increase one number when a new row is created.

To create the tables and respective columns are used queries. For instance, to create the table "proposal"it was used the following query:

```
    CREATE TABLE ‘proposal‘ (
‘proposal_id‘ int unsigned NOT NULL AUTO_INCREMENT,
‘proposal_name‘ varchar(35) DEFAULT NULL,
‘abstract‘ varchar(300) DEFAULT NULL,
‘acronym‘ varchar(15) DEFAULT NULL,
‘duration_days‘ int unsigned DEFAULT NULL,
‘state‘ tinyint unsigned NOT NULL DEFAULT ’0’,
‘call_id‘ int unsigned NOT NULL,
‘files‘ varchar(300) DEFAULT NULL,
‘user_id‘ int unsigned NOT NULL,
‘duration_days_aux‘ int unsigned DEFAULT NULL,
PRIMARY KEY (‘proposal_id‘),
FOREIGN KEY (‘call_id‘) REFERENCES ‘calls‘ (‘call_id‘),
FOREIGN KEY (‘user_id‘) REFERENCES ‘users‘ (‘user_id‘) );
```

4.1.3 Relationships between objects

In order to have a functional database, there was a need to prepare and draft very well the relationships between every table. There are three types of possible ways to relate two tables in MySQL:

1. 1:1
2. 1:many

3. many:many

The 1:1 relationships means that only one row of the table can be related to one and only row on another table. One example for this is a finger print and an ID number. There is only one finger print that corresponds to a specific ID number. And the same happens for the opposite, there is only one ID number for each finger print. The way this relationships are integrated on MySQL is making the primary key of one table the foreign key of the second table, as a primary key has to be unique.

The 1:many relationship is a object can be related more than one time to another table but the opposite cannot happen. For example one woman can have multiple children, but each child can only have one mother, which makes this a relation 1 to many. MySQL makes this connection by creating a foreign key with a column from another table. This foreign key can be a primary key or a normal column of the other table.

Finally, there is a many to many relationship. This is where multiple objects of a column can be related to many objects of a column from another table. For instance one recipe can have multiple ingredients and one ingredient can be used for multiple recipes. This is the most complicated case to connect on MySQL, where is needed to create an auxiliary table with two foreign keys from the tables that should be connected.

Every relationships between tables were analysed and classified:

1. Every user, proposal and call can have multiple sectors and multiple areas of interest, and all sectors and areas of interest can have multiple users, proposals and calls (many:many)

2. Both proposals and calls can only have one creator (users). But users can have multiples calls and proposals (1:many).

3. Each proposal is only submitted for one call, but a call has many proposals associated (1:many).

4. Call has the respective evaluation form connected, and the evaluation form has an unique call(1:1).

5. Evaluation form is constituted by multiple questions forms, but it is a 1:many relationship because each question form only has one evaluation form. Similar to this the evaluation has multiple questions form, but each question only has one evaluation. On the other hand, each evaluation is connected to one evaluation form, but an evaluation form is connected to multiples evaluations (1:many).

6. Lastly, each evaluation is connected to a proposal and proposals have multiples evaluations (1:many)

On Figure 4.1 is the diagram of the final database on MySQL. It is possible to observe every table and their columns and the relationships between them.

