

# The Design of an Ontology Management Tool

Jacob Persson and Stella Johannsen

DEPARTMENT OF DESIGN SCIENCES  
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY  
2024

MASTER THESIS



---

# The Design of an Ontology Management Tool

---

Stella Johannsen  
st4670jo-s@student.lu.se

Jacob Persson  
ja8481pe-s@student.lu.se

May 30, 2024

Master's thesis work carried out at Pinteg AB.

Supervisors: Joakim Eriksson, joakim.eriksson@design.lth.se  
Per Morau, per.morau@pinteg.se

Examiner: Günter Alce, gunter.alce@design.lth.se

# The Design of an Ontology Management Tool

Copyright ©2024 Stella Johannsen, Jacob Persson

*Published by*

Department Design Sciences  
Faculty of Engineering LTH, Lund University  
P.O Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design MAMM01  
Division: Ergonomics and Aerosol Technology  
Supervisor: Joakim Eriksson  
Co-supervisor: Per Morau, Pinteg  
Examiner: Günter Alce

## Abstract

Semantic modeling aims to create a common understanding from different perspectives. In particular, the goal of semantic data modeling is to create a better understanding of the human, the data, and their relation to computer systems.

This thesis aims to create a management tool for semantic data modeling for Pinteg, a company engaged in data privacy and protection.

The design process of the thesis is divided into four phases, where the first phase creates an understanding of the project and product, and the other three phases iterate through creating, refining, and evaluating the product. These phases applied relevant interaction design theory and handled both paper and digital versions. Through iterative cycles of designing and prototyping, usability testing, and evaluation with the stakeholders, these prototypes evolved into more refined versions. The mid-fi prototypes were developed in Figma, and finally, the hi-fi prototypes were created with the frontend framework React.

The design process accumulated in a final prototype, and the evaluation concluded future development. The results showed that the concept of ontologies was difficult to explain to users with a limited background in semantic data modeling. Even when given a context, there is no natural conceptual connection to the concept of semantics. In the future, the applications' purpose can be taken in different directions. One natural direction is to use the tool as a straightforward ontology manager, similar to existing alternatives, which requires the user to be familiar with the concepts. Another direction could be to make the tool similar to other community-driven platforms. In this case, it could be beneficial to use established features like subscriptions. In either case, version control would be necessary to handle different resource versions deriving from the same resource, and to keep track of history.

**Keywords:** ontology, web application, semantic data model



## Sammanfattning

Semantisk modellering syftar till att skapa en gemensam förståelse för olika perspektiv. Målet med semantisk datamodellering är särskilt att skapa en bättre förståelse för människan, data och deras relation till datasystem.

Dett arbete syftar till att skapa ett hanteringsverktyg för semantisk data modellering för Pinteg, ett företag som arbetar med dataskydd och dataintegritet.

Designprocessen för arbetet är indelad i fyra faser, där den första fasen skapade en förståelse för projektet och produkten, och de andra tre faserna itererade genom att skapa, förfina och utvärdera prototyper. Dessa faser tillämpade relevant interaktionsdesign teori och hanterade både pappers- och digitala versioner. Genom iterativa cykler av design och prototypning, användartester och utvärdering med intressenter utvecklades prototyperna till förfinade versioner. Mid-fi prototyperna utvecklades i Figma, och slutligen skapades hi-fi prototyperna med frontend-ramverket React.

Designprocessen kulminerade i en slutlig prototyp, och utvärderingen drog slutsatser för framtida utveckling. Resultaten visade att konceptet ontologier var svårt att förklara för användare med begränsad bakgrund inom semantisk datamodellering. Även när sammanhang gavs fanns det ingen naturlig konceptuell koppling till begreppet semantik. I framtiden kan applikationens syfte tas i olika riktningar. En naturlig riktning är att använda verktyget som en enkel ontologihanterare, liknande befintliga alternativ, vilket kräver att användaren är bekant med koncepten. En annan riktning skulle kunna vara att göra verktyget liknande andra community-drivna plattformar. I detta fall kan det vara fördelaktigt att använda etablerade funktioner som prenumerationer. I båda fallen skulle versionskontroll vara nödvändig för att hantera olika resursversioner som härstammar från samma koncept och för att hålla reda på historiken.

**Nyckelord:** ontologi, webbapplikation, semantisk data model

# Acknowledgements

---

We would like to thank Joakim Eriksson, our supervisor at LTH, for his guidance throughout our project.

We would also like to thank Per Morau at Pinteg, who contributed to our comprehension of ontology management and provided insights into Pinteg's vision for a more sustainable society.

Lastly, we would like to express our gratitude to all the individuals who participated in the various tests throughout the process. Their positive attitudes and valuable feedback were of great importance in the completion of this thesis.

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Background . . . . .	5
1.1.1	Pinteg . . . . .	6
1.2	Purpose and Research Questions . . . . .	6
1.3	Sustainable Goals . . . . .	6
1.4	Ethic . . . . .	7
1.5	Related Work . . . . .	8
<b>2</b>	<b>Technical Background</b>	<b>9</b>
2.1	Web application . . . . .	9
2.2	Web API . . . . .	10
2.2.1	HTTP . . . . .	10
2.2.2	JSON . . . . .	12
2.3	The API . . . . .	13
<b>3</b>	<b>Theoretical Background</b>	<b>15</b>
3.1	User-Centered Design . . . . .	15
3.2	Usability . . . . .	16
3.3	Design Process . . . . .	17
3.4	Data Gathering . . . . .	18
3.4.1	Interviews . . . . .	18
3.4.2	Think Aloud . . . . .	18
3.4.3	Observation . . . . .	19
3.4.4	System Usability Scale . . . . .	19
3.5	Prototyping . . . . .	20
3.6	Usability Testing . . . . .	20
<b>4</b>	<b>Method</b>	<b>21</b>
4.1	Timeline of Key Activities . . . . .	21

---

<b>5</b>	<b>Phase 1: Conceptual design</b>	<b>24</b>
5.1	State-of-the-Art Analysis . . . . .	24
5.2	Interview . . . . .	25
5.3	Establishing Requirements . . . . .	26
5.3.1	Personas . . . . .	28
5.3.2	Use cases . . . . .	29
5.3.3	Functional Requirements . . . . .	30
<b>6</b>	<b>Phase 2: Lo-Fi prototyping</b>	<b>31</b>
6.1	Design and prototyping . . . . .	31
6.1.1	Search . . . . .	32
6.1.2	Manage . . . . .	33
6.1.3	Collaboration . . . . .	35
6.2	Usability Testing . . . . .	36
6.3	Result and Analysis . . . . .	37
6.3.1	Usability test . . . . .	37
6.3.2	Evaluation with Stakeholders . . . . .	38
<b>7</b>	<b>Phase 3: Mid-Fi Prototyping</b>	<b>39</b>
7.1	Design and prototyping . . . . .	39
7.1.1	Overview . . . . .	39
7.1.2	Search . . . . .	40
7.1.3	Manage . . . . .	42
7.1.4	Collaboration . . . . .	44
7.2	Usability Testing . . . . .	44
7.3	Result and Analysis . . . . .	45
7.3.1	Usability Test . . . . .	45
7.3.2	Evaluation with Stakeholders . . . . .	47
<b>8</b>	<b>Phase 4: Hi-Fi Prototyping</b>	<b>49</b>
8.1	Design and prototyping . . . . .	49
8.1.1	Overview . . . . .	49
8.1.2	Search . . . . .	50
8.1.3	Management . . . . .	51
8.1.4	Collaboration . . . . .	56
8.2	Usability Testing . . . . .	57
8.2.1	Evaluation Questions . . . . .	57
8.2.2	Participants . . . . .	58
8.2.3	Data Gathering . . . . .	58
8.2.4	Procedure . . . . .	62
8.2.5	Result handling . . . . .	63
8.3	Results and Analysis . . . . .	63
8.3.1	Usability Test . . . . .	63
8.3.2	Evaluation with Stakeholders . . . . .	67

---

<b>9 Discussion</b>	<b>68</b>
9.1 Overview of Design Process . . . . .	68
9.2 Result and Analysis . . . . .	69
9.3 Future work . . . . .	71
<b>10 Conclusion</b>	<b>72</b>
References	73
Appendix A Personas	76
Appendix B Terms used in Usability Testing, Phase 4	79

# Chapter 1

## Introduction

---

*This introductory chapter aims to highlight areas of semantic data modeling and data privacy protection. It begins by discussing the relevance of ontologies and proceeds to introduce the company of this study, Pinteg, providing insights into its operations and context. Following this, the purpose, goals, and scope of this thesis work are outlined, delineating the specific objectives and areas of focus.*

### 1.1 Background

Semantics is the study of meaning and relations, connected to humans' interaction with the real world. Semantic modeling aims to create a common understanding from different perspectives [1]. In particular, the goal of semantic data modeling is to create a better understanding of the human, the data, and their relation to computer systems [1].

To capture relationships between entities, a visual representation can be created. A common graphical representation used when describing relations is knowledge graphs, where the nodes represent entities and the edges represent the relation between nodes.

The building blocks in the knowledge graphs have to be defined in an ontology, which is a formal representation of a set of concepts within a domain and the relationships between those concepts. Ontologies provide a structured framework for capturing the meaning and semantics of data elements.

Ontology modeling has seen limited development in recent years. Outdated tools and a focus on machine learning have overshadowed theoretical aspects. Despite this, semantic data modeling remains crucial for many companies. There is a need for modern tools bridging traditional and contemporary approaches. An updated ontology management tool is essential for revitalizing the field and ensuring its continued importance in shaping data organization for future applications [1].

### 1.1.1 Pinteg

This project aims to create a tool for semantic modeling, designed to assist Pinteg, a company engaged in data privacy and protection [2]. Pinteg aims to help organizations and businesses to understand and apply data privacy and protection laws and regulations [2]. With Pinteg, organizations get help with personal data management and now they want to improve their service with a tool that helps domain experts with limited semantic modeling knowledge, to actively contribute to and manage ontologies. This ensures the relevance and coherence of these structures within the broader knowledge graph ecosystem.

For a concrete use case, consider this: A basketball organization in Lund focuses on the development of young talents, and is mainly managed by volunteering parents. Occasionally, they employ coaches for their teams. This requires a payroll process. The people in charge mostly use email to communicate on such matters, and a player in the junior team's parent sacrificed a couple of evenings to create a formal document. At the same time, a handball organization in Malmö is in a similar situation, and similarly, someone takes their time to put together such a document. Now, what if this could be handled through a shared community, where the "Payroll Process" is a concept that both organizations could use for themselves? The concept can in this case be viewed as a template for what a payroll process is. Also, if one of them decides to share their instance of the process, other organizations in the same field could import their process and implement it for themselves. This is the idea of the ontology management tool, to both make life easier for organizations, but also create verified concepts that guarantee data privacy and protection.

## 1.2 Purpose and Research Questions

The primary purpose of this project is to design, build, and test a tool that facilitates the management and maintenance of ontologies, particularly targeting users, domain experts, with no or limited knowledge of semantic modeling. To achieve this purpose, three key research questions were formulated:

1. What specific demands and requirements must be addressed in the development of an ontology management tool?
2. What design choices are considered optimal for such a tool?
3. How does one construct an ontology management tool to ensure comprehension and accurate utilization by users lacking semantic expertise?

## 1.3 Sustainable Goals

Aligned with the United Nations Sustainable Development Goals [3], SDGs, this project contributes to several key objectives, primarily focusing on Goal 9: Industry, Innovation, and Infrastructure, and Goal 17: Partnerships for the Goals. The SDGs can be viewed in figure 1.1. By developing an ontology management tool tailored for domain experts, this initiative promotes sustainable industrialization and fosters innovation. Through efficient data

management facilitated by the tool, organizations can streamline their operations, reduce redundancy, and enhance productivity, thus contributing to economic growth and industrial resilience, Goal 9. Moreover, by fostering partnerships between organizations within and across sectors through the shared community platform, the project promotes collaborative approaches to addressing common challenges, thereby advancing the overarching agenda of sustainable development, Goal 17.



**Figure 1.1:** The 17 sustainable development goals.

Furthermore, by emphasizing data privacy and protection within the tool's framework, the project contributes indirectly to various other SDGs, including Goal 3: Good Health and Well-being, and Goal 16: Peace, Justice, and Strong Institutions, by ensuring the ethical and responsible use of data in support of societal well-being and equitable access to resources.

## 1.4 Ethic

When discussing the ethical aspects of developing an interface for semantic modeling and ontology management, it is important to consider several dimensions of ethics that can impact users, data management, and society at large.

Firstly, it is crucial to ensure that users' privacy is fully respected. Since this tool will handle personal data such as login credentials, organization names, etc., it must be designed to protect this information from unauthorized access.

Secondly, it is important to consider how usability and accessibility affect users. An interface that is difficult to use or requires specialized knowledge to navigate may result in the exclusion of certain user groups, which can be unethical. By prioritizing user-centered design and conducting thorough usability testing, we can ensure that the interface is intuitive and accessible to all users, regardless of their technical skill level.

Thirdly, it is important to consider how this interface may impact society and the broader ecosystem of data management. By promoting openness, collaboration, and standardization within the industry, we can contribute to the ethical use of technology and avoid harmful



consequences such as the creation of monopolies or the manipulation of information for individual or group interests.

## 1.5 Related Work

A literature review was conducted to establish an understanding of ontology management and user-centered design principles. Various databases such as IEEE Xplore and Google Scholar were utilized to gather relevant research articles, conference papers, and books on ontology management, semantic models, and design processes. These databases are commonly used for literature research and are also referenced by LTH's database [4]. Additionally, sources such as textbooks and online articles were consulted to deepen the understanding of design methodologies and usability testing techniques.

The literature review focused on understanding the theoretical foundations of ontologies, including their structure, semantic modeling principles, and internal relationships. With a deeper understanding of ontologies, a better interpretation of the project goals was obtained. This involved reviewing relevant articles in the field.

To further deepen our understanding of ontology management, the literature review also focused on finding literature on ontology management tool analysis and comparisons. There were several interesting conclusions found during the review. Ontology development is mainly an ad-hoc approach [5], and for a user to make the most out of its usage they have to find which tool would work better for the task at hand [5], as there are several viable alternatives [5]. Some tools are niche in the sense that they contain ontology information that is narrow compared to the broader domain. Therefore, some tools are created to centralize information and facilitate lookup [6]. This can be solved in several different ways, and one such example is the Ontology Lookup Service, OLS, where ontology information is gathered from several sources and saved in a single database [6]. The data is then presented in a consistent interface [6]. The articles [5, 6] and the conclusions presented in this section served as a foundation for the state-of-the-art analysis described in section 5.1.

To gain information regarding the design process, theory regarding UCD, usability principles, and data gathering, were studied. These insights were used to understand the importance of iterative design, user feedback incorporation, and usability testing in creating intuitive and effective software applications.

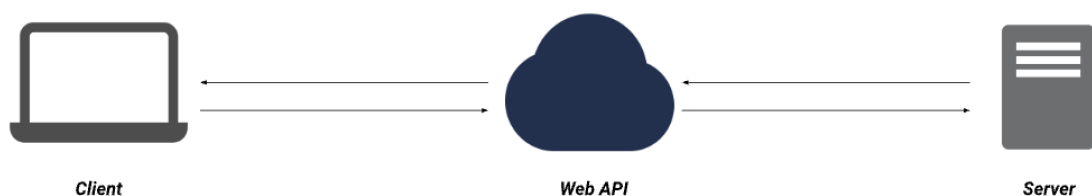
# Chapter 2

## Technical Background

---

To fully grasp the design process of an ontology management tool, it is essential to understand its technical background. This chapter provides insight into the fundamentals and functionality of web application development, as well as in the context of ontology management. By gaining a deeper understanding of web development elements, readers will be equipped with the necessary knowledge to recognize the reasoning behind the design decisions of the design process.

More precisely, this chapter will explain web application basics such as the client-server architecture and how a Web API can be used to communication in the client-server architecture. A visual representation of this is shown in figure 2.1. We also cover how ontologies are represented in the Pinteg backend. The representation of ontologies is an important aspect of the implementation of the tool, and it is, therefore, necessary to understand how Pinteg has defined ontologies and how the elements change during the project.



**Figure 2.1:** Client-server architecture using a Web API.

## 2.1 Web application

A web application is an application that is accessed and used via a web browser [7]. The accessibility of a web application is one of the main advantages. Users simply need internet access to enter the application as the application is deployed on a remote server [7]. The implementation of a web application often follows the client-server architecture;

- **Client:** The client refers to the application that triggers requests. The client often has a user interface, which is the interface that the user interacts with. The client can request information from the backend of the application on interaction. It is then called that the client requests data from the server [8]. The client could for example be a web browser or mobile application.
- **Server:** The term server is often mentioned together with backend, and refers to the software and/or hardware that is responsible for hosting the application [8]. The server processes requests from the client [8] and generates responses depending on the request. Processing requests can refer to routing the request to the correct handlers, executing necessary business logic in the handlers to handle the request, interacting with databases or external services, and generating responses for the client, to name a few.

## 2.2 Web API

The server-side in a client-server architecture often defines an *Application Programming Interface*, API, to handle the communication between client and server. An API is used to define how information between the two parts should be shared. This commonly involves sharing how requests should be designed, and what will be returned in the response. A Web API is an API that is exposed over the web using standard web communication protocols.