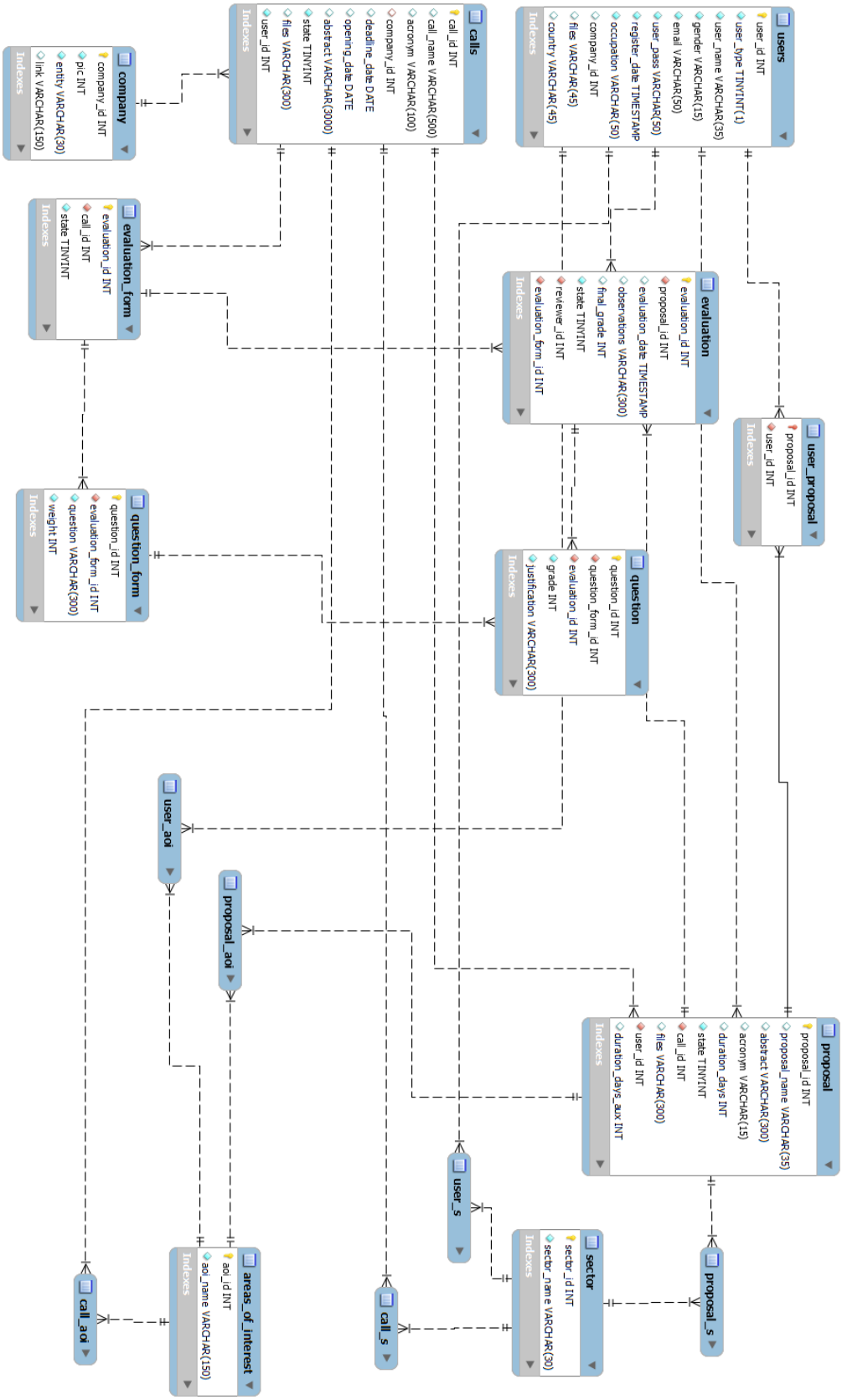


Figure 4.1: ER Diagram

4.2 User Experience

The second hypothesis of this dissertation is to create a easy, user friendly experience for all types of users to navigate the website. This chapter aims to demonstrate the implementation of the website in all the features implemented.

As this platform is a prototype and it doesn't need to go live it was decided to create the platform on a local host (port 3000) and it was created an API was 8080 (with 6 routes). Every connection with the database was executed through express with post or get requests. It was implemented a simple style to be cohesive and ease the users to focus on the important part

4.2.1 Register & Login

When a user enters the website, is greeted with an home page (demonstrated on Figure 4.2) where the user can register and create an account or, in case the user already has an account, can login into their account.

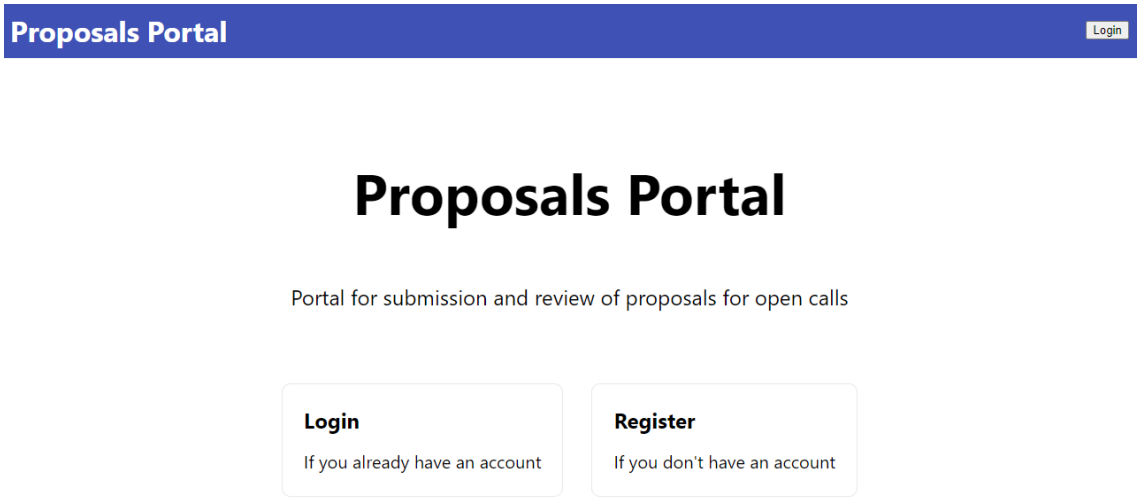


Figure 4.2: Home page

If the user doesn't have an account, will go to the register page (Figure 4.3) where they will have to fill their name, choose a gender (feminine, masculine, other), email (has to be unique), password, occupation and country. When selecting the "Create account"button, through events, the data filled in every field is sent to an API that connects with the database on MySQL and makes a query to create a new user row and new company row (if necessary). As referred before, when an account has been created, it will be created as a candidate.