### 2.2.1 HTTP

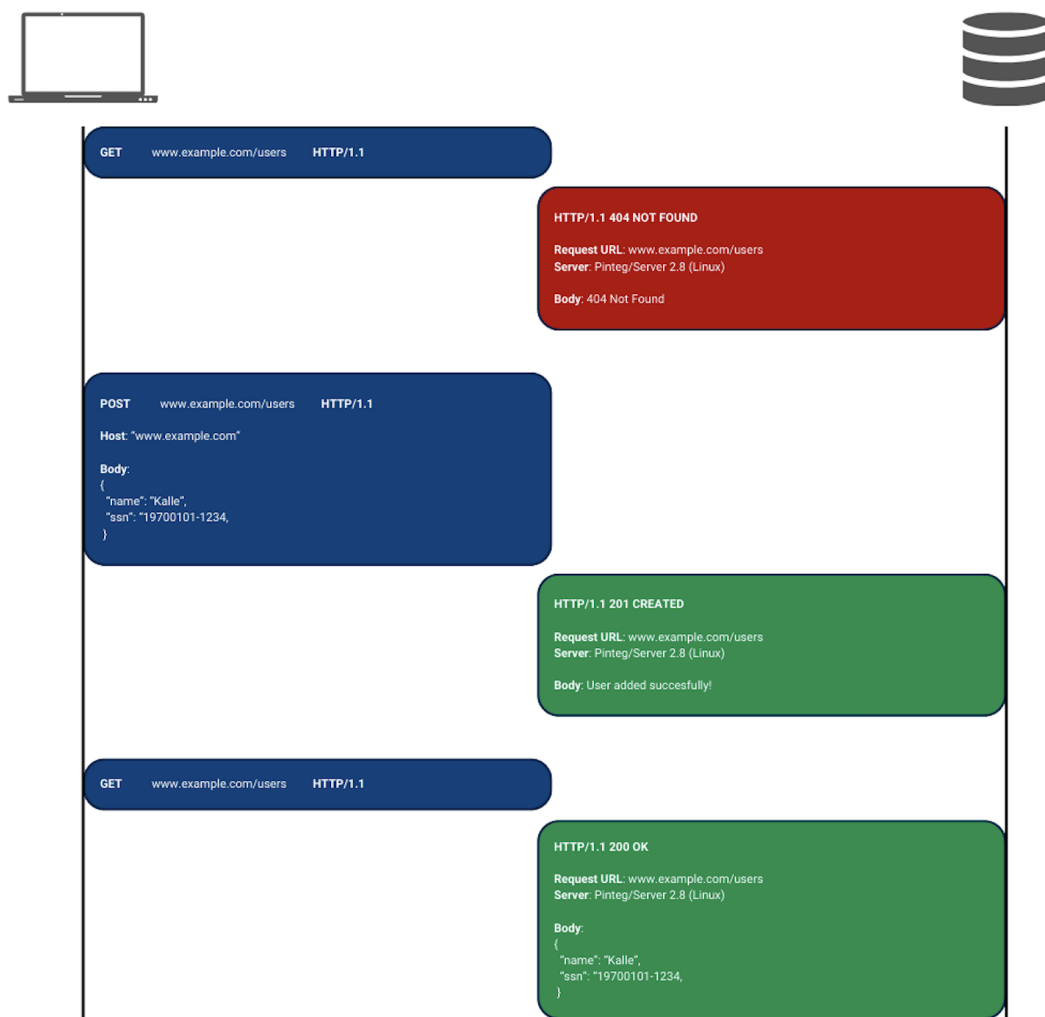
*Hypertext Transfer Protocol*, HTTP, is one of the most commonly used communication protocols on the web. The protocol has fundamental features that are essential to web communication. One of these features is the status code. The status code property is returned by the server [9] and serves as feedback to the client. The client can define its logic depending on the status code that is returned. The codes are classified into classes or categories, where the most essential to understanding the scope of this thesis are

- 2xx - Successful, and
- 4xx - Client Error.

Figure 2.2 shows an example usage of an exchange of HTTP requests and responses between a client and a server, where the status code can be seen at the top of the server responses.

Another fundamental feature of HTTP is request methods. A request method defines the client's actions on the server [9]. In the context of this thesis, relevant actions include

- **GET:** fetches information from the server,
- **POST:** writes information to the server,
- **DELETE:** removes data from the server, and finally
- **PATCH:** updates information on the server.



**Figure 2.2:** A common example of a client-server exchange of HTTP requests and responses.

Furthermore, a relevant feature to the context of this thesis is the HTTP *Uniform Resource Identifiers*, URI, which is commonly referred to as endpoints. An endpoint points to a certain resource on the server. In practice, the client defines exactly what resource they are trying to take action on in the request to the server [9]. In the figure 2.3, this field can be seen after the HTTP request method.

Finally, the last HTTP feature worth mentioning is the message body. The message body is the field in the request where the client adds information to either add or update on the server and where the server adds information for the client to read [9]. This field is the last in the request and can be seen in figure 2.4.

```
GET    www.example.com/users    HTTP/1.1
```

Figure 2.3: The HTTP GET-message from figure 2.2.

```
POST    www.example.com/users    HTTP/1.1
```

```
Host: "www.example.com"
```

```
Body:
```

```
{  
  "name": "Kalle",  
  "ssn": "19700101-1234",  
}
```

Figure 2.4: The HTTP POST-message from figure 2.2.

## 2.2.2 JSON

The information added to the message body of an HTTP request is commonly in the file format of *JavaScript Object Notation*, JSON. The file format is essential to web applications as it is the industry standard for storage and transfers. JSON is in practice a key-value store [10], called a dictionary, where a key can hold a value of certain types. Common types are strings, numbers, arrays, or other dictionaries. In figure 2.5, one can see a JSON object used in the client.

```
{  
  "uri": "resource-1",  
  "library": "library-1",  
  "baseCulture": "en",  
  ...  
  "label": "Resource 1"  
  "description": "The first resource created."  
  "published": true,  
  "type": "Concept"  
}
```

Figure 2.5: An example resource that is returned from the API, in JSON format.

## 2.3 The API

A valid HTTP request can have a method, endpoint, and possibly some data in the JSON format depending on what method is chosen. An action where something should be added or changed on the server often requires some form of data attached. An action that reads or removes resources often does not need any extra data.

A common practice during the implementation of a client is to design requests after specifications in the Web API. The client requests resources based on the specifications, and the server responds based on what resource and action is requested. For the web application of this thesis, Pinteg has provided the backend and specified what server-side operations are possible with OpenAPI Specifications.

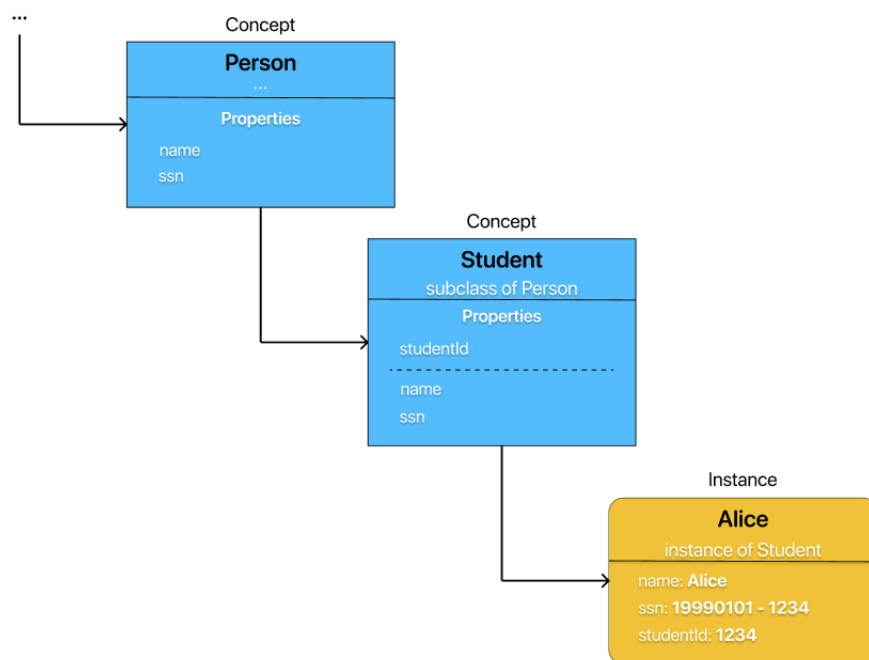
Through the Pinteg Web API, it is possible to manage ontologies. Ontologies are denoted as **resources** in the API and hold relational values that specify what ontology it is or belongs to. A resource can in turn be either a **concept**, or an **instance**. A concept can itself be instantiated and is then called an instance. Concepts and instances can be created, modified, and removed through the API. We want to underline the usage of terms here. We will refer to ontologies as **ontologies**, but in the context of the tool the definition of an ontology will differ throughout the process. Previously in this chapter, we mentioned that a client can fetch and edit resources on a server. With resources, we referred to a general resource on a general server, for example, a song or artist on Spotify<sup>1</sup>. From now on, when we refer to a resource we specifically mean the concept or instance that is stored on the Pinteg server.

To set the context in an example, say that "Person" is a concept. In real life, a person has a name and a social security number and therefore we define properties to the concept called "Name" and "Social security number", ssn. A "Student" is also a "Person", and we define it as a subclass of "Person". "Student" is also a concept, and can therefore define its properties, like "StudentId". "Student" must also take the properties of "Person", as it is a subclass of "Person". "Alice" is an instance of "Student". She therefore has to take the properties of "Student", and has a name, social security number, and student ID. For visualization, see figure 2.6

In the Pinteg API, concepts and instances are commonly denoted as resources as they share certain properties, but as one can see in the example above, they also have certain properties that do not exist on the other.

---

<sup>1</sup><https://open.spotify.com/>



**Figure 2.6:** The ontology example described in the previous paragraph.

# Chapter 3

## Theoretical Background

---

*In this chapter, the theory used throughout the project will be presented. This includes key concepts such as user-centered design, usability, design processes, data-gathering techniques, prototyping, and usability testing. By exploring these principles and methodologies, readers will gain insights into the framework guiding the creation of intuitive, efficient, and satisfying user experiences.*

### 3.1 User-Centered Design

User-centered design, UCD, or Human-centered design, HCD, is a foundational approach in human-computer interaction, emphasizing the importance of designing products and systems around the needs, preferences, and capabilities of users [11] [12]. It prioritizes understanding users' behaviors, goals, and contexts of use to create intuitive and effective user experiences. The design process involves iterative cycles of design and evaluation, with a focus on involving users throughout the design process [12]. This approach ensures that the resulting products meet users' needs and expectations, leading to higher levels of user satisfaction and usability. UCD includes some important ideas that help designers make things centered around users. These ideas are the basic building blocks for making solutions focused on users [13]:

- Early focus on users and tasks
- Empirical measurement
- Iterative design

These principles emphasize the importance of understanding users' needs and behaviors, measuring their reactions and performance at an early stage, and continuously iterating and adapting the design based on user testing. This approach can lead to more tailored and user-centered computer systems [11, 13].



## 3.2 Usability

Usability is a fundamental aspect of interactive system design, focusing on the ease of use and user satisfaction with a product.

Usability principles focus on ensuring that interactive products are easy to learn, effective to use, and enjoyable from the user's perspective. The key goals of usability can be divided into six aspects [11]:

1. **Effectiveness:** Refers to how well a product accomplishes its intended tasks. Are users able to carry out their work efficiently and access the information they need?
2. **Efficiency:** Concerns the ease with which users can complete tasks. Is the product designed to support users in carrying out their tasks in a minimal number of steps or actions?
3. **Safety:** Involves protecting users from dangerous conditions and undesirable situations. Does the product prevent users from making serious errors, and does it provide a means for easy recovery if errors occur?
4. **Utility:** Describes the extent to which the product provides the right functionalities for users to accomplish their tasks. Does the product offer the appropriate set of functions to enable users to carry out their tasks effectively?
5. **Learnability:** Refers to how easy it is for users to learn to use the product. Can users quickly understand how to use the product's features, especially for infrequently used tasks?
6. **Memorability:** Addresses how easy it is for users to remember how to use the product once they have learned it. Can users recall how to perform tasks, especially for features they use infrequently?

Each goal is accompanied by specific questions that aid in assessing the product's usability concerning that aspect. By answering these questions, designers can identify potential design problems early in the process and create a more user-friendly product. These usability principles serve as criteria for evaluating the product's performance and user experience, ensuring that it meets the needs and expectations of its users [11].

In addition to the principles outlined above, Nielsen's 10 usability heuristics provide further guidance on creating user-friendly interfaces [14]:

1. **Visibility of system status:** Ensure users are always informed about what is happening within the system.
2. **Match between the system and the real world:** Design interfaces that speak the users' language and follow real-world conventions.
3. **User control and freedom:** Allow users to easily navigate and undo actions, providing a sense of control.
4. **Consistency and standards:** Maintain consistency in design elements and follow established conventions to reduce cognitive load.

5. **Error prevention:** Aim to prevent errors from occurring in the first place rather than relying solely on error messages.
6. **Recognition rather than recall:** Minimize users' memory load by making information visible and easily retrievable.
7. **Flexibility and efficiency of use:** Provide shortcuts and customization options to cater to both novice and expert users.
8. **Aesthetic and minimalist design:** Keep interfaces focused on essential information and avoid clutter.
9. **Help users recognize, diagnose, and recover from errors:** Present clear error messages and provide assistance to help users resolve issues.
10. **Help and documentation:** Offer easily searchable documentation focused on user tasks when necessary.

These principles complement the six aspects of usability, providing a comprehensive framework for evaluating and improving the user experience of interactive products. By adhering to these principles, designers can create interfaces that are intuitive, efficient, and enjoyable for users to interact with.

## 3.3 Design Process

The design process for interactive products is a critical phase that begins after the establishment of specific requirements. The design process outlined consists of four steps of important activities [11]:

1. **Establishing Requirements:** This initial stage involves gathering information about user needs, preferences, and tasks. It often includes techniques such as interviews, surveys, and observations to understand user requirements comprehensively.
2. **Designing Alternatives:** Once requirements are gathered, designers generate multiple design alternatives to address these requirements. This stage encourages creativity and exploration, allowing designers to consider various approaches and solutions.
3. **Prototyping:** Prototyping involves creating tangible representations of design concepts, ranging from low-fidelity sketches to high-fidelity interactive prototypes. Prototypes enable designers to visualize and test design ideas, gather feedback, and refine the design iteratively.
4. **Evaluating:** Evaluation is a critical stage where designers assess the effectiveness and usability of the proposed designs. This may involve usability testing with real users, heuristic evaluations, or expert reviews. Evaluation helps identify usability issues, validate design decisions, and inform further iterations of the design.

By following this iterative design process, designers can systematically address user needs, refine design concepts, and ultimately create interactive systems that are intuitive, effective, and satisfying to use [11].

## 3.4 Data Gathering

Gathering relevant data is crucial for informing and guiding interactive system design. Different types of data can be collected; quantitative or qualitative, and subjective or objective [11].

Quantitative data refers to information that can be quantified and measured in numerical terms. It focuses on collecting data in the form of numbers and statistics to analyze patterns, relationships, and trends. On the other hand, qualitative data describes qualities and characteristics that cannot be measured numerically. It focuses on gathering in-depth understanding [11].

Subjective data reflects personal opinions, experiences, and interpretations. It represents an individual's viewpoint and can be influenced by their emotions, values, and biases. In contrast, objective data is impartial and independent of personal opinions or interpretations. It is based on facts and can be verified by multiple individuals. Objective data is typically more reliable and credible for research and analysis [11].

In this section, we explore various methods of data collection and the different types of data that can be gathered with each method.

### 3.4.1 Interviews

Interviews play a crucial role in gathering insights and understanding user perspectives in interactive system design [11]. There are four main types of interviews: open-ended, structured, semi-structured, and group interviews [11]. The choice between these depends on the research goals, the questions to be addressed, and the project's stage.

The open-ended interviews resemble exploratory conversations. They delve deep into topics without predetermined expectations, allowing both the interviewer and the participant to guide the discussion naturally. Structured interviews, on the other hand, are based on predetermined questions. Each participant receives identical questions, ensuring standardization. Semi-structured interviews blend aspects of both open-ended and structured approaches. With a basic script, the interviewer explores topics with predetermined questions and probes for additional insights. Maintaining neutrality and avoiding bias is crucial [11].

Planning and conducting interviews involve preparation. During the interview, it is important to listen actively and to have a structured approach to data collection. An introduction, warm-up, main session, cool-off, and closing sequence outline the interview process, ensuring consistency across participants [11]. The data collected by performing interviews is subjective and can be both qualitative and quantitative, depending on the interview style [11].

### 3.4.2 Think Aloud

Think Aloud is a usability testing method that is both simple and effective, encouraging users to verbalize their thoughts while interacting with a system [15]. This method is cost-effective, robust, and easy to implement, providing valuable insights into users' experiences and reactions to the design. The data collected using this method is subjective qualitative data [11].

### 3.4.3 Observation

Observation is a method to collect data [11]. Direct observations involve being present during the user's activity and capturing non-verbal signals and behaviors in real time. Indirect observations involve collecting data through recordings or log files, which can provide insights over time or on a larger scale. Through observations, data is gathered about users' behavioral patterns and preferences to inform design decisions and enhance the user experience. The data collected is objective and can be qualitative and quantitative [11].

### 3.4.4 System Usability Scale

System Usability Scale (SUS) is a simple and quick method to assess the usability of a system, where usability refers to the effectiveness, efficiency, and satisfaction with which users can interact with a system in a particular context [16].

The SUS consists of ten five-point Likert scale items, each covering different usability aspects such as complexity, integration of functions, consistency, ease of learning, need for support, confidence in usage, and perceived need for training. Respondents rate each item on a scale of 1 to 5, from "Strongly Disagree" to "Strongly Agree".

To use the SUS effectively, respondents should provide their immediate responses to each item without prolonged consideration. If they cannot respond to a particular item, they should mark the center point of the scale. The SUS score is later calculated to range from 0 to 100, with higher scores indicating better usability.

To calculate the SUS score, one should follow the following calculation:

$$X = \text{sum of scores of odd numbered questions} - 5 \quad (3.1)$$

$$Y = 25 - \text{sum of scores of even numbered questions} \quad (3.2)$$

$$\text{SUS Score} = (X + Y) \times 2.5 \quad (3.3)$$

The SUS score should be interpreted as visualized in figure 3.1. A score above 80 indicates the system is very well-designed and easy to use. The median SUS score is considered to be 68 [17]. If the score is below 50, it indicates poor usability, and significant improvement is needed.



**Figure 3.1:** A visualization of how to interpret a SUS score, displayed as a scale.

The SUS is particularly useful in industrial usability evaluation where there is a need for a quick and reliable measure of overall system usability. It generates subjective and quantitative data [11].

## 3.5 Prototyping

To effectively evaluate the design, designers can create prototypes of their concepts [11]. Prototypes can range from simple paper models to more sophisticated versions resembling the final product. The starting point for the design process can either be from scratch or by modifying an existing product.