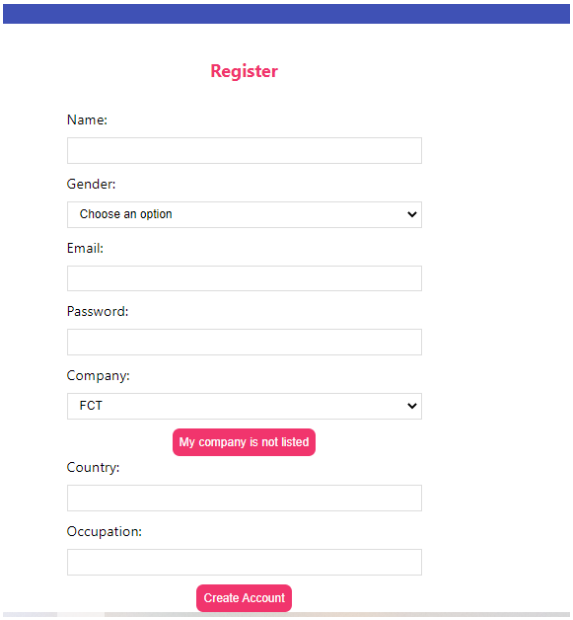


Figure 4.3: Registration

Additionally there is a dropdown menu where they must select the company/entity they are representing. In case this last parameter is not on the list (Figure 4.4), the user can insert a new company into the data base through the "my company is not listed" button. This button will open a pop up page to insert the data regarding the company (Figure 4.5). After adding the entity, if they go back to the register page, the new entity will be listed on the dropdown menu.

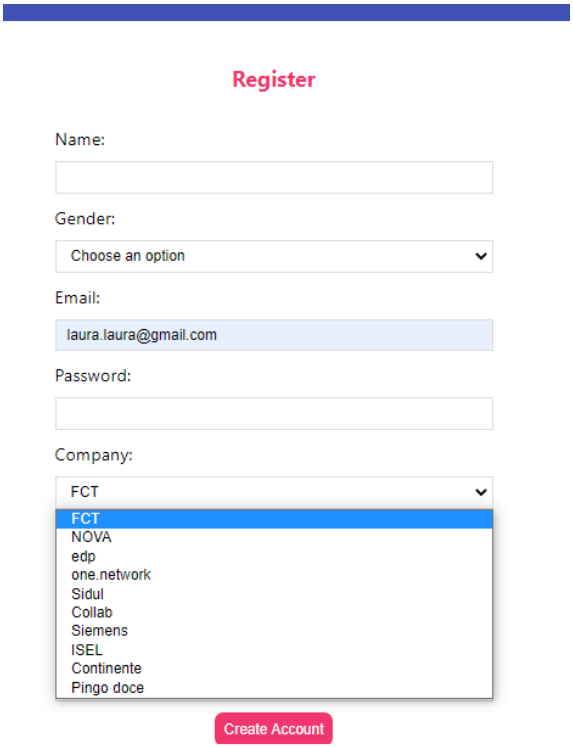


Figure 4.4: Companies on registration

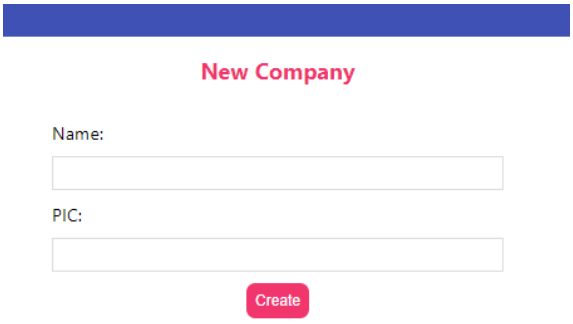


Figure 4.5: New Company

When the account has been created, the candidate can log in simply with their email and password. This will communicate with the database and check if there is one row in the "users"table with the email and password entered. After being logged in, there is a button on the right top page that allows the user to log out at any moment. In that case, the website will go back to home page, to login or register a new account.

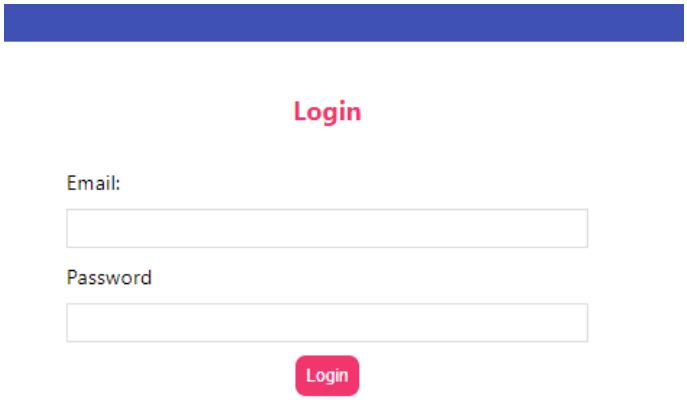


Figure 4.6: Login

In case the email and password match, the portal will direct the user to their profile (Figure 4.7). The method used for session is a simple method using the local storage variables which saves the id of the user on a global variable.

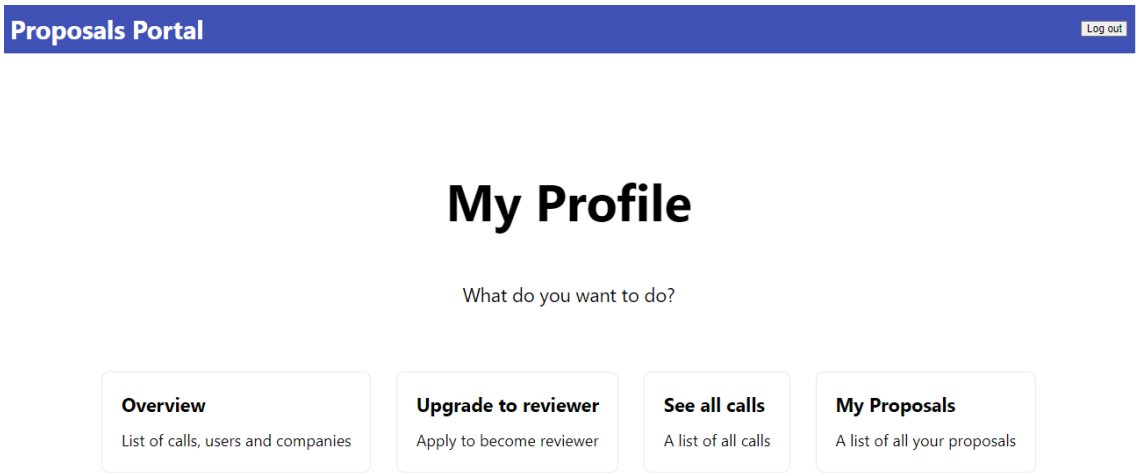


Figure 4.7: Profile of a candidate

Inside the profile, the candidate can choose what is the goal of their visit on the portal. It is possible to search for calls to see if they are eligible to one, on the "See all calls" button; they can see each proposal they have created ("My Proposals"), see their state and edit the proposals that are still in the draft state. Lastly they can request an update to become a reviewer ("Upgrade to reviewer").

4.2.2 Creating a Call

If the session logged in is an administrator, on the profile it is possible to create a new call. It is necessary to insert the title of the call, its acronym, abstract, the day which the call will be open and proposals are admitted and finally, the date which the call stops

allowing proposals to be submitted. Once the "Publish" button is pressed, the *frontend* will send the inserted information into a query to MySQL which will create a row on the table calls.

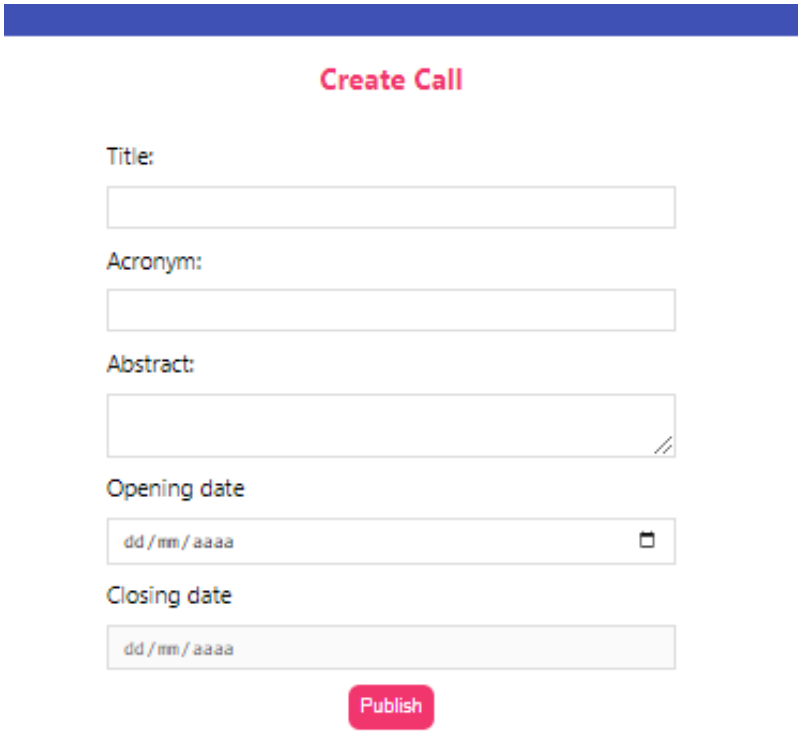


Figure 4.8: Create Call page

4.2.3 Search for calls

A user has always the option to see all calls, including calls published that aren't open, open calls or even closed and archived calls. The calls are listed with the title, acronym, abstract, the date of when the call is open and the deadline of admitting proposals (Figure 4.9).