Prototypes are not only important for evaluating the design but also serve as an effective communication tool between developers and stakeholders [11]. By showcasing prototypes, developers can convey their thoughts and ideas about the product clearly, while stakeholders gain a more concrete understanding of how the final product may look and function.

Prototypes can be categorized into different types depending on their complexity and resemblance to the final product. Low-fidelity prototypes are simple and quick to produce, often used in the early stages of the design process to explore different design options [11]. Mid-fidelity prototypes fall somewhere between low and high fidelity in complexity and resemblance to the final product. These prototypes can be used to test both design alternatives and technical aspects in a slightly more advanced manner than low-fidelity prototypes but without the full level of detail of high-fidelity prototypes. On the other hand, high-fidelity prototypes are more advanced and closely resemble the final product, making them suitable for testing technical issues and pitching ideas to stakeholders [11].

## 3.6 Usability Testing

Usability testing is a method aimed at evaluating and improving the user experience of products and services by involving actual users in controlled or natural environments. The goal is to identify and address potential issues and areas for improvement to ensure that the product or service is intuitive, efficient, and satisfying for users [11, 18].

A fundamental principle behind usability testing is to gather data about users' performance of specific tasks and their reactions to the product or service during use. This may involve observing user behavior, collecting feedback through surveys or interviews, and recording performance metrics such as task completion time and error rates [11, 18].

To conduct usability tests, careful planning of test scenarios and tasks reflecting the real needs and goals of users is required. Test participants should represent the actual user base for the product, and testing should occur in an environment similar to where the product or service will be used. There are several methods and techniques for usability testing, including tests in a controlled environment where users perform tasks in a laboratory, field tests where the product or service is tested in real-world usage environments, and remote usability tests where users perform tasks remotely using computer-based tools [11, 18].

The results of usability tests are used to identify usability issues and areas for improvement, which can lead to design changes and optimizations of the product or service [18].

# Chapter 4

## Method

---

*In this chapter, the method used to conduct the project is presented. The project is structured into four phases, with each phase focusing on different aspects of creating, refining, and evaluating the product. The aim was to establish a deep understanding of the project and the product, as well as to iteratively develop and enhance the product's user experience.*

### 4.1 Timeline of Key Activities

The project is divided into four phases, the first phase includes creating an understanding of the project and product, whereas the other three phases include iterations of creating, refining, and evaluating the product. Through iterative cycles of designing and prototyping, usability testing, and evaluation with the stakeholders, these prototypes evolve into more refined versions. The process is visualized in figure 4.1.

In **Phase 1**, existing literature on ontology management was examined, alongside evaluations of legacy tools. The goal was to establish a foundation by gathering insights from empirical studies. Simultaneously, qualitative insights were gathered through user interviews to broaden the understanding of user needs and pain points. Later, the focus shifted from theoretical groundwork to practical implementation. Personas and use cases were developed as initial iterations, serving as a platform for exploring design concepts and user interactions.

The personas and use cases created in the first phase were later used throughout the process. User feedback was gathered from usability testing, emphasizing a user-centric approach prioritizing usability and accessibility. The prototype and the results from the usability test were then evaluated and analyzed with the stakeholders. The insights given through this evaluation were later used for the development of the next prototype. This iterating approach can be seen in figure 4.1. **Phase 2**, the focus was put on creating a lo-fi prototype.

**Phase 3** included the creation and evaluation of a mid-fi prototype. As the prototype progressed to a more realistic product, the focus shifted towards refining specific features and simulating real-world usage scenarios. Usability testing was performed on the devel-

oped prototypes across different scenarios, from basic interactions to complex use cases. An evaluation with the stakeholders was also performed in this iteration.

**Phase 4** represents the project's culmination, where a hi-fi prototype was developed. This stage is crucial for fine-tuning the user experience and addressing nuanced aspects of functionality and design. The developed product later underwent testing. Usability testing in real-world contexts is performed to validate overall functionality and user experience. The results from the testing, as well as the hi-fi Prototype, were later evaluated with the stakeholders.

By integrating usability testing and evaluation with the stakeholders in each iteration of development, we aim to create an ontology management application that not only meets but exceeds user expectations, setting a new standard for usability and effectiveness within the domain.

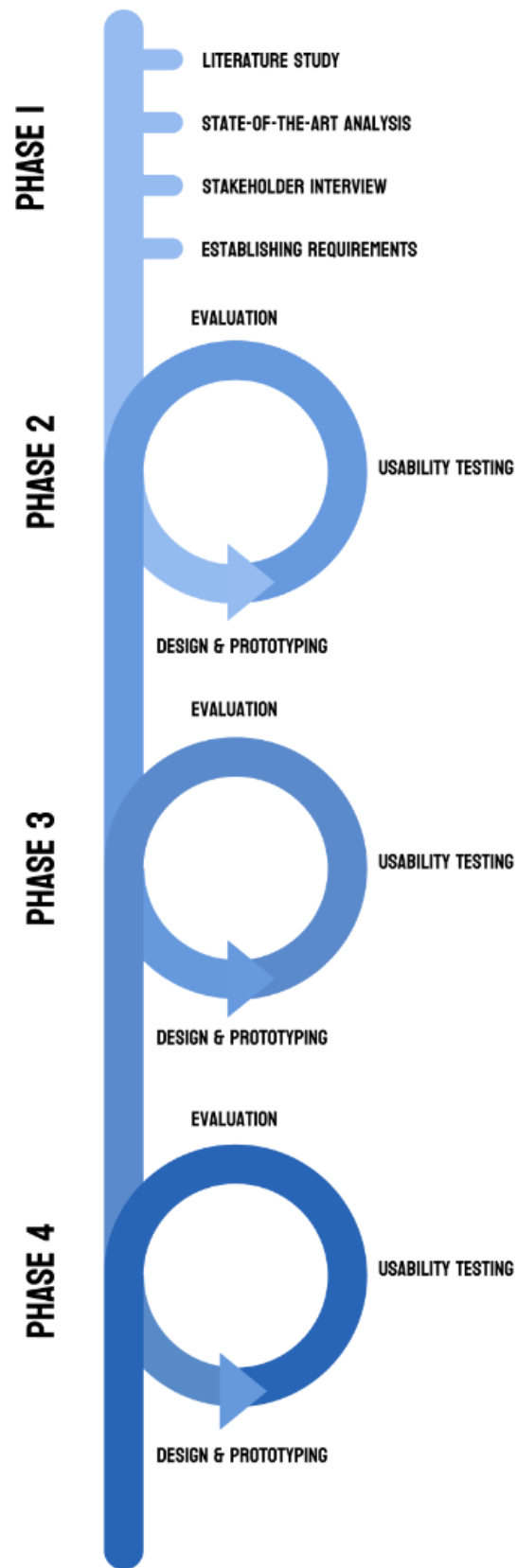


Figure 4.1: Overview of the Design Process.



# Chapter 5

## Phase 1: Conceptual design

---

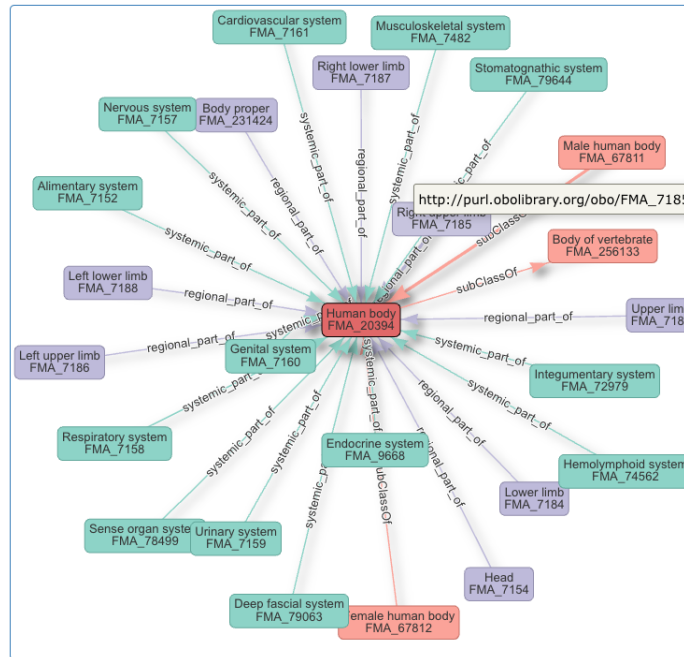
*The initial phase began with a deep dive into fundamental principles within ontologies, ontology management, and user-centered design. The phase includes a literature review on usability, a state-of-the-art analysis of ontology management, and an interview with stakeholders. With the gathered information, personas and use cases were then developed. At the end of the phase, functional requirements were defined based on the use cases. The literature review is presented in section 1.5.*

### 5.1 State-of-the-Art Analysis

To gain a comprehensive understanding of the landscape of ontology management tools, a review and analysis of available tools were conducted. It was crucial to develop a deep understanding of how ontology management tools work and could look. This process included an overview of the tools' features, user experience, and technical capabilities. By using the tools in practice, insights about their strengths and weaknesses were gathered, helping the assessment of the suitability for the projects' specific use case. Two of the tools reviewed were OLS4 and Protégé, as they were considered some of the most used [5].

OLS4 is a popular ontology manager that primarily focuses on providing easy access to existing ontologies in bio-informatics [6]. One of its main strengths is its ability to handle large amounts of data and visualize it in a user-friendly manner. Additionally, OLS4 offers advanced tools for data integration and semantic search, making it a versatile tool for researchers and developers in various domains [6]. The website is updated daily to ensure the relevancy of its ontologies.

However, OLS4 cannot effectively visualize versions of ontologies, which can be a limitation for users who want to track changes over time. Another interesting observation was how the application displays ontologies. It uses two main visualization tools, a hierarchy view and a graph view. Figure 5.1 displays the graphical view of an ontology in OLS4. Regarding the user interface, OLS4 has a simple design with clear colors, but the visualization of ontologies was still a challenge to understand directly.



**Figure 5.1:** Graphical view of a Human body ontology from OLS4.

Protégé is a leading tool in ontology development and management, offering a comprehensive set of features and tools for creating, editing, and analyzing ontologies. One of its main strengths is its flexibility and adaptability, making it an ideal tool for a variety of use cases and domains [5].

Despite many advantages, Protégé had some shortcomings that prevent it from being a complete ontology manager for certain users. When managing ontologies, we had a hard time understanding how to use the tool, and we concluded that its learning curve may be steep for new users. Because of this deduction, it was important to design test cases and questions for our tool that will examine the learning curve.

Generally, the user interfaces of other alternative tools looked outdated. As for OLS4 and Protégé, both the web applications and downloadable applications are still being managed. As for Protégé, the source code is available on GitHub<sup>1</sup> and is labeled as an Open Source community, i.e. any user can contribute to it and the source code is available. But while testing these tools, it was striking how the user interface looked old, and there was no direct tutorial for new users.

## 5.2 Interview

An open-ended interview with the stakeholders, who initiated the product request, was conducted during this phase. This process was pivotal for several reasons, especially considering that the stakeholder also represented a potential end-user.

Engaging in this interview allowed us to delve deeply into the stakeholders' perspectives and requirements. By gaining insight directly from the stakeholders, who also embodied the

<sup>1</sup><https://github.com/>

role of prospective users, we understood their specific needs, expectations, and challenges regarding ontology management.

The following points represent the outcome of the open-ended interview and were compiled in a prioritized order based on the importance of the described feature.

1. **Handling of Ontologies:**

- The application should enable adding, modifying, and deleting ontologies.
- There are predefined templates that must be used when creating ontologies.

2. **Login and Organization Linkage:**

- Users should be able to log in via their organization.
- The ontologies in the application should be tied to a specific organization.

3. **Submit Functionality:**

- Users should be able to submit and share data from their organization.
- Data submitted should be viewable by others outside the organization.
- There must be clear ownership of submitted data, with one or more individuals as owners.
- Ownership of data must be preserved after publication.

4. **Verified Standard:**

- All submitted data can be verified by Pinteg to create a common standard, a way to verify the authenticity and approve submissions.

5. **Collaboration and Contribution:**

- Users should be able to view all global data by filtering the available data.

6. **Data Import via API:**

- The application needs to be able to fetch a large file via API.
- It should then be able to read the file and export it to other formats.
- The work should be done against the current data model.

## 5.3 Establishing Requirements

Using the results from the interview with stakeholders as guidance, the design process began by creating personas.

1. **Understanding User Needs:** The process began by synthesizing insights gathered from the Understanding Phase, including literature reviews, state-of-the-art analysis, and stakeholder interviews. Analysis of this information helped identify common features, requirements, and preferences, forming the foundation for the design process.

2. **Developing Personas:** Personas were created to represent distinct user types within the target audience. These personas were based on job roles, goals, and features recommendations, derived from research. By personifying users, it became easier to understand their perspectives and tailor design solutions accordingly.
3. **Identifying Use Cases:** Use cases were documented to outline the functional requirements of the ontology management application. These described the specific actions users would perform within the system to achieve their goals, ensuring that design solutions are closely aligned with user needs and workflow requirements.

Throughout this phase, an iterative approach was employed, refining design artifacts based on ongoing feedback from the stakeholders. Continuous iteration and refinement of conceptual models, personas, and use cases ensured that design solutions remained closely aligned with user needs and objectives, ultimately leading to a more effective and user-friendly ontology management application.

The results from the first phase are summarized in table 5.1, which presents the initial requirements for the tool. These requirements serve as the starting point for our exploratory process in developing a user-centered solution. Preliminary, they provide a solid foundation for guiding our design and implementation efforts to create a solution tailored to user needs.

Personas and use cases were later created from the specified requirements.

**Table 5.1:** Requirements defined given conclusions from the state-of-the-art analysis, literature review, and user interview.

ID	Requirement
1	<b>Management of Ontologies:</b> Implement functionality to manage resources according to the given data model.
2	<b>Organization Management:</b> Login must be associated with an organization. Objects must be linked to a specific organization.
3	<b>Publication:</b> Ability to publish data from an organization. Accessibility to view published data outside the organization. Requirement for ownership attributes for one or more private individuals. Determination of ownership post-publication.
4	<b>Collaboration:</b> Integration with and ability to use existing ontologies defined in the tool.
5	<b>Version Control:</b> Implement version control to manage changes and revisions of data over time.
6	<b>File Access via API:</b> Ability to retrieve a large file through the API. Work with the provided data model to read in and export data to other formats.

### 5.3.1 Personas

Firstly, three personas were created to represent distinct user types within the target audience. These personas are summarized below. The full descriptions of these personas are defined in appendix A.



#### **Sofia, the Knowledge Seeker**

Sofia represents a typical student who often encounters intensive tasks in her education. As a student striving to find information through the school's organizational login, her role emphasizes the importance of providing user-friendly and easily accessible tools to visualize and understand information from the ontology tool. Despite not actively using the tool's functions, Sofia must be able to visualize the information clearly and comprehensibly to facilitate her studies and research. By including Sofia as a persona, we aim to ensure that the ontology tool offers simple and intuitive visualization of information, making it an invaluable resource for users who need to navigate complex knowledge domains.



#### **Emma, the Domain Expert**

Emma is an expert in data security laws with extensive experience in the field. She understands the importance of accuracy and precision when interpreting and applying laws and regulations. Emma is an invaluable asset to her organization when it comes to understanding and navigating legal issues and regulations.

By including Emma as a persona, we aim to meet the needs of users with specialized knowledge in a specific area. Emma will use the tool to share the deep knowledge she possesses within her domain. By effectively incorporating this knowledge into the tool, other members within her organization, as well as external parties, can benefit from her expertise and make informed decisions that comply with applicable laws and regulations. Emma strives to make complex legal concepts and processes accessible and understandable to all users of the tool, thereby promoting the overall efficiency and compliance of the organization.



#### **Oscar, the Project Manager and Lead Backend Developer**

Oscar's persona was developed based on insights from the primary stakeholder at Pingteq, who embodies similar responsibilities and expertise. Oscar serves as the central link between technical requirements and user needs. He comprehends the demands of backend development and the expectations of end-users, as he belongs to both groups. Oscar's responsibilities extend to shaping the requirements for the ontology management tool and leading the backend development efforts. His role involves coordinating technical functionality with

user-centered design principles to ensure a cohesive and user-friendly experience.

By integrating Oscar as a central persona in the process, we ensure that the application's technical aspects are robust, scalable, and efficient. Simultaneously, we emphasize usability and accessibility for end-users. Oscar's expertise and leadership ensure that backend development not only meets technical requirements but also adapts to meet the needs and expectations of users.

### 5.3.2 Use cases

After thorough consideration, two personas were chosen: Emma, the domain expert, and Sofia, the information seeker. Oscar, despite his valuable role as a central link between technical requirements and user needs, was not chosen as a persona. This decision was made because Oscar's profile closely resembled the stakeholders in the project, who would be communicating and conducting evaluations throughout the project. Including Oscar as a persona was deemed redundant, as the project stakeholders naturally would provide continuous input and feedback similar to what Oscar would offer.

Emma was chosen as the primary focus due to her crucial role within the domain and alignment with the project's goals. However, considering the broad user base, Sofia was also selected to represent another significant user group. While Emma's expertise aligned with the core functionalities, Sofia's perspective could provide insight into the user experience for a broader audience. The use case defined for Sofia, the knowledge seeker was:

1. **Access and Exploration:** Sofia needs intuitive access to ontologies and concepts relevant to her role without extensive domain expertise. Use cases include browsing, searching, and filtering features to explore existing ontologies and discover relevant concepts.

The use cases for Emma, the domain expert, include various functionalities aimed at creating and facilitating the management of ontologies and concepts within, and outside of, her organization:

1. **Creating New Ontologies:** Emma needs to easily create new ontologies, even with limited knowledge of semantic modeling, to contribute effectively to data management. She wants to access existing ontologies, create a new ontology, define relevant concepts, relationships, and properties, and save the resources for browsing.
2. **Editing Ontologies:** Emma requires the ability to make changes to existing ontologies easily to adapt to evolving data management requirements. She wants to navigate to the ontology within the management tool, select the "Edit"-option for the desired resource, make necessary modifications using user-friendly editing tools, validate changes for consistency and compliance, and save or publish the updated ontology for team access.
3. **Collaborating with Team Members:** Emma needs seamless collaboration with team members to ensure ontology accuracy and completeness. She wants to share ontologies, publish resources for team use, and track changes and revisions made by team members for transparency and accountability in the ontology management process.

4. **Collaborating with Other Organizations:** Emma desires effortless importation of existing ontologies from other organizations to leverage external resources and best practices. She wants to access the subscription feature, browse published ontologies from subscribed organizations and global standards, select desired ontologies for subscription, import resources into her organization's ontologies, and review imported ontologies for relevance and coherence, making necessary adjustments.

### 5.3.3 Functional Requirements

Before starting the second phase, which included the development of a lo-fi prototype, priorities were made regarding requirements.

Since there were currently no features supporting the functionality for collaboration in the backend system, it was decided to initially only implement the feature where a user could publish changes. Later iterations of the project could then build upon this, gradually introducing collaboration features as the underlying system functionalities were developed and refined. Another requirement that was not prioritized in the initial implementation was the version control, this since there were no backend implementation for this feature either.

Although the ability to import data via API is useful and can enhance the system's usability, it was assessed that it was not an immediate priority compared to other fundamental functions for ontology management and publishing. Implementing this function was considered more resource-intensive and required integration work with external systems and data sources.

The three requirements to be developed in the lo-fi prototype were:

1. **Search for Ontologies:** Taking into account both the literature review and the stakeholder interview, it was realized that users need to be able to search for existing ontologies to explore and use them in their work. This requirement is supported by the need to provide an intuitive and accessible platform to facilitate knowledge discovery and access for users.
2. **Manage Ontologies:** After identifying users' needs and preferences in the interview, it was realized that users must be able to create, edit, and delete ontologies, concepts, and instances. This requirement will support users' needs to customize and manage their ontologies based on their specific requirements and changing needs over time.
3. **Publish Changes:** Based on insights from discussions with stakeholders, the importance of users being able to publish their ontologies after making their changes was understood. This requirement will facilitate the sharing and reuse of ontologies among users and organizations, promoting collaboration and exchange of knowledge and resources.

These three functional requirements were chosen to address the key needs and requirements that emerged from the literature review, state-of-the-art analysis, stakeholder interviews, as well as the development of personas and use cases. By incorporating these requirements into the system, a user-centered and effective solution for ontology management will be offered, meeting users' needs and expectations.

# Chapter 6

## Phase 2: Lo-Fi prototyping

---

*Following the process outlined in the method, Phase 2 includes the creation and evaluation of Lo-Fi prototypes. Before diving into the details of design, it's essential to comprehend the foundational steps undertaken to ensure the prototypes align with user needs and project objectives. In this chapter, key features and design decisions, along with visual representations and explanations, will be presented.*

### 6.1 Design and prototyping

The process of creating low-fidelity prototypes began with translating the previously defined requirements and use cases into simple, rough sketches, capturing the basic structure and layout of the ontology management application without delving into detailed design elements.

Based on the use cases identified earlier, storyboards were created to illustrate how our selected persona would interact with the application to accomplish specific tasks. These storyboards helped visualize user journeys and identify crucial touchpoints for design improvements.

During the process, the focus lay on simplicity and functionality, aiming to craft something easily understandable to a general person to ensure a wide range of test participants during the evaluation. A simple ontology was therefore created, intended to be comprehensible to numerous potential users. In this iteration, the focus lies on concepts, to avoid complicating the logic between concepts and instances for the test participants.

To simulate interaction, paper prototypes were crafted using different paper components to represent views and actions. Each page in the prototype corresponds to a part of the interface or a specific action. This method facilitated swift iteration and exploration of different design possibilities while ensuring the prototypes remained understandable and accessible to both stakeholders and users.

The functional requirements defined in section 5.3.3, formed the foundation of the lo-fi prototype.



### 6.1.1 Search

When designing the search feature, the approach was to create a user experience similar to well-known search engines, such as Google Search<sup>1</sup> and Microsoft Bing<sup>2</sup>, making it intuitive and user-friendly. Figure 6.1 shows a search bar accompanied by a prompt "Search for ontologies". This was included to provide users with a familiar interface and make it easy to start their search.

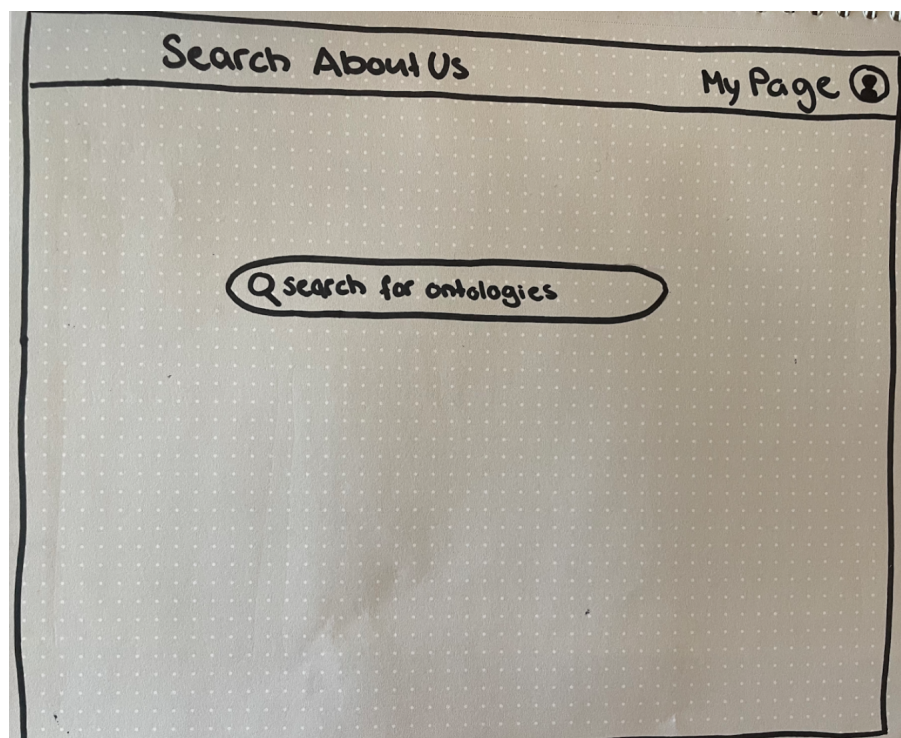


Figure 6.1: The search bar.

To further enhance the user experience, a navigation bar was incorporated at the top of the page, providing an overview of the available features on the website. This navigation bar serves as a guide for users, making it easy for them to locate and use the various features of the website quickly. By using established design patterns and including a familiar navigation structure, the aim was to make the search function intuitive and accessible to users, thereby increasing the usability of the website.

The page visualizing the search results, shown in figure 6.2, was also designed similarly to well-known search engines, for example, Google Search.

<sup>1</sup><https://www.google.com/webhp>

<sup>2</sup><https://www.bing.com/>

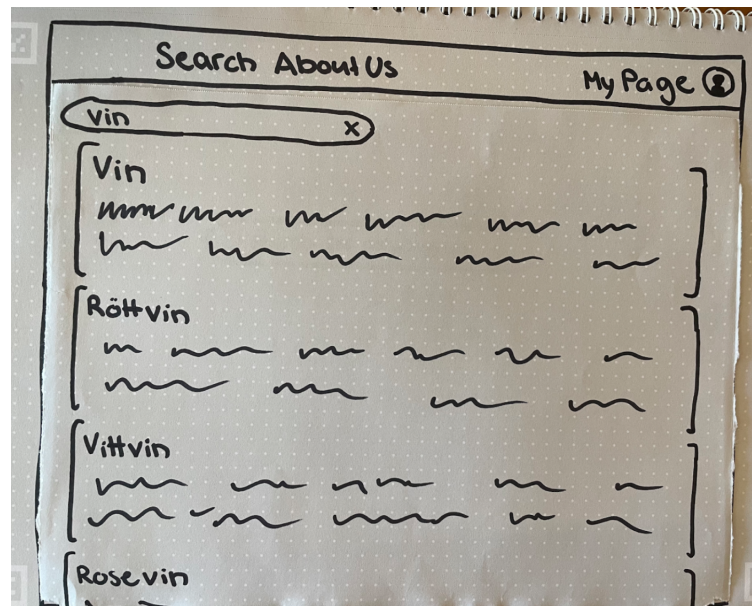


Figure 6.2: Search results for Wine.

## 6.1.2 Manage

A key consideration in our design process was to maintain the integrity of the user flow. When users attempted to access restricted areas, such as "My Page" which can be visualized in figure 6.3, without authentication, they were seamlessly redirected to the login prompt, shown in figure 6.4. This constraint-driven method ensured a cohesive user journey and reduced frustration by guiding users toward the appropriate action.

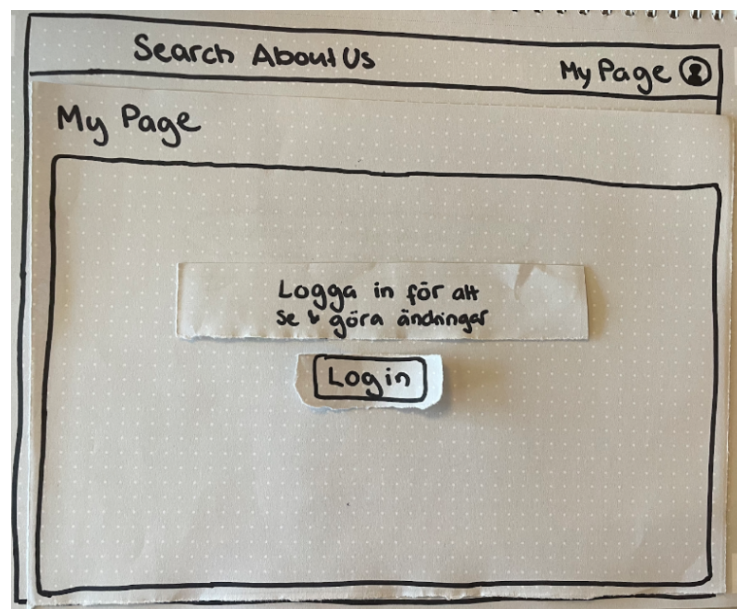


Figure 6.3: "My Page" when the user is not authenticated.

Pinteg utilizes Auth0, a easily adaptable authentication and authorization platform, to manage users and organizations [19]. The lo-fi prototype's login had a simple version of what

the Auth0 login prompt looks like, but as the login prompt was already adapted and implemented by Pinteg, its integration will be discussed in later phases.

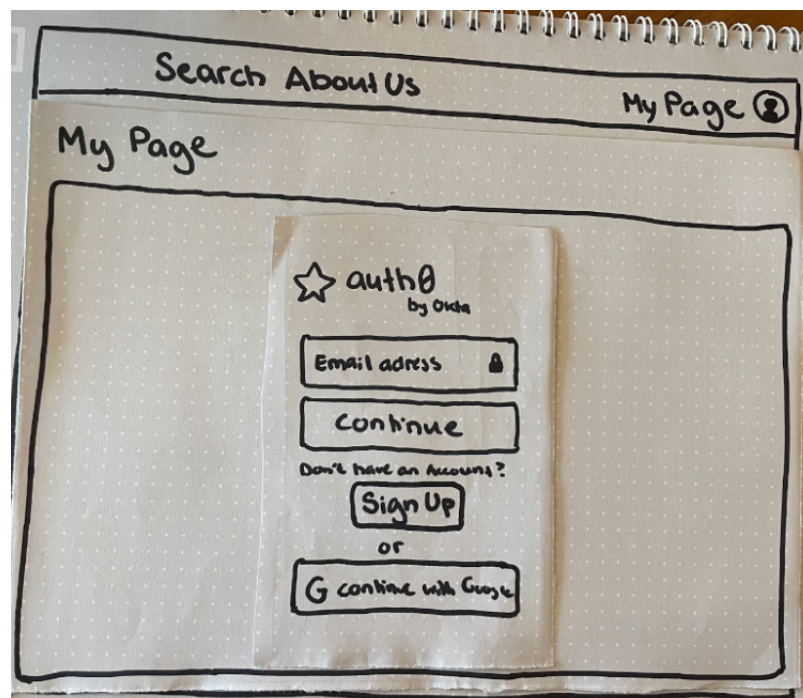


Figure 6.4: Log in process with Auth0.

For the functionality regarding creating, modifying, and deleting ontologies, the design approach was to offer users a streamlined and intuitive experience for managing ontologies. A dedicated page, named "My Page", was established where users could access all their ontologies and related activities. The page was thought of as a centralized dashboard and served as a hub for users to efficiently organize, create, and modify their ontologies.

To enhance navigation within the "My Page" dashboard, a sidebar was created to allow users to search among their ontologies and view other relevant activities. This sidebar was added to provide quick access to different sections of the dashboard.

The main part of "My Page" was devoted to displaying detailed information about the user's ontologies. If users had existing ontologies, they could view relevant details, make modifications, and delete them as needed. Moreover, a prominent "Create new ontology" button was placed on the main page. This design choice aimed to provide users with a clear and direct way to start creating ontologies from the dashboard. A user could also edit a concept by viewing its details, as shown in figure 6.5. The dashboard in "My Page" was created to ensure users that they could manage their ontologies easily without distractions, ultimately enhancing the user experience.



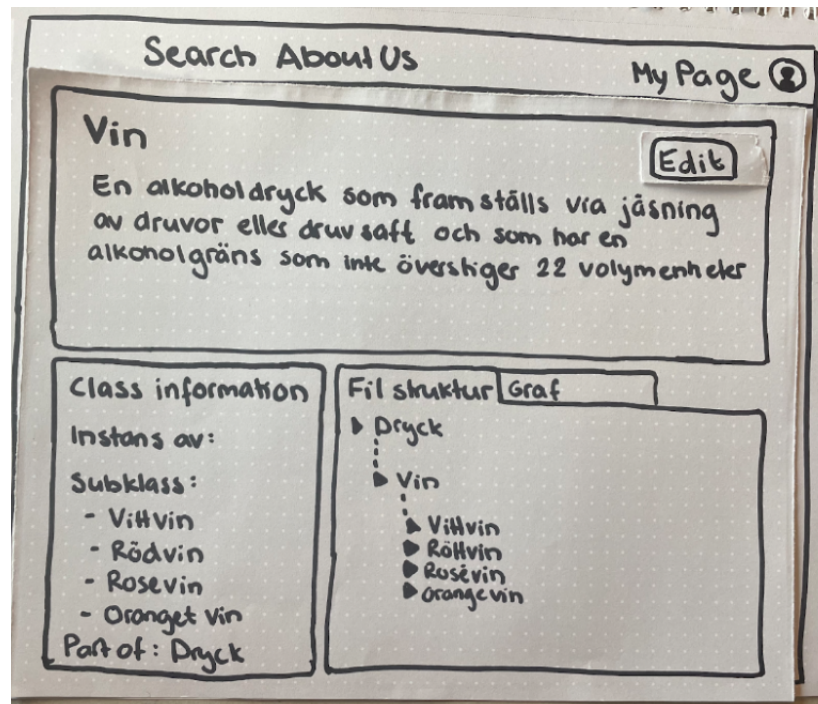


Figure 6.5: Detailed view of the concept Wine.

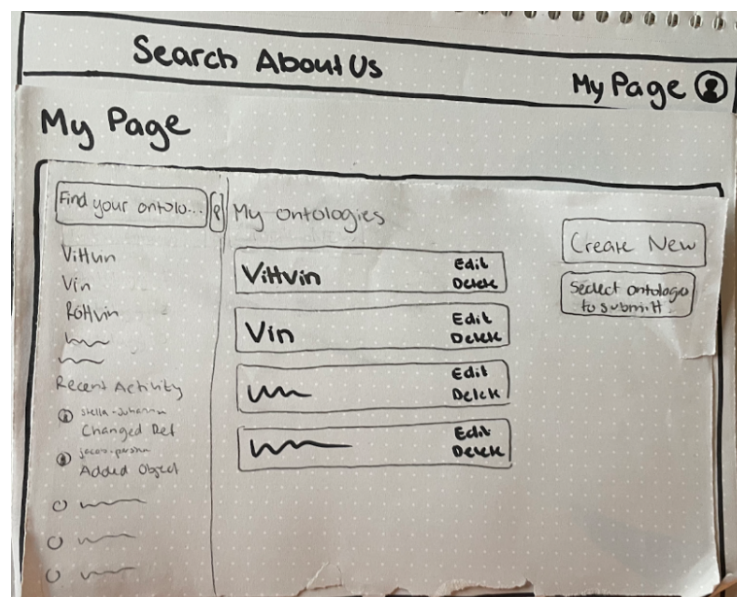


Figure 6.6: Viewing an ontology on "My Page".

### 6.1.3 Collaboration

For this functionality, a "Select Ontologies to Submit" button was created on the "My Page" dashboard, shown in figure 6.6, enabling users to choose which ontologies they wished to publish. After clicking on the button, boxes were shown beside every ontology to provide users with flexibility and control over the ontologies they want to share. See figure 6.7.

Placing the submit button on the dashboard created an intuitive and easily accessible location for users to manage their publishing choices directly from the main page. The aim is to make the publishing process seamless and user-friendly, offering users simplicity and efficiency when sharing their ontologies with others.