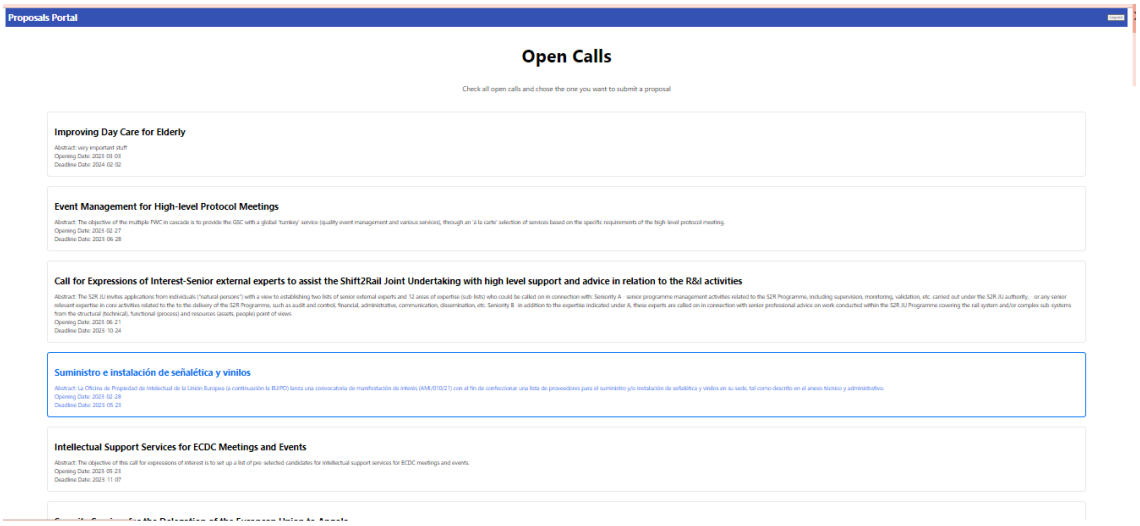


Figure 4.9: Available Calls

Each call is a button that redirects the user to see the details of the call and the option to submit a proposal for the mentioned call, like it can be observed on Figure 4.10. It contains the name of the creator of the call, the company to which the company is for and the sectors and areas of interest. At the end, in case the call is open, there is the option for the user logged in to create a proposal for this call.

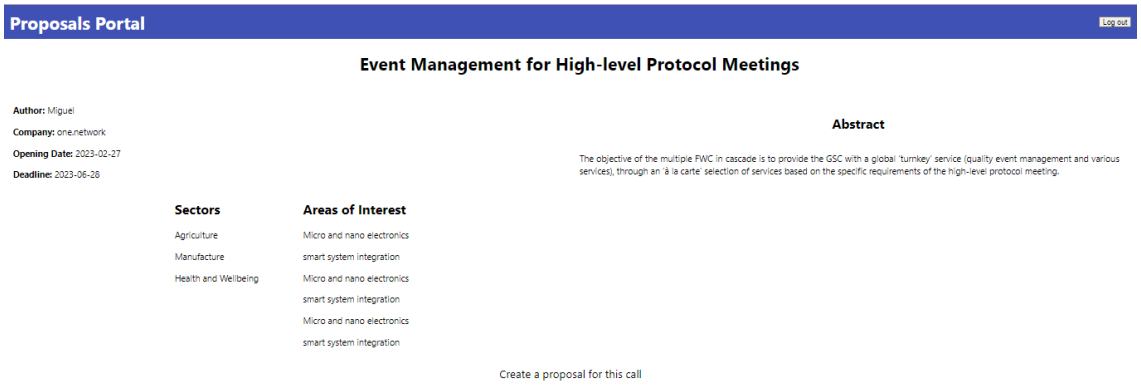


Figure 4.10: Available Calls

4.2.4 Creating a proposal

In case the candidate wishes to create a proposal for a call, on the page off the call, in the bottom there will be an option to "Create a proposal for this call" in which it will redirect the user to a page of a proposal for that call. This proposal from will automatically have the ID of the call and the ID of the user s those are parameters that don't have to filled by the user. Besides that, the candidate has to create a name for the proposal, an abstract, acronym and the duration of the implementation of the proposal