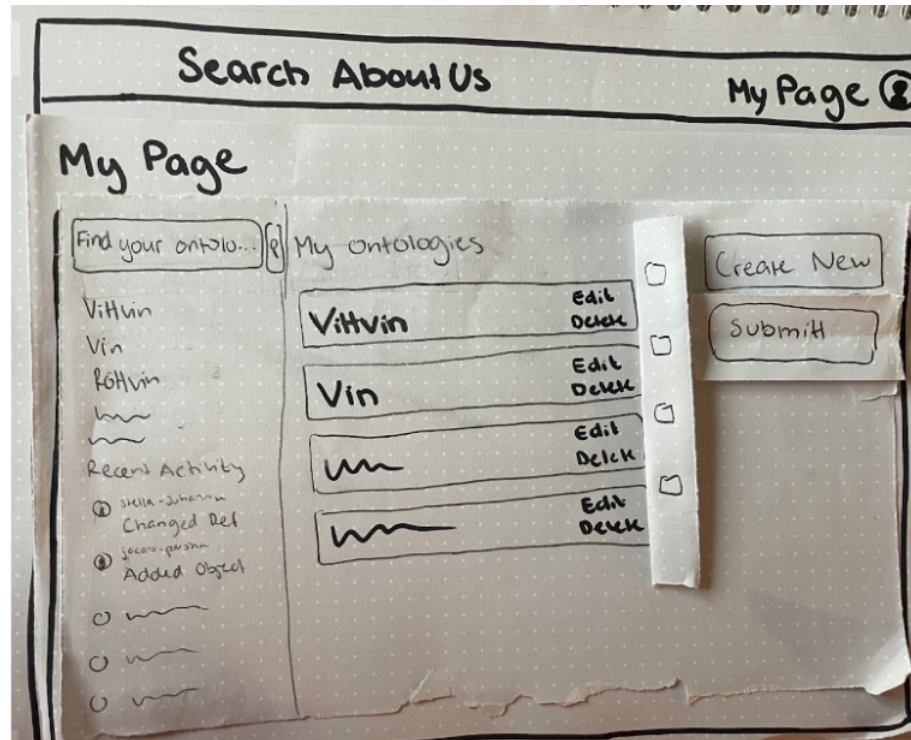


Figure 6.7: View when selecting ontologies to submit.

## 6.2 Usability Testing

Throughout the development of the lo-fi prototyping, feedback was requested from the stakeholders. When the prototype was done, a usability test was performed. During the usability testing of this phase, the focus was to ensure that the conceptual model was understandable.

In this phase, a notable challenge emerged: the difficulty in sourcing test participants who mirrored the intended user group. Within Pinteg, there existed a shortage of potential customers available for testing.

While exploring alternative options, consideration was given to recruiting experts from diverse fields for testing purposes. Nonetheless, this approach was deemed impractical due to the time and resources it would demand. Furthermore, ensuring that their expertise aligned with the features of the prototype presented a significant hurdle.

As a result, the decision was made to test individuals without expertise in a specific domain. Moving forward, the limitations of our testing pool are acknowledged. Nevertheless, efforts will be made to refine testing methods within the constraints of available resources.

The usability test for the lo-fi prototype contained individual test sessions with four participants. The participants were all female students, with an average age of 22, currently enrolled at Lund Tekniska Högskola. The tests were conducted using the think aloud method,

allowing participants to express their thoughts while navigating the prototype. This enabled us to observe their interactions and make note of their opinions. The purpose of this method was to assess the usability and structural coherence of the website. By using a paper prototype, the focus was solely on evaluating the navigation flow and layout.

The test cases, described in table 6.1, provided a framework for evaluating the four functionalities previously specified. Each test case was designed to cover a specific aspect of the system and offer a comprehensive assessment of its usability and functionality.

**Table 6.1:** The test cases defined during Phase 2.

Feature	Description	Test Cases
Search	Users should be able to search for an ontology and view its information.	Users utilize the search function to look for "Wine." They review the search results, select a relevant ontology, and examine its file structure, graphical representation, and detailed information.
Manage	Authenticated users should be able to create new concepts and modify existing ones.	Users log in, access their page, and initiate the creation process. They fill in required fields accurately and save the new ontology. Additionally, users modify an existing ontology through the editing feature.
Collaboration	Users should be able to submit ontologies for public access.	Users navigate to their page, select the ontologies they wish to publish, and initiate the submission process.

The search functionality was evaluated to ensure that users could effectively search for ontologies and understand the presented information. Test cases regarding the creation of ontologies were included to evaluate if the process was straightforward, from entering details to associating concepts. Users' ability to modify existing ontologies via "My Page" was also examined, along with how effectively the user interface facilitated this. Finally, the process of submitting ontologies for other users to access was assessed.

Together, these test cases provided a comprehensive evaluation of the prototype's capabilities and user experience.

## 6.3 Result and Analysis

This section describes and discusses the results of the lo-fi prototype usability testing, as well as the evaluation with stakeholders.

### 6.3.1 Usability test

Firstly, the majority of participants, three out of four, indicated difficulties in understanding the difference between an ontology and a concept. This suggested the need for clearer ex-

planations of terms and perhaps grouping them in a way that enhances user comprehension. One participant also expressed confusion regarding how an ontology is visually represented, further emphasizing the need for clarity and understanding of these concepts.

Secondly, participants noted the absence of clear login and logout buttons in the navigation bar. This indicates a usability issue, highlighting the importance of placing these functions in an easily accessible location to facilitate user account management.

In summary, the results from the user testing demonstrate several strengths of the prototype, including a smooth user flow and straightforward navigation. However, important areas for improvement were identified, such as clarity around terminology, placement of key functions, and enhanced visual representation of ontologies. These insights will be valuable in informing iterative improvements to the prototype to enhance its usability and effectiveness.

### **6.3.2 Evaluation with Stakeholders**

After the usability testing, the prototype was presented and tested with stakeholders. Additionally, the feedback received from testing was reviewed, leading to the conclusion that the conceptual model needed improvement to enhance users' understanding of ontology concepts. Consequently, the handling of ontologies in the backend was defined more clearly. Note that the meaning of this is purely how they are handled, while the conceptual description hasn't changed from how it was described in section 2.

- A resource should contain audit metadata, ie. information regarding who created or modified the resource and when.
- Concepts can subtype one or more other concepts. This means that the object saved on the server holds a list of concepts.
- A resource starts in draft state and can then be published to move into published state.
- Once the resource is published, it cannot be deleted or modified, including changing state back to draft.

With more distinct definitions of the central concepts and functions, the process moved on to the next iteration, where the focus lay on mid-fi prototyping.

# Chapter 7

## Phase 3: Mid-Fi Prototyping

---

*In this chapter we cover the design, evaluation, and analysis of the implementation of the mid-fi prototype, according to the process outlined in figure 4.1. From the results and analysis of the lo-fi prototype, the design process transitioned from lo-fi- to mid-fi prototyping, leaving paper and scissors and focused on the integration of visual and functional details.*

### 7.1 Design and prototyping

The transition from lo-fi to mid-fi prototyping involved a shift towards more polished and interactive designs, by using digital tools for enhanced fidelity and functionality. Our process began with using Figma <sup>1</sup>, a collaborative design tool, to translate conceptual sketches and wireframes into mid-fi digital prototyping. Using Figma, we refined the visual design elements, such as colors, typography, icons, and layout, to create a more polished and cohesive user interface. Interactive elements, including buttons, menus, and navigation bars, were added to simulate user interactions and provide a more immersive user experience.

#### 7.1.1 Overview

As seen in the figures in section 6, the lo-fi prototype generally had rounded objects such as buttons, background boxes, and a search bar. This was an early design choice to make the application visually more pleasing, and therefore more user-friendly. The design choices of fundamental objects in the mid-fi follow the idea that it's crucial to recognize the importance of modernization and innovation upon established design practices. One of the findings from the literature review, in section 5, was that existing alternative ontology management tools in our domain had clear patterns of outdated design elements and user interfaces. Before venturing into the design choices of the requirements, it should also be stated that the general

---

<sup>1</sup><https://www.figma.com>



design of the tool took inspiration from most existing web applications, having the navbar attached to the top of the screen and including most of the core features as tabs to enable simple navigation. The navbar created in the mid-fi prototype is displayed in figure 7.1. As concluded in the previous phase, the lo-fi prototype lacked buttons for log in and log out - which were added in the mid-fi.

The darker color tone of the website is also a design choice for a more visually pleasing experience, given its popularity [20]. Recent studies show that dark mode have drawbacks in reading efficiency [20], but as the application will most likely have to adapt to existing Pinteg applications, the decision of having a darker color tone remained. Web applications often give the user the choice of having a dark- or light theme, but as this wasn't a relevant feature for this thesis the darker tone will be consistent throughout the application.

The Auth0 login integration, discussed previously in section 6, was as earlier stated already designed and implemented. Therefore, our application simply added the integration, and logic to prohibit users from accessing the editing features of the application if not authenticated.



Figure 7.1: Mid-fi navbar.

The first version of the mid-fi prototype was a close to exact implementation of the lo-fi prototype, with some design changes given the result of the previous phase. Otherwise, the features were implemented in the same way, where most effort was put into the search feature and related ontology views.

The initial discussions about ontology definitions began after the usability tests of the lo-fi prototype. As ontology is an abstract word, and not very common, the use of the word would probably create confusion among users. Therefore, the definition of "ontology" used in the application, and its relation to resources, were discussed. The discussion gave the following result:

- An ontology would be defined as a collection of resources.
- The ontology would be the name of the collection, and also the root parent of all the resources in the collection.
- To conceptually map ontologies as collections, they will be referred to as *Projects* in the application, whilst the page where ontologies are managed would be referred to as *My Projects*.

## 7.1.2 Search

In line with the example ontology created for the evaluation of the lo-fi prototype, the search page was populated with wine resources as if the user had searched for "wine". The information displayed in a search result was chosen to be the resource name, description, and parent resource. These resource attributes were believed to be the most prominent information for the user to view at first glance. The search page can be seen in figure 7.2.

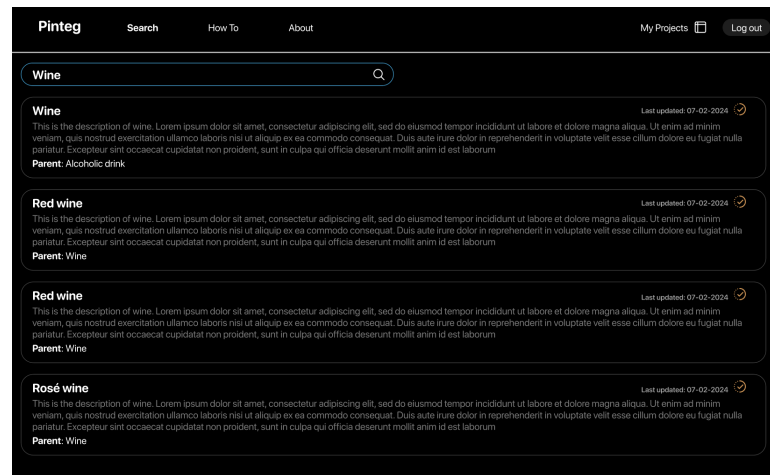


Figure 7.2: Search results when searching for "Wine".

After choosing a resource in the list, the user would be navigated to the resource view, which can be seen in figure 7.3. The view would show fields for description, more detailed information, and a hierarchical view of the ontology. Resource information was split into more and less detailed information, which also presented choices for the user. Therefore, parents, subclasses, and properties were thought to be the most relevant information fields and presented as the "active" tab, called "Classes", in the left box. The other tab, "Details", presented more detailed information that can't be changed by the user, such as creator, creation date, and modification history. The class information parents and subclasses were all clickable and were therefore affordances. To signify the choices, the text added a underline, similar to a link, on hover. The buttons for "Edit" and "Delete" had appropriate colors to signify potentially dangerous actions. The colors were meant to stand out from a page of mostly information, as they were deemed the most likely actions.

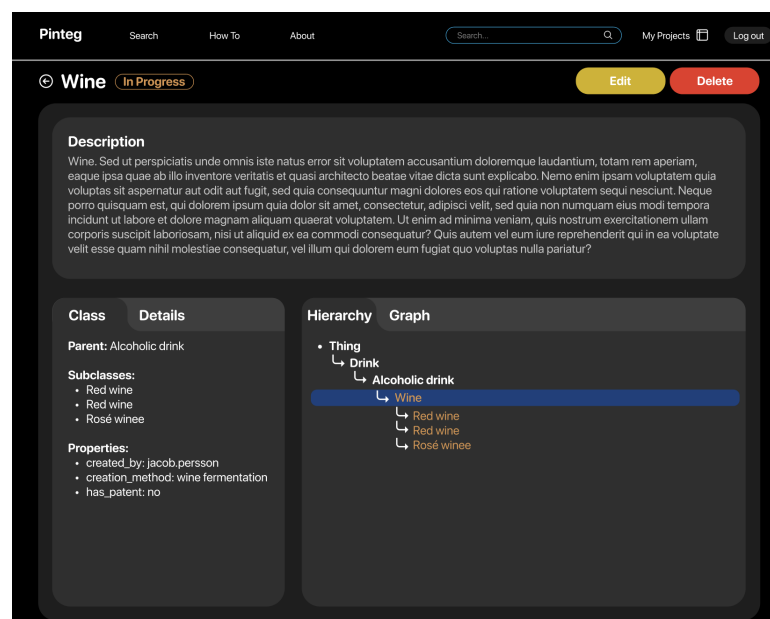


Figure 7.3: The resource view.

The hierarchical view had two tabs, "Hierarchy" and "Graph", which both visually presented the relation between resources in the ontology. The conceptual model of the hierarchy view is a file structure where the user can press a list item to navigate to the resource, similar to the class tab. The graph view shows relations as they would conceptually be for semantic data, where parents, subclasses, and properties were given a property to further explain the relation to the active resource. This feature is similar to existing alternative ontology tools<sup>2</sup>, which all present a knowledge graph. Figures 5.1 and 7.4 display an existing graph and our graph, respectively.

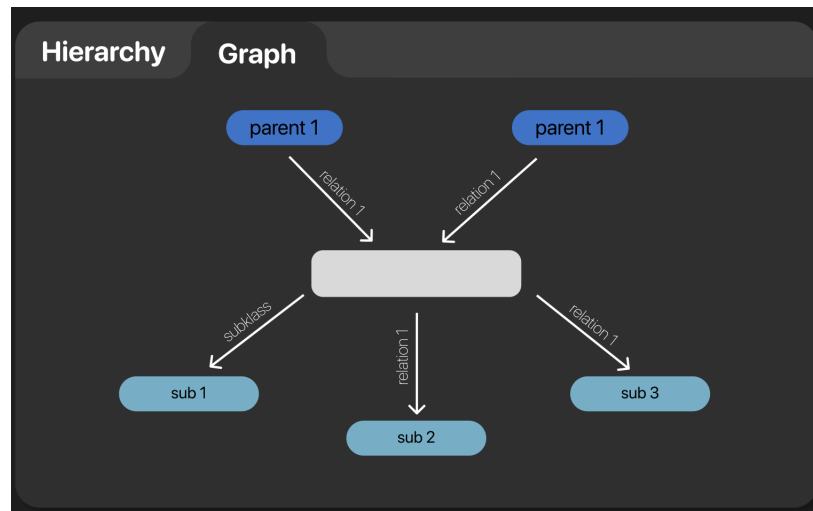


Figure 7.4: Graphical representation of a resource.

### 7.1.3 Manage

To create an ontology, the user has to navigate to "My Projects" to find the creating feature, given they had logged in. The reason for not having a faster creation feature from the search-and view pages were to divide these two features. The user is allowed to view certain pages without logging in, but if the user wants to manage their ontologies they will have to be located on their project page.

As creating, editing and viewing a resource are similar actions, the edit page was created to be similar to the view page, as shown in figure 7.5. A creation is the same as an edit, only the page is not populated with any resource information. The fields for describing the resource in the view-page, were replaced with input fields. The class- and details box is the source of input, whereas the hierarchical box remained as a visualization for the created/edited resource.

The first component introduced in the user projects was the sidebar. It served as an integral part of the page as the management of projects is an important feature, and also a measurement of whether users understand the conceptual model of projects as a collection of resources. A sidebar can be used as a traditional menu, as well as being minimized outside of the user's view, but it was decided that the sidebar should be visible for efficient navigation

<sup>2</sup>Example of an ontology tool with a relational graph from <https://www.ebi.ac.uk/ols4>

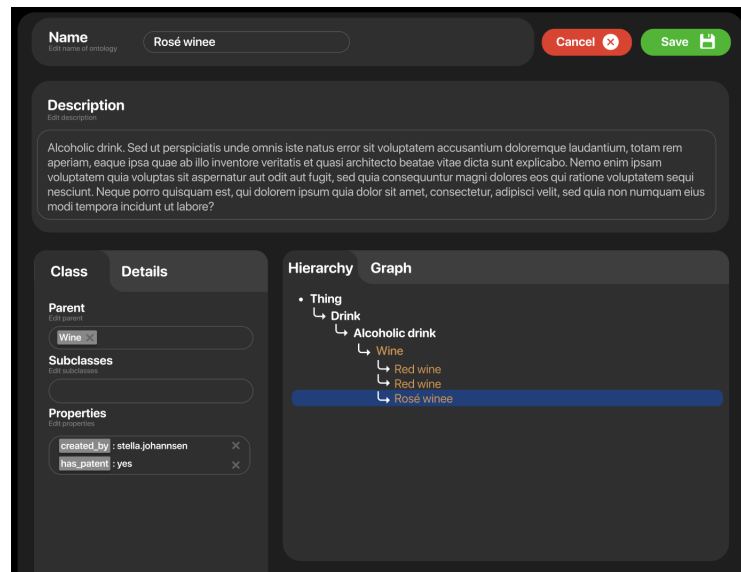


Figure 7.5: The edit-view of a resource.

between projects. The sidebar was placed on the left side because of the F-shape pattern [15]. The title and header names are also placed on the left side of the screen.

The sidebar has three fields, projects, activity, and organizations. These three fields were chosen as they all are important features of the tool. "Projects" are the previously mentioned collections of resources, "Recent activity" shows recent changes made by the user, or members of the user team, to owned or shared projects, and finally a "Team" shows what resources and projects are shared among users in the team. The "My Projects"-page can be seen in figure 7.6.

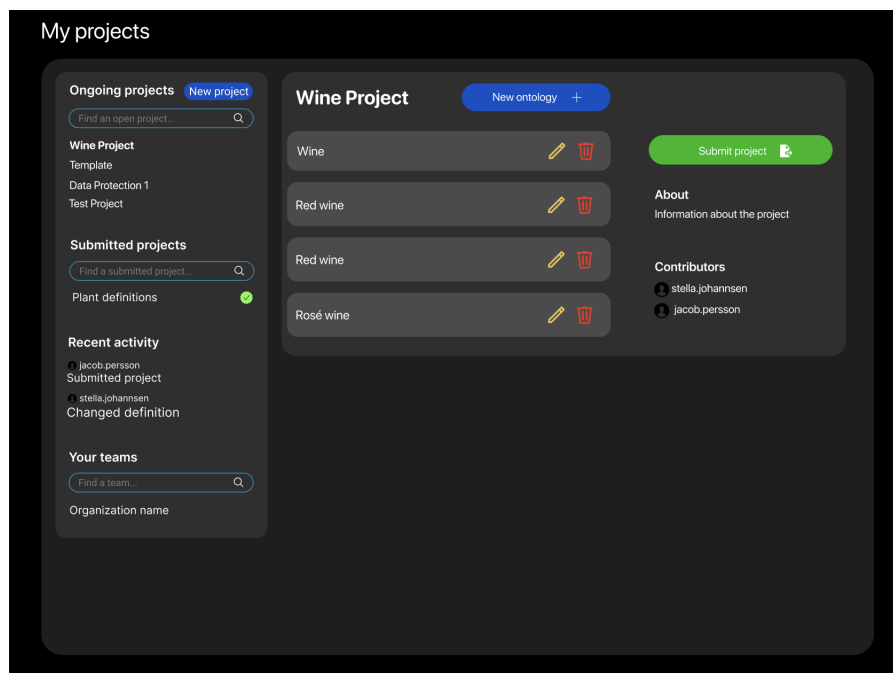


Figure 7.6: "My projects"-view in the mid-fi prototype.

When accessing a project, the resources of the project are listed. Users are given the choice of viewing, editing, and deleting the resources directly from this page. On the page, there is also information on the project, and options to create new ontologies and submit the project for everyone to view.

### 7.1.4 Collaboration

The delete and publish options in the project view prompt a confirmation pop-up to give the user the necessary context that these are dangerous operations, and should therefore be considered before confirming. Colors for dangerous actions are often red, but we chose to use green for the submission as this is a positive action in comparison to the removal of a resource.

After the user publishes the project, a information box was shown for a couple of seconds for the user to get feedback of the submission. Such a box is often referred to as a snackbar. Also, the project view is updated where all actions are removed and instead, the information field is updated with additional details regarding the project.

## 7.2 Usability Testing

In this phase, we conducted more comprehensive testing than for the lo-fi prototype. Test cases were designed to assess both the overall usability of the application and specific functionalities. Additionally, open-ended interviews followed the testing sessions. The same test cases as in the previous usability testing, see section 6.3.1, were used as a base to test the core functionality, but more specific tests were added to evaluate specific functionalities.

The change of defining ontologies as "Projects" was significant in the prototype and it was important to ensure that users understood its purpose and functions intuitively. To achieve this, a test case where users interacted with the project by selecting, editing, adding, and removing concepts, was added. Their reactions and navigation through the project interface were observed to assess its usability and effectiveness. By noting any confusion or issues, areas for improvement could be identified.

Understanding the file system and the tree structure was crucial for users' ability to effectively navigate and manage information in our application. A test case that required users to use the file structure to find specific information was included, as well as test cases where they would visualize data in the tree. By observing their interactions and how they handled these tasks, the usability and clarity of our file/tree structure could be assessed. Any difficulties or uncertainties users encountered during testing were noted.

The test included six participants, four women and two men. Their ages varied from 21 years to 56 years. An important factor was that none of them had any previous expertise in ontology management, ensuring that their assessments and opinions were based on their instincts and understanding rather than pre-existing knowledge in the field.

During the testing process, the think aloud method was applied which allowed for insights into the participants' thought processes while they interacted with the prototype. By carefully observing the participants' interactions, we could gain a deeper understanding of how they used the system, the challenges they faced, and how they attempted to solve them.

This approach helped capture both explicit and implicit reactions and identify any shortcomings or opportunities for improvement in the user experience.

Interview questions were added to ensure that users not only completed the test correctly but also understood its purpose and significance. The questions asked depended on the users' interaction with the product and were designed to capture users' thoughts and reasoning behind their actions. Some examples of questions include;

- Find and describe the different functionalities. When could these be used?
- What indicates that the project is published?
- Which functionality is restricted when a resource is published? Does this feel intuitive?

By listening to their comments and observing their behavior during the test, the questions could be adjusted to address any confusion or uncertainties. This approach allowed for a deeper understanding of experiences and obstacles encountered.

## 7.3 Result and Analysis

This section describes and discusses the results of the mid-fi prototype usability testing, as well as the evaluation with stakeholders. The usability test results are divided into observations and comments, to differ between what was **found** interesting from the observer, and what was **commented** by the tester.

### 7.3.1 Usability Test

The usability test gave interesting qualitative results. The main takeaways are grouped by the source, to differentiate between the perspectives of observer and participant. In this case, the observers had information of how the prototype was implemented, and they could expect some results but may have overlooked something else. If something was overlooked, the results can be compared to the comments to see if the test participant picked up on something that wasn't observed.

#### Observations

- All participants displayed confusion in regards to the context, i.e. ontologies and projects.
- There were instances where participants didn't follow the intended user flow, especially when prompted to navigate to a certain project.
- The mapping between the hierarchy and parent/subclass of the resource created confusion.
- Some participants struggled to navigate between resources when viewing a resource.
- When asked to add a "child resource", participants tried to press amongst the children in the hierarchy, as if to trigger the creation from there.
- The orange color was confusing.

## Comments

- Confusion regarding what ontologies and projects refer to.
- Ontology actions could have icons instead of text.
- Submit-button looked like a save button.
- Ontologies displayed in search are only public ontologies, not drafted.
- More clearly visualize what role the resources have in the project view.
- Participants wanted to be able to edit resources directly from the project view.
- Generally a pleasing design of the web application.

As we wanted to test how the conceptual model was received by users with a more general background, the test script was designed to give a background context regarding ontologies. From the tests, it became clear to us that this context was either not explanatory enough, that the conceptual model was not met, or a combination of both. Either way, most of the subjects did not fully grasp *what* they were doing.

One explanation for this could be that the naming or definition of entities in the tool didn't meet the user expectations. Definitions like "Ontology", "Project", "Parent", "Child" and "Hierarchy" can be hard to understand in the context of ontologies, as the tester is used to encountering the words in other circumstances. Some of the other observations may originate in this problem. For example, a hierarchy may traditionally be mapped to the hierarchy of employee roles in a company, whilst in this context it's the hierarchy of concepts. A proposal to mitigate this issue came from a participant, that suggested to more clearly visualize the role of the resource in the project view by changing the size of the resource item.

There were also some general design choices that were tested to see if they fit the context, but some were expected to fail during this iteration because of time constraints during development. The observation of adding a child resource was interesting because the observation showed that users would like some form of quick action in the resource view, triggering additions from the hierarchy.

The orange color didn't fit this user group. An explanation for this could be that the color represents a dangerous action. The color will be tested again since the color itself should indicate that something is partially complete, but there may be other colors that suit better. The green color is usually used for saving, and it was interesting to see comments on it regarding the submissions. The color was already used in the application upon saving, so for the next iteration of the application the submission color will have to be investigated.

Another interesting comment was regarding the search feature, where one participant expected the feature to only have projects that were public, rather than their own drafted projects. As this wasn't in scope for this iteration, it's something we revisited in later iterations.

The problem of navigating resources probably derives from the links not having enough affordance. The parents and subclasses of a resource can have more visible buttons, rather than plain text.

## 7.3.2 Evaluation with Stakeholders

Together with Pinteg, the prototype and the results from usability testing were analyzed. The results were discussed, and several key improvement points were prioritized. Firstly, clearer definitions of the various states of resources were needed. Additionally, it was concluded that the search function should only include collections. Furthermore, the discussion centered around what a group of resources within a particular collection should be called, and collectively, "Library" was agreed upon. The discussion also included what a Library should include, and how that information should be visualized.

### States

To begin with, we redefined the states, where states refer to the state that a project or resource has. Currently, projects are either public, private, or global, whilst resources are public or private. During the interview, the states were discussed as the current implementation had some constraints. In this implementation, users can create resources and edit them but once they are published, they can no longer be edited. Therefore, it was interesting to test how users expect this to work. Once a resource is created, it is in draft mode. When it is published, the published property of the resource is changed to **true** and it is then not possible to edit it anymore. The use case was:

1. The user is still editing the **resource** and keeps the resource as a draft, and therefore the state is still **private**.
2. Another user in the organization wants to use the **same resource**, but as it is in draft mode, and therefore still **private**, it is not possible for this user to use the resource.
3. The resource must then be published for the second user to use the resource. This will update the visibility to **public**.
4. If the **project** is public, other users on the platform can import the resource and use it for their work, but if the project is private, no other **organization** can see the resource regardless of the resource state.

### Libraries and Subscriptions

To ensure that users would better understand the meaning of a collection of resources, it was decided to change the name from "Projects" to "Library". The new name was chosen after discussions regarding the conceptual model, i.e. how users perceive the word projects. In real life, a library holds many kinds of texts and often group them by subject. An ontology can therefore be perceived as a library. Openness regarding the name was maintained to further improve if better alternatives arose in the future.

During the discussion regarding the development of a library, the primary goal was to clearly define its functions and purpose:

- Libraries function as collections of resources sharing a common theme, such as those applicable to a specific industry.



- Each library possesses a unique name and serves as a means to organize and distribute resources.
- Resources may optionally belong to a single library.
- Every library has an owner which is set at creation and cannot change.
- Libraries can be categorized as either **Public** or **Private**, affecting their visibility.
- Organizations have the option to subscribe to public libraries, facilitating automatic synchronization of published resources.
- Resources from subscribed libraries can be utilized in relation to other resources, including those from different libraries.
- Global libraries automatically synchronize resources across all organizations, rendering them inherently public. Consequently, global libraries cannot be set to private status.

## Cloning

Another functionality that was defined during the evaluation with the stakeholders was resource cloning. This feature became relevant as users would be able to use resources from subscribed libraries. Cloning works as follows:

- Any resource can be cloned which creates a *shallow copy*, which is a new object whose variables are identical to the original resource.
- The newly created resource has a unique id and starts in **Draft** but all referenced resources point to the same resources.

With updated definitions regarding functionality, the project could proceed to the next iteration where development shifted towards a high fidelity prototype.

# Chapter 8

## Phase 4: Hi-Fi Prototyping

---

*For the final phase, the design process transitioned from mid-fidelity to high-fidelity, and from Figma to software development using programming. The software was developed in TypeScript and React, and the choice of framework and language was based on the decision to use modern web application frameworks.*

### 8.1 Design and prototyping

Once the mid-fi designs were finalized and evaluated in Figma, we transformed the static mockups into dynamic, interactive web applications with enhanced functionality and responsiveness. React is arguably the most commonly used library for frontend development [21], and it was decided that the choice of programming language similarly came down to the same conclusion as to the choice of React. JavaScript is regarded as the programming language of the web, but TypeScript was developed to enhance JavaScript rather than to compete with it. Also, TypeScript offers some advantages when it comes to static typing and tool support, which can lead to improved code quality and fewer bugs [22]. The possibility of furthering our experiences with modern frameworks like React and TypeScript was a motivating factor that influenced these choices.

We developed components based on the design specifications created in Figma, ensuring consistency between the visual designs and the final implementation. This involved writing code to handle user interactions, data input and output, and navigation flows.

#### 8.1.1 Overview

In the preceding phase, we came to realize that the discussion surrounding the conceptual model held greater significance than we had initially anticipated. As described in the evaluation of the mid-fi prototype, see section 7.3, our first mitigation was to create an object that could hold resources. This object was called **Libraries**, and were created through a new API

---

endpoint in the backend. A library aimed to hold an entire ontology, and took the place of what was previously called **Projects**. The aim of a library was no longer to be the root of the ontology, instead, a library only holds the chosen collection of resources to group resources of similar subjects together.

As for a real library, it holds many kinds of texts and often group them by subject. Therefore, it was decided that to make use of the metaphor, a **Subject** would make a better definition of an ontology in the application. This change was decided after further discussions with stakeholders, as the tool itself is the library and the object is holding a *subject*. Subsequently, "Parents" and "Children" were changed to **Superclasses** and **Subclasses**, as we placed less emphasis on the parent resource than in the previous iteration.

## 8.1.2 Search

During the hi-fi implementation of the search page, the layout was updated and functionality for subscribing to subjects was added. The search results were changed to only visualize relevant information, which can be viewed in figure 8.1. As the subjects themselves didn't hold any additional information other than name, owner, and visibility, this information was presented in a search result. Each item also contained a button to subscribe or unsubscribe, depending on whether the user was subscribed or not. To clearly distinguish between features, the search page had most of the functionality of subscribing as the managing of subjects was thought of more as an editor. To minimize confusion in these features, they were held on different pages.

An item in the results would only be clickable if the user was either an owner or a subscriber, so to view the subject the user had to first subscribe to the subject. After subscribing, the user would be sent to the *subject* view on the home page, which was previously referred to as the project view. Also, the evaluation of the mid-fi prototype had comments on what subjects should be visible in the search results, as the test participants expected only the public subjects to be displayed. This is something we agreed with, and implemented in the hi-fi prototype.

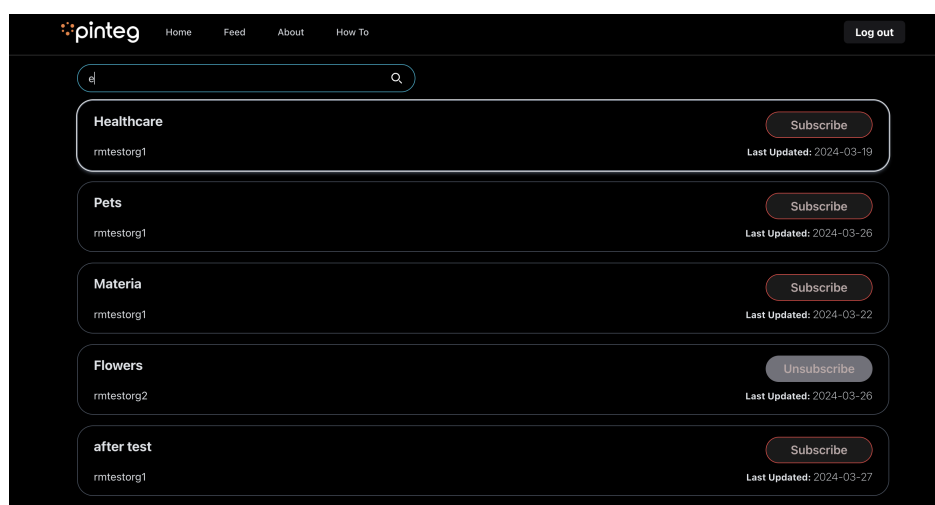


Figure 8.1: Hi-fi search page, displaying search results.

### 8.1.3 Management

The **My Projects** page was given a new definition in the hi-fi prototype, to more clearly describe what the main feature of the tool was. The page was defined **Home** since the page should be the hub for editing and managing all the users' subjects. The Home tab is also more centralized than the previous project tab, which was located on the right of the navbar, see figure 7.1, indicating the purpose of the feature.

The sidebar was also changed, shown in figure 8.2, to be more straightforward. Under *Subjects*, a list of all user-owned subjects is shown, and *Subscriptions* displays all subjects the user is subscribed to. After discussions with the stakeholders, a feature to see all user-owned resources was also added, at the bottom of the sidebar, for the possibility of seeing all resources the user has access to.

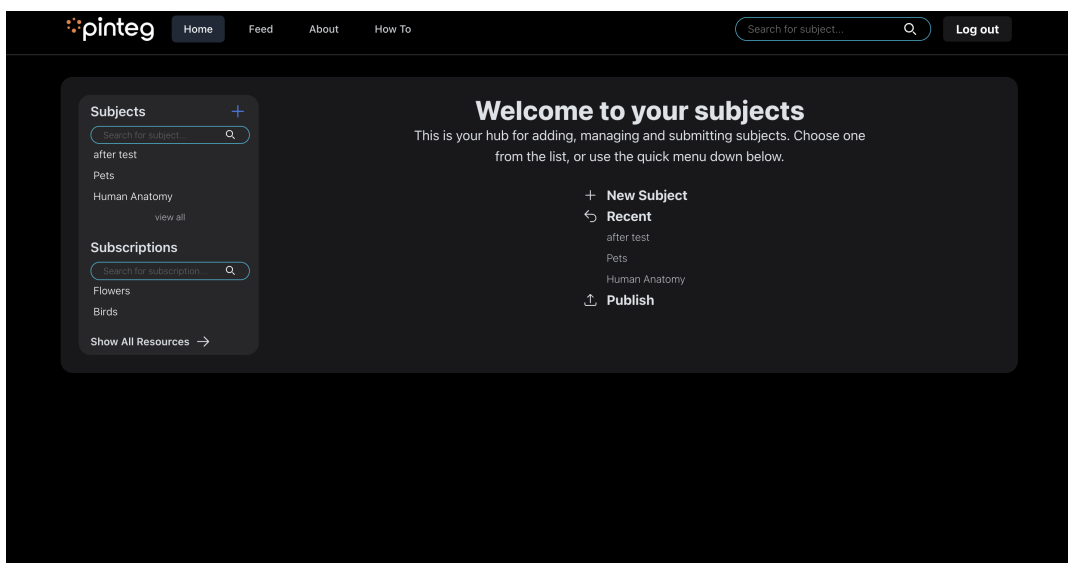


Figure 8.2: Hi-fi home page.

The sidebar has a single plus-icon, and as the sidebar manages subjects the button displays the feature of creating a new subject. The modal presented, shown in figure 8.3, has two fields, name, and visibility, to simplify the creation. The visibility input could be "Public" or "Private". On save, the new subject will be displayed in the sidebar.

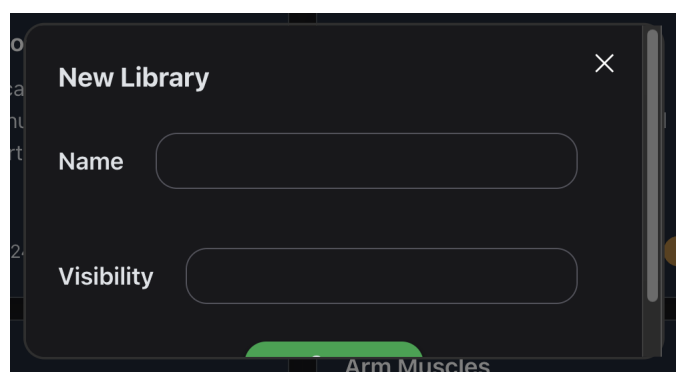


Figure 8.3: Modal for creating a new Library.