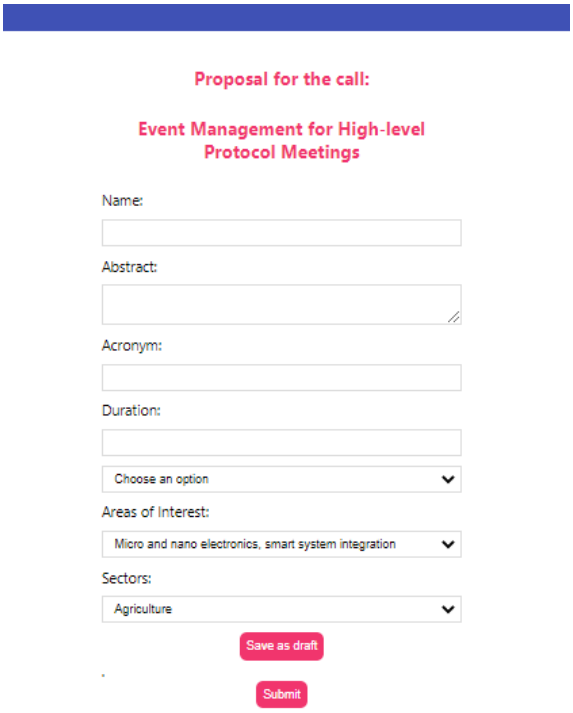
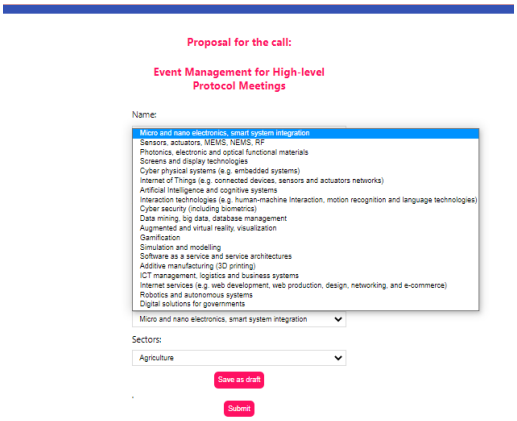
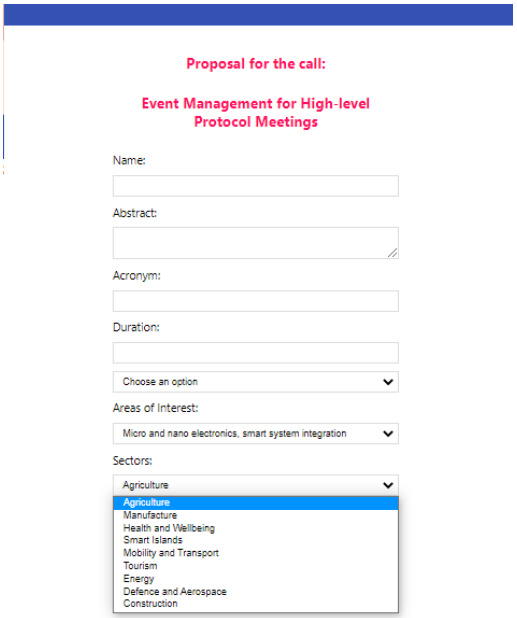


Figure 4.11: New Proposal

Additionally, as can be observed on Figure 4.12, there are two more dropdown menus, one for the candidate to select which sectors is the proposal inserted into and another one for the candidate to select which areas of interest are used to implement the proposal. This is achieved by making a query that returns all the names of sectors and areas of interest.



(a) Possible areas of interest on a new Proposal



(b) Possible sectors on a new Proposal

Figure 4.12: New Proposal

4.3 Tests & Requirements

This section aims to collect the information on the chapters before and see an overview of the goals of this thesis and the results obtained. On the Figure 4.13 it is listed the actions required to have a functional portal of calls, proposals and evaluation of proposals.

	Database	Portal
Create an account	✓	✓
Login/ logout	✓	✓
Create a call	✓	✓
Create an evaluation form	✓	✗
See all calls	✓	✓
Create a proposal for a call	✓	✓
Upgrade to reviewer	✓	✓
See all your proposals	✓	✗
Select reviewers	✓	✗
Create an evaluation	✓	✗
Select winner of a call	✓	✗

Figure 4.13: Actions proposed and results

Overall the requirements had a good outcome, having most of them been completed. When it comes to the database, all requirements were made. It is possible to create an account, to login and logout, to create a call, create an evaluation form, get information about all calls, create a proposal for each call. Both calls and proposals can be connected to sectors and areas of interest on the database. It is also possible to alter a candidate to a reviewer, see their proposals, the administrator can select reviewers for evaluations and can select the winner of a call.

The implementation of the portal had a less positive outcome although it was still implemented six out of eleven requirements. It is possible on the portal to create an account, to login and logout, to create a call, to list all calls, create a proposal for a call and it is possible to upgrade to reviewer.

4.4 Research Questions & Hypothesis Validation

This chapter answers the initial prepositions, the research questions and this dissertation hypothesis. Contemplates the outcome of the proposed and documents how was the result affirmed, positive or negative.

4.4.1 Research Questions

In the beginning of this dissertation it was created one research question that guided the work carried out in this thesis which has the goal of answering it. This question was:

> "Is it possible to represent the data collected in an open call into categories, properties and relationships between different concepts and entities to guide and simplify users tasks and decisions?"

When it comes to the database, there are parameters needed in order to support this platform. It was possible to create a complex database that stores all information necessary to run a platform of calls, proposals and evaluation of the proposals. All data regarding calls is organized and saved, as well as the information regarding each type of user, the proposals created and the evaluations created. It was possible to create a system that connects different calls through similarities, like sectors and areas of interest. This connection was also made with proposals. It was successful creating the states for each object, in order to be clear the different phases of this process and to certain actions be only available with a specific state. With the states feature, the database holds an important opportunity to create drafts for each call, proposal or evaluation.

One of the most unique necessities of this portal was to have custom evaluations for each call, where the creator of the call decides what and how many are the parameters evaluated for each proposal and what weight has each of them. The database was successful in this sense, creating an evaluation form with an auxiliary table of question form, which allowed the flexibility necessary to create a different number of questions for each evaluation.

Additionally, this database was structured in a flexible way, where it is possible to add new tables, columns or restrictions. This is especially important in a platform that hasn't been completely implemented because it allows for the database to adapt to new features desired.

Lastly, it is important that the database has the necessary structure to be possible to choose reviewers for each call, having in account the work done on by the reviewers on the portal. The database allows this as it saves which area of interest or sector have the reviewers worked on before and the areas of interest and sectors calls had that the reviewer has evaluated. This enables the possibility to create a strong and efficient algorithm to choose reviewers.

With the previous arguments, it is justified that the research question is answered, with a positive outcome.

4.4.2 Hypothesis

The hypothesis that must be validated or not of this dissertation is:

> "It is possible to create a portal for submissions of proposals, in a digital way, where all users have a fast, simple and easy experience navigating it by organizing and structuring the data base?"

As mentioned on the subsection 4.4.1, it was possible to create a complex database that allows the portal to have complete functionality, and create numerous features, from drafts to features.

With the implementation of this database and portal, the administrator have their job significantly simplified as they can create a call in an easy way, with structures defined, in the same place can create the structure that the reviewers must follow to evaluate their proposals, as well as they are able to choose the reviewers, all in one place. The candidate can see the Open calls and the call that will be open in the future. The can directly create a proposal for the call in just two clicks. Additionally, candidates can see the state of the calls they have submitted a proposal at any moment.

One of the best features that was possible to create with this portal was a flow that allows users to have always accessible their next action on the page they are. This optimizes the user's experience as they don't need to loose time going back to other pages or even looking for them.

With the information that the database was successful and the portal was efficient and user-friendly, is is possible to confirm the hypothesis of this thesis.

5

Final considerations & Future Work

This chapter aims to document the final considerations of this dissertation, compare the goals and objectives with the project implemented and its functionalities.

It is also considered the possible work that can continue this platform and ways to improve it.

5.1 Conclusion

The motivation behind this project was to improve and optimize the experience of creating calls, submitting proposals and evaluating the proposals. Before implementing, it was studied on chapter 2 what platforms exist and which are the advantages and disadvantages of each one. It was found a discrepancy between the level technology that exists right now to the process to create a call, submit proposals for it and choose the right reviewers. With this in mind, there was a motivation to decrease this gap and improve this system.

As referred on chapter 1, the goal of this dissertation is to understand if it is possible to create a database that can be used on a portal of proposals for calls where all the process for the portal is on the same place, including calls, proposals, selection of reviewers and evaluation of proposals.

Additionally it was aimed to evaluate whether it's feasible to enhance the experience of the users with a user-friendly portal with the least amount of work possible for each action and all the tasks are organized on the same place.

The concept and structure of the database and portal was designed on chapter 3 and it was concluded, after the implementation, on chapter 4 that it is possible to create a database that structures the desired portal and enables the important features, like drafts, association with sectors and areas of interest, the possibility to customize each evaluation for a call, select reviewers and winners of calls. Additionally when

For these reasons, after considering the work implemented it is considered that this dissertation was successful in testing and validating the hypothesis proposed.

5.2 Future Work

Although the final result of this dissertation is successful, there are a few improvements that could be made.

First, the security of the data. Especially when dealing with personal information it is very important to have a secure and robust security system. For the validation of the hypothesis of this dissertation it was not necessary to create a secure platform, so it wasn't a priority on the work developed but for future use of the platform it would be mandatory to improve this area. This feature would require an evaluation of GDPR because this database contains sensible data.

There are a few missing details that could be defined on this dissertation. Like what happens in a tie. It would be very interesting to define the criteria to untie two proposals. For example, it could be the rate of woman working in the proposal or if the proposal helps underdeveloped countries or not. Additionally it can be decided how many proposals can a reviewer evaluate at the same time and if they have the possibility to refuse a proposal at any state of the call.

Another area which was not a priority for this thesis but would improve a lot the experience of using the platform would an algorithm that was mentioned through the chapter chapter 4 and chapter 3 which has the goal of suggesting the best reviewers to evaluate proposals, optimizing the work of administrators. Although this was not implemented on the website prototype, the database is structured to hold this algorithm and has the data necessary to accomplish the desired goal.

Finally, it would make sense to have two features that have not been implemented, drafts and filters. It would be helpful for all users to save drafts of calls, proposals or evaluations because this can easily be long and time consuming documents that can take more that one go to write. Also to optimize the candidate experience, it would be helpful to choose filters to search calls, for instance sector or areas of interest. The database was made for both of this features as the objects have states in which is known if the document is still a draft or published and the sectors and areas of interest are connected to proposals and calls.