The view of the public subject "Human Anatomy" can be seen in figure 8.4. When viewing public and private subjects, there are some differences in available actions. As described in the analysis of the previous phase, see section 7.3, a public subject could as of now not be deleted because subscribers could depend on its availability. Therefore, on deleting a public subject, the user would be prompted with a dialog saying "this action is not possible".

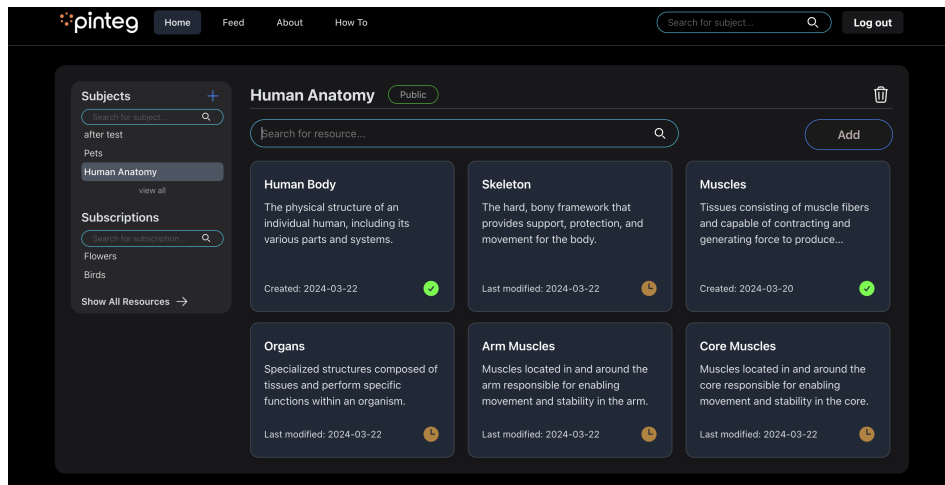


Figure 8.4: Hi-fi subject-view of a public subject.

On deleting a private subject, for example **School** shown in figure 8.6, a pop-up would be displayed. The pop-up, visualized in figure 8.5, informs the user regarding the consequences of the action. To handle potentially dangerous actions, there was a dialog created with a button that was outlined in red, and on hover it turned red. This dialog was used in dangerous actions across the application but with different titles and text depending on the action. Other actions that had the dialog were publishing and deleting resources.

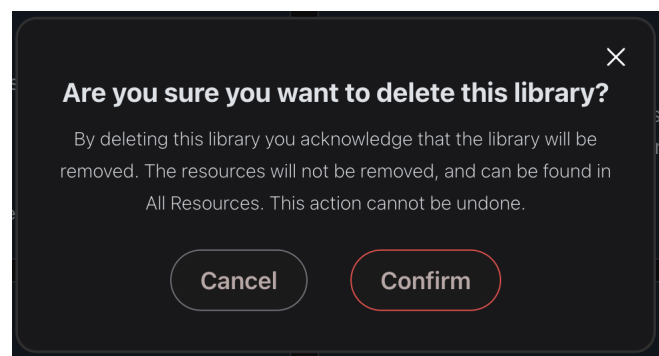
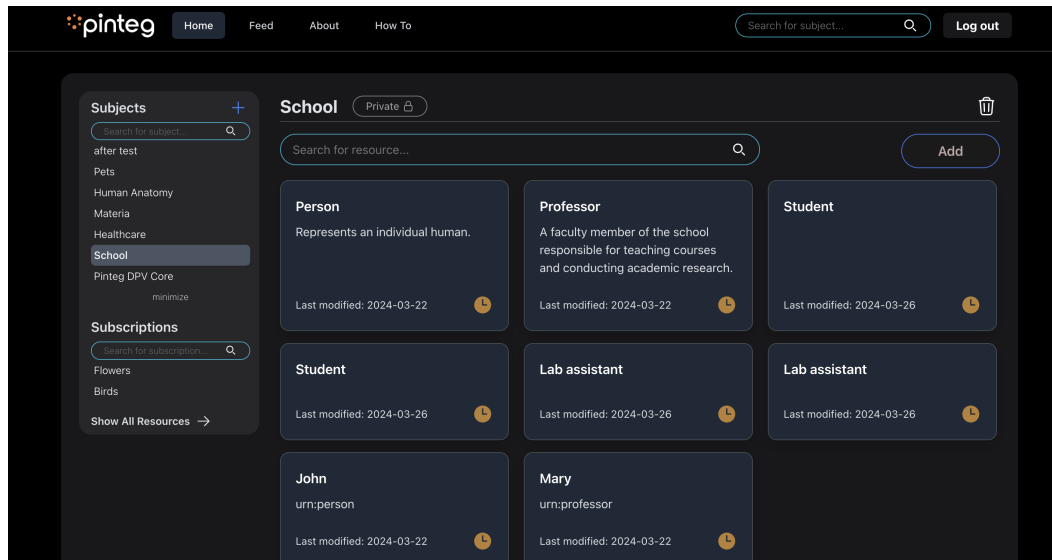


Figure 8.5: Confirmation dialog on delete.

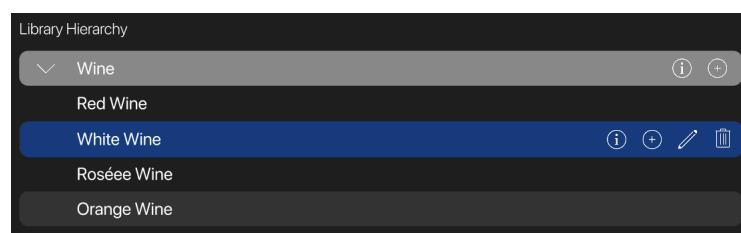
The design of displaying a resource changed to boxes, as this would display more information about each resource and utilize more space on-screen as the resources are regarded as the most important objects of the subject collections. From the evaluation in the third phase, participants found it confusing to see the relations between resources in an ontology. In this prototype, we thought of the subject more as a collection, and the relations between resources were not as important to visualize. Even so, we kept the takeaways from the previous evaluation in mind for the usability test in this phase.

The boxes has icons for the visibility of each resource. This can best be seen in figure 8.4, where both icons are displayed. The clock is meant to show that the resource is still being edited and is therefore still in draft mode. The orange color was chosen to signify that the resource is not done yet. The check and the green color combined show that the resources are "done" and can be viewed by anyone, if the subject is public.



**Figure 8.6:** Hi-fi subject-view of a private subject.

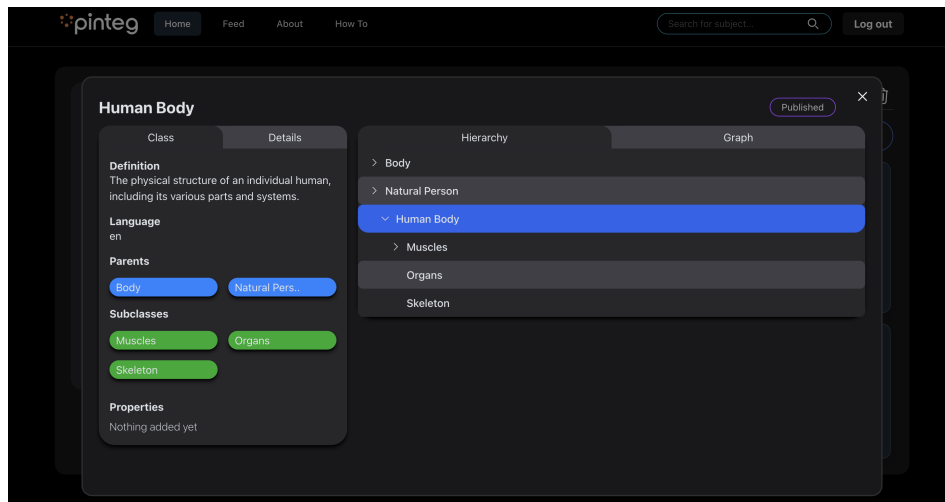
Another topic that was observed in the evaluation of the mid-fi prototype was the actions available for resources. Some testers expected there to be more quick actions to create, edit, and delete. This was something that was already thought of in the previous prototype, but it was decided that it wasn't necessary to allocate time for this feature. In this phase, we similarly decided that this feature would have to wait, but we had some ideas of how to implement it. In Figma, we had early designs on what that would look like, see figure 8.7, but as we decided to go for boxes the feature of quick actions would have to wait for future iterations. In the boxes, there is space to add icons for such a feature.



**Figure 8.7:** Mid-fi implementation of quick actions.

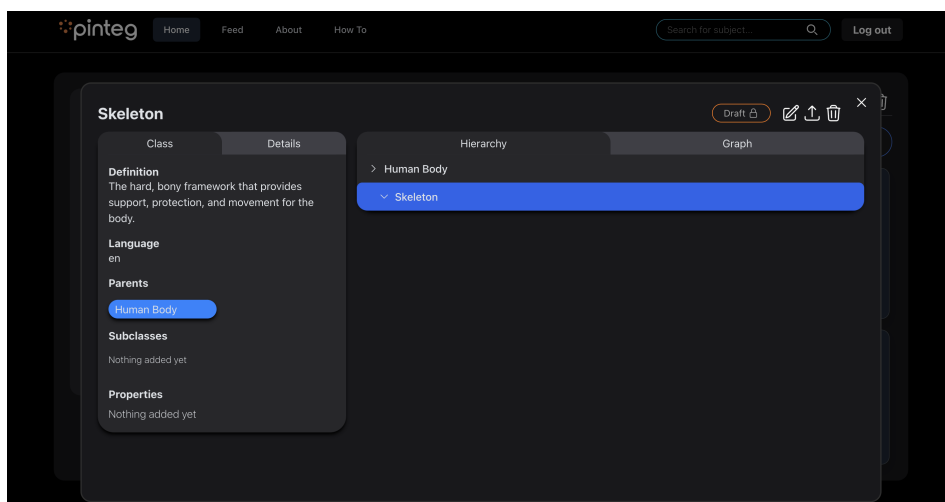
When navigating to a resource, we decided that a pop-up of a similar page as the resource view in the mid-fi prototype, see figure 7.3, would indicate that the user was still in the subject rather than on a new page. The modal can be seen in figure 8.8. Otherwise, the different fields and boxes were the same, or very similar. The definition was moved to the class box, as descriptions were believed to be of less importance than the visualization of relations. The class box had some new designs of super- and subclasses, as these seemed difficult to find for

test participants in the mid-fi prototype. The idea was to make them more button-like, and therefore signify that it was possible to navigate the classes.



**Figure 8.8:** Hi-fi resource-view of a published resource.

The hierarchy was also updated from the previous phase. Firstly, it should be visible that the active resource is the resource which is the main element of the hierarchy. Every resource above the active resource are superclasses, and every resource under is a subclass. Similar to the library hierarchy displayed in figure 8.7, we wanted it to be possible to toggle each resource, to view its subclasses. Therefore, we decided to remove the arrows from the mid-fi prototype, see figure 7.3, and chose icons that only had the head of an arrow. When toggled, they change angle to point down to signify that the list is expanded. Even if the previously used icons may be clearer, we thought the toggle would be more understandable with the arrow icons. We also compared the hierarchy view to other similar services, particularly applications that have some form of toggle function. Applications commonly known such as *My Documents* on Windows and *Finder* on Mac both have similar functionalities when navigating files, and we wanted to create a similar feel to how the hierarchy worked.



**Figure 8.9:** Hi-fi resource-view of a private resource.

The view of a resource in draft-state was updated after taking observations from the evaluation of the mid-fi prototype into account. The view is shown in figure 8.9. Resource actions were changed to icons, as this would divert attention from the resource view and visualizations. Similarly to the explanation for not including quick-actions in the resource boxes, it was also decided that this feature had to wait as it would take time from the development.

In the following two figures, 8.10 and 8.11, one can see that the same pop-up has been used, but the edited resource information is filled in. The pop-up is designed to be similar to the resource pop-up, but give direct feedback upon save. To request a save in the backend, some fields had to be inputted and if these were not inputted the color around the filed would display red and present an error message. The visualizations, hierarchy, and graph, are updated with the information inputted.

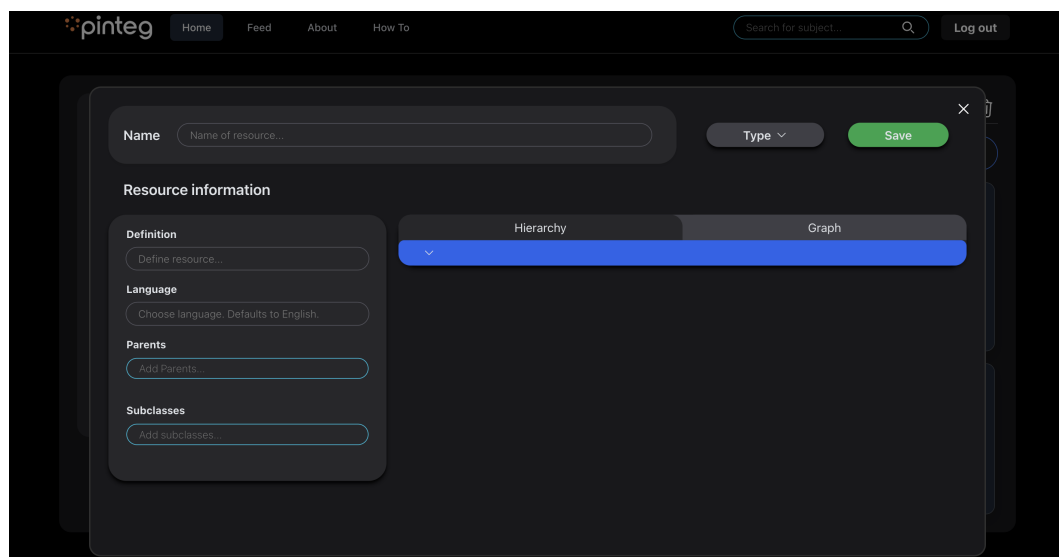


Figure 8.10: Hi-fi resource creation-view.

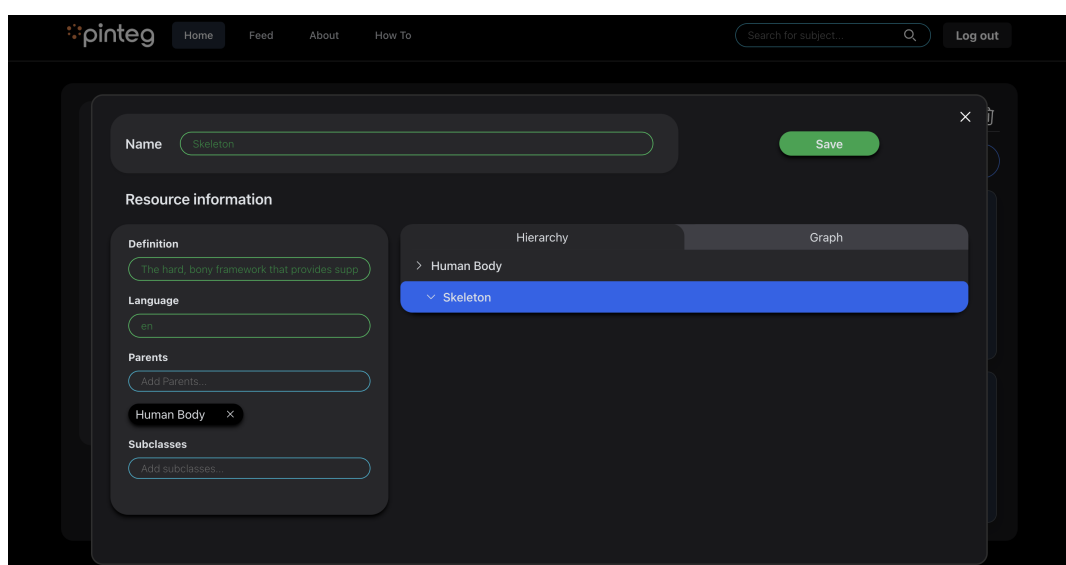


Figure 8.11: Hi-fi resource edit-view.



The graphical visualization of the resource relationships, figure 7.4, was implemented as an alternative to the hierarchical view, and to display the relations in more detail. The idea was also to introduce the resource properties in this view, but as properties weren't added to the backend in time, the graph shows the same thing as the hierarchy. The idea of the graph is to display the properties in the same way as the super- and subclasses, but have some difference in the sense of color and positioning, of the resource.

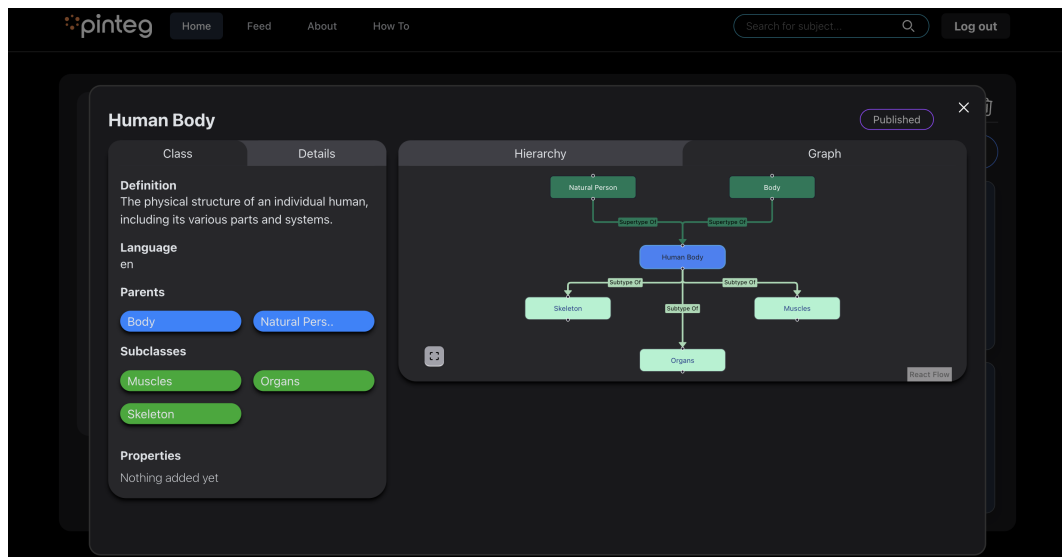


Figure 8.12: Hi-fi graph-view in a resource.