Bibliography

[1] D. M. Procter. "The Academic Proposal". In: *University of Toronto Available: http://www. writing. utoronto. ca/advice/specific-types-of-writing/academic-proposal* (2009) (cit. on pp. 1, 6).

[2] D. Jones and S. Behrens. "Online assignment submission: an evolutionary tale". In: *Proceedings of HICSS*. Vol. 36. 2003 (cit. on p. 1).

[3] S. Shadlou, C. K. Pong, and S. Sukumaran. "Proposal Submission System-A Content Management System Approach for Proposal Submission". In: *International Journal of Web & Semantic Technology* 2.2 (2011) (cit. on pp. 1, 7).

[4] G. Antoniou and F. Van Harmelen. *A semantic web primer*. MIT press, 2004 (cit. on p. 1).

[5] J. W. John W. Chinneck. *How to organize your thesis*. Sept. 1999. URL: http://www.sce.carleton.ca/faculty/chinneck/thesis.html (cit. on p. 3).

[6] L. M. Camarinha-Matos. *Scientific research methodologies and techniques*. 2009 (cit. on pp. 3–5).

[7] E. Commission. *European Commission- Funding & tender opportunities*. 2018. URL: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/home (visited on 05/15/2022) (cit. on p. 7).

[8] F. Mauri and A. Bourrouet. *Europa Web Guide*. 2019. URL: https://wikis.ec.europa.eu/display/WEBGUIDE/Europa+Web+Guide (visited on 06/07/2022) (cit. on pp. 7, 8).

[9] I. Bursuk et al. "Online cofounder search in tech startups". In: *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*. 2016, pp. 482–488 (cit. on p. 12).

[10] *F6s connects 4 million founders and startups to funding, Jobs & Free Hosting deals*. URL: https://www.f6s.com/ (cit. on pp. 12, 13).

[11] *MarTera call 2021*. URL: https://www.martera.eu/submission/call4 (cit. on p. 15).

[12] *Products and Tools*. URL: https://carsa.es/productos/?lang=en (cit. on p. 15).

[13] R. Poli, M. Healy, and A. Kameas. *Theory and applications of ontology: Computer applications*. Springer, 2010 (cit. on p. 16).

[14] Y. C. Cardoso. "Creation and extension of ontologies for describing communications in the context of organizations". PhD thesis. Universidade NOVA de Lisboa (Portugal), 2010 (cit. on pp. 16, 17, 19, 20).

[15] R. N. S. C. Gomes. "An ontology-based representation of an agent-based controlled robotic cell". PhD thesis. Faculdade de Ciências e Tecnologia, 2010 (cit. on p. 16).

[16] O. Corcho, M. Fernández-López, and A. Gómez-Pérez. "Methodologies, tools and languages for building ontologies. Where is their meeting point?" In: *Data & knowledge engineering* 46.1 (2003), pp. 41–64 (cit. on pp. 17, 18).

[17] N. Law et al. "A review of ontology development aspects". In: *International Journal of Advanced Computer Science and Applications* 10.7 (2019), pp. 290–298 (cit. on pp. 17, 20).

[18] A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media, 2006 (cit. on p. 17).

[19] M. Donnelly and G. Guizzardi. *Formal Ontology in Information Systems: Proceedings of the Seventh International Conference (FOIS 2012)*. Vol. 239. IOS Press, 2012 (cit. on p. 17).

[20] F. A. G. Cavaco. "Ontologies learn by searching". PhD thesis. FCT-UNL, 2011 (cit. on pp. 18, 21).

[21] S. Banzal. *XML Basics*. Mercury Learning and Information, 2020 (cit. on p. 19).

[22] S. Decker et al. "The semantic web: The roles of XML and RDF". In: *IEEE Internet computing* 4.5 (2000), pp. 63–73 (cit. on p. 19).

[23] V. Maniraj and R. Sivakumar. "Ontology languages-a review". In: *International Journal of Computer Theory and Engineering* 2.6 (2010), p. 887 (cit. on p. 19).

[24] M. Klein. "XML, RDF, and relatives". In: *IEEE Intelligent Systems* 16.2 (2001), pp. 26–28 (cit. on p. 19).

[25] K. Munir and M. S. Anjum. "The use of ontologies for effective knowledge modelling and information retrieval". In: *Applied Computing and Informatics* 14.2 (2018), pp. 116–126 (cit. on pp. 20, 21).

[26] S. Blakowski and P. Brune. "May the Ontologies Be with You! Towards a user-friendly web-based editor for semantic web service description". In: *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE. 2016, pp. 203–210 (cit. on p. 21).

[27] L. Stojanovic and B. Motik. "Ontology Evolution within Ontology Editors." In: *EON*. 2002, pp. 53–62 (cit. on p. 21).

[28] T. Slimani. "Ontology development: A comparing study on tools, languages and formalisms". In: *Indian Journal of Science and Technology* 8.24 (2015), pp. 1–12 (cit. on p. 22).