## 8.1.4 Collaboration

As new requirements were established in the stakeholder interview in section 7.3.2, two new features were considered in the hi-fi prototype. As stated, the search- and home page considered subscriptions, a way of integrating community functionality into the tool. The idea was for organizations to be able to take inspiration from existing alternatives when implementing their subjects. This feature was also aimed to create standards in data protection and privacy-related subjects. As described in section 1.1.1, a company in a certain industry could look at how other companies employee payrolls processes. Instead of all the companies implementing their process, a standard payroll process could be created and stored in the community. Companies can then import the process and instantiate it with their modifications. The idea comes from the saying, "Don't re-invent the wheel".

In the process of creating a community, the feature of importing resources from subjects that one is subscribed to requires a way of importing and versioning, resources. The feature of cloning, described in section 7.3.2, is a way of importing an existing resource by cloning it. The copy created is private to the importing user and can be modified without creating issues with the original resource as it is a *new* resource. In the current implementation, the import button will import a resource into the active subject by cloning the resource. For the evaluation of the hi-fi prototype, we intend to question test participants about what they think of the feature, and what they expect happens to the imported resource.

## 8.2 Usability Testing

After the hi-fi prototype development, we conducted testing to validate the functionality, usability, and performance of the application.

The purpose of this test is to evaluate the effectiveness, usability, potential usability challenges, and overall satisfaction of an ontology management tool among users with limited knowledge of semantic modeling. By conducting this study, the aim is to assess the tool's intuitiveness, and its ability to facilitate the creation and management of ontologies within knowledge graphs, as well as to identify any usability issues that arise during the interaction. Additionally, the evaluation will focus on understanding how the logic behind the management of ontologies, concepts, and instances is structured, and if it is to gain a comprehensive understanding of the tool's functionality. The results from this test will inform improvements to the tool's design and functionality, ultimately enhancing user experience and usability.

### 8.2.1 Evaluation Questions

The questions below aim to explore different aspects of the tool's performance and user experience, with a particular focus on investigating the previously identified problematic areas. By addressing these questions, we strive to gain a deeper understanding of the tool's functionality and potential areas for improvement.

1. To what extent does the tool facilitate the creation and management of ontologies within knowledge graphs?
2. What are the main usability challenges encountered by users when interacting with the tool?
3. How does the underlying logic for managing ontologies, concepts, and instances affect users' ability to effectively create and manage ontologies within the tool?
4. How satisfied are users with the overall user experience and functionality of the ontology management tool?

The first question examines how well the tool helps users create and manage ontologies, focusing on the tool's ability to support users' tasks, utility, and how effectively users can perform these tasks, effectiveness. The second question identifies the main usability challenges users encounter, particularly regarding how easy it is to learn and use the tool, learnability, how safe it is against errors, safety, and how well the tool supports users' workflows, effectiveness. The third question analyzes how the tool's underlying logic affects users' ability to effectively create and manage ontologies, which involves how well the technical design supports users' needs and tasks, utility. Finally, the fourth question evaluates the overall user experience and functionality of the tool, focusing on users' satisfaction with how well the tool supports the effectiveness of their tasks, and how easy it is to learn to use the tool effectively.

## 8.2.2 Participants

Eight participants, comprising 5 women and 3 men aged 19 to 25, were selected to conduct the test. This decision aimed to ensure adequate data collection within the specified time frame and resource constraints. Additionally, this number facilitated in-depth observations of each participant's interactions with the tool, enabling the identification of potential usability issues.

Due to constraints in accessing future users, the selection of test participants aimed to include a suitable group for the study, based on availability rather than preference. However, it's important to consider that the participants' background in computer science can introduce biases into the study. Their understanding of hierarchy structures and other technical concepts may not align with those of the intended users, potentially influencing the test results. Nevertheless, they were chosen due to their availability and capacity to generate substantial results, given their familiarity with usability testing and methods like think-aloud protocols.

## 8.2.3 Data Gathering

By collecting a variety of data, a comprehensive and versatile understanding of the user experience and the tool's performance can be obtained. Each type of data complements the others to provide a complete picture of how participants interact with the tool and experience its functionality. The different data that were collected during the evaluation are stated below:

- Task performance data
- Observational notes and video recordings
- Semi-structured interview answers
- System Usability Scale (SUS) scores

Task performance data provides us with objective quantitative measurements of users' efficiency and performance, while observational notes and video recordings offer detailed insights into their behavior and reactions, which are both objective and quantitative. The semi-structured interviews conducted during and after the test allowed us to capture participants' subjective opinions and feedback about their experience, while SUS scores provided quantitative subjective data.

The test cases, presented in table 8.1, primarily focus on evaluating the interface and user interactions within the ontology management application. These test cases evaluate whether users understand how various tasks, such as creating and editing ontologies and managing resources within existing topics, should be performed. By testing these actions, we can assess the usability of the interface and how effectively it supports users in their workflow.

**Table 8.1:** The test cases for the usability test.

<b>1. Test Cases for Creating a New Ontology</b>	
1.1	Login and access your page
1.2	Create a new ontology
1.3	Create a new resource
<b>2. Test Cases for Editing a Subject</b>	
2.1	Navigate to an existing subject
2.2	Create a new resource (concept) in an existing subject
2.3	Edit an existing resource (concept)
<b>3. Test Cases for Collaborating with Team Members</b>	
3.1	Publish a resource
3.2	Visualize the details of a resource
<b>4. Test Cases for Collaborating with other Organizations</b>	
4.1	Find a public subject in your Home-page
4.2	Use the hierarchy and graph to find information about the subject
4.3	Search
4.4	Subscribe
4.5	Import a resource from subscribed subject
4.6	Unsubscribe from used subject
4.7	Unsubscribe to unused subject

The interview questions, as shown in table 8.2, are intended to complement the test cases by examining users' understanding and perceptions of various features and processes within the application. These questions were used as a guide to gain insight into users' expectations, any confusion, and their experience of the application's features and logic. The decision to conduct semi-structured interviews after the test was motivated by the need to capture participants' subjective opinions, feedback, and deeper insights into their experience.

**Table 8.2:** The interview questions defined for the usability test.

Test Case	Interview Questions
1.1	What were your initial expectations when you clicked on the login button? Did you find the login process straightforward? Can you describe any difficulties you encountered?
1.2	Did you find the "Home" section easily accessible? What do you think a "Subject" is? What form fields were you expecting when creating a "Subject"? Did you encounter any confusion?
1.3	What is a resource? Were the form fields intuitive when filling out the new resource information? Did you encounter any challenges while saving the new resource? Does it feel intuitive to create a new resource inside of a library?
2.1	Did you find the navigation to the subject straightforward? What improvements, if any, would you suggest for navigating to your existing subject?
2.2	Can you explain the process you followed to create a new resource within an existing subject? Were the options for selecting the resource type clear and understandable?
2.3	How did you initiate the process of editing an existing resource? Did you find the navigation to the edit-button intuitive? Were the editing options clear and accessible? Did you encounter any challenges while saving the edited resource? Does it feel intuitive to be able to edit a resource?
2.4	How did you select the parent resource for the given resource in your new subject? Were the options for selecting the parent resource intuitive? Was it clear that you could access a resource that you created in a different library? Can you describe why this feature could be necessary?
2.5	Were the functionality for importing and editing the resource clear and straightforward? Can you describe why this feature could be necessary?
3.1	What were your expectations when you decided to publish a resource? What did you think it meant to "Publish" a resource? How important do you think it is for team members to be able to publish resources easily?

Test Case	Interview Questions
3.2	Before accessing the details of a resource, what did you expect to find? Did the details of the resource meet your expectations? Why or why not? How useful do you find the tab for visualizing the hierarchy? How useful do you find the tab for visualizing the graph? Which of them would you prefer to have? Why? When? Do you think that any additional details would be valuable to include in the resource details.
4.1	Did the search results meet your expectations? Why or why not? How important is it for you to be able to find relevant resources from other organizations through search? When would you use the search feature?
4.2	Were the subscribed libraries what you expected to find in your search? How valuable do you find the ability to subscribe to libraries from other organizations? What are the reasons to subscribe to a library?
4.3	How important do you think it is to be able to import resources from other organizations into your own library?
4.4	What benefits do you expect to gain from making an ontology public? Are there any concerns you have about making an ontology public? What are they? How do you think making ontologies public can impact collaboration with other organizations?
4.5	What were your expectations when you decided to unsubscribe from a library from which you imported a resource from? Did the result of unsubscribing meet your expectations? Why or why not? What were your expectations when you decided to unsubscribe from a library you have referenced a resource from? Did the result of unsubscribing meet your expectations? Why or why not? What do you think happened to the resources in the subscribed library? Do you think you still have access to them? Is it intuitive that you can subscribe from a library from which you have used resources?
4.6	What were your expectations when you decided to unsubscribe from an unused library? Do you have any specific reasons or criteria used to determine which libraries to unsubscribe from?

The SUS questionnaire, shown in Table 8.3, was used to quantitatively assess participants' perceptions of the ontology management system's usability. By incorporating SUS into the evaluation methodology, we aimed to gather standardized and quantitative data on participants' perceptions of the system's usability.

**Table 8.3:** Each statement within the System Usability Scale.

Nbr	Statement
1	I think that I would like to use this system frequently
2	I found the system unnecessarily complex
3	I thought the system was easy to use
4	I think that I would need the support of a technical person to be able to use this system
5	I found the various functions in this system were well integrated
6	I thought there was too much inconsistency in this system
7	I would imagine that most people would learn to use this system very quickly
8	I found the system very cumbersome to use
9	I felt very confident using the system
10	I needed to learn a lot of things before I could get going with this system

By combining different types of data, we can identify both objective and subjective, as well as quantitative and qualitative aspects of the tool's performance and user experience. This comprehensive approach provides us with a solid foundation for formulating conclusions and recommendations to improve the tool's design and functionality.

## 8.2.4 Procedure

The test included two different roles: a moderator and an observer. The moderator's tasks included the welcoming of participants, explaining the purpose of the study, introducing tasks, answering questions, and conducting the interview. The observer's task was to observe and document participants' actions, behaviors, and comments during the test.

The sessions were conducted in a quiet room, and the equipment needed included two computers, one able to record the participants' actions, and one with the software for testing.

Before each test, the participant gave consent to how the results would be handled by filling out a form. At the beginning of the session, the participants received a brief orientation about the purpose of the test and a paper with definitions, i.e. ontologies, concepts, and instances, and a context of the test, these definitions can be viewed in appendix B. Later, the participants performed the specified tasks and answered the interview questions after the

tasks were done. This was while their actions were observed and recorded. After the test, the SUS form was filled out.

## 8.2.5 Result handling

Guidelines regarding how to handle the results and data were defined:

- Results will be reported through a comprehensive analysis of task performance, questionnaire responses, and SUS scores.
- Video-recorded material will be securely stored and only accessed by us.
- Participants' identities and personal data will be protected by ethical guidelines and data protection regulations.

## 8.3 Results and Analysis

This section presents the outcomes of the hi-fi prototype usability testing, together with a short evaluation of the prototype with the stakeholders. The analysis includes the users' experiences and challenges encountered while interacting with the ontology management tool.

### 8.3.1 Usability Test

The results of the usability testing have been categorized according to the evaluation questions defined in section 8.2.1, to provide a more structured overview of users' experiences and challenges with the ontology management tool.

#### **1. To what extent does the tool facilitate the creation and management of ontologies?**

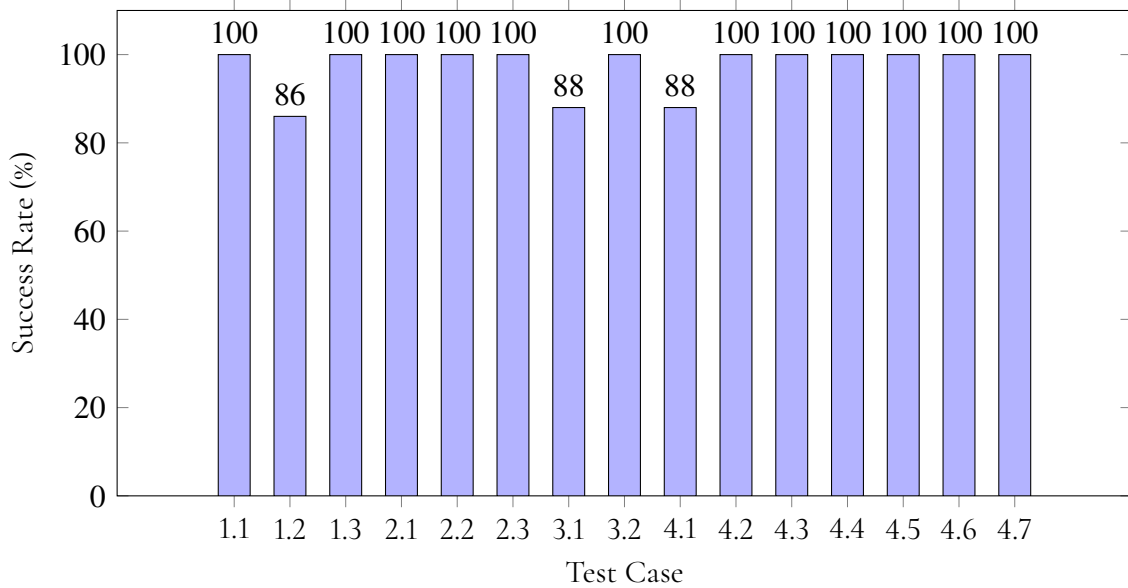
The extent to which the tool facilitates the creation and management of ontologies was examined. The tests revealed that the application performed well from that perspective, as most users were able to complete all test cases. The success rate for each test case is displayed in figure 8.13.

The results show that the tool enables the creation and management of ontologies efficiently, with a high success rate for most test cases. Despite some minor challenges, such as a slightly lower success rate in creating a new ontology and publishing a resource, the results indicate that users were able to complete the tasks correctly.

#### **2. What are the main usability challenges encountered by users when interacting with the tool?**

The main usability challenges that users encountered when interacting with the tool were analyzed. Users had difficulties understanding the navigation during login due to a lack of clarity regarding the **Home** page. Users suggested that the page should be the default destination after logging in and that the label "Home" should be clarified to reduce confusion.





**Figure 8.13:** Success Rate for each test case.

The interface's lack of clarity confused users, notably during the creation of new subjects. The save button was only visible if the user scrolled in the modal. We suspect this might be due to our screen recording using Zoom because when testing the application before the usability testing, this issue did not occur. Therefore, participants struggled to find the button. Another deficiency that users experienced in this modal was the inefficient way of choosing between making a subject "public" or "private". Instead of offering a clear and user-friendly interface design where users could easily select the desired state by clicking on a checkbox or selecting the state from a drop-down menu, they were forced to type the word "private" or "public" into an input field.

When it came to creating a new resource, users encountered several obstacles that made the process unnecessarily complicated. One of the most notable problems was the difficulty in identifying which fields were mandatory to fill in. Although the system had functioning error handling, where users received warnings if they tried to save the resource without filling in all the necessary fields, there was still a desire for clearer marking of these mandatory fields to avoid uncertainty.

Users also provided feedback on various aspects of the interface, including navigation, button design, and color usage.

The results indicate that there are several areas where small improvements can be made to enhance user-friendliness. Among the identified issues are confusion regarding navigation, lack of clarity in the interface, and difficulties in understanding certain functions. Incorporating clearer feedback to users and unifying the button design, could help reduce these challenges and increase user satisfaction with the tool.

### **3. How does the underlying logic for managing ontologies, concepts, and instances affect users' ability to effectively create and manage ontologies within the tool?**

Challenges arose with the tool's logic for handling ontologies, concepts, and instances. Additionally, difficulties in understanding the connection between ontology elements led to

problems with organizing ontologies within the tool. An example of this was that users had difficulty understanding the meaning of publishing a resource. Some users confused the publishing function with the sharing function and associated it with the resource becoming visible to other users. There was also confusion about the concept of publishing a resource on a subject and whether it would affect its availability to other users. The lack of clear feedback after publishing a resource made it difficult for users to understand if the action had been performed correctly. Users also had trouble understanding how subscriptions to subjects worked and their consequences. Some users felt they should be able to see the content of a subject before deciding to subscribe to it. There was also confusion about what would happen to the subscription after unsubscribing from a subject and whether the resources would remain available to the user.

When creating a new resource, another challenge arose, namely the ambiguity surrounding the choice of "Type" and "Parent" for the new resource. Many users did not understand the difference between the various options for "Type", i.e., whether the resource was a concept or an instance, but the biggest problem was how to choose an appropriate "Parent". The majority of users assumed that the name of the subject the resource was created in would be the parent's name, leading to incorrect choices and misunderstandings about how the resource would be structured within the system.

When it came to publishing a resource, users encountered several challenges and confusion about what it meant for a resource to be in "draft" or "published" mode. Many users found it difficult to understand the difference between these two states and how it affected the visibility and availability of other users. Some users believed that a published resource automatically became visible to everyone, regardless of whether the subject it was in was private or public. Others believed that it was only visible to themselves, but that it was now published, which meant that it could not be modified and therefore could be referred to. This lack of understanding led to confusion and uncertainty about who had access to the resource and how it could be used. Another problem highlighted was that there was no option to edit a resource after it had been published. Although many users understood that a resource should be immutable to be used as a reference, frustration was created as users could not correct any errors or update the information in a published resource, limiting their ability to maintain and improve their resources over time.

The terms "public" and "private", describing the state of a subject confused users. The main issue lay in users not fully understanding the implications of a subject being "private" and its consequences. Many users argued that the term "private" was misleading as it did not fully explain who had access to the subject and its content. There was a tendency to associate "private" with personal access rather than limited access for a specific group or organization. Other users thought it had some form of restriction on who could see or use the resources. This led to inconsistent expectations and misunderstandings about the rights and permissions that would apply to subjects in different states.

Many participants expressed dissatisfaction with the subscription feature, primarily because they couldn't view what they were subscribing to unless they had already subscribed. This lack of visibility was a point of frustration for users who wanted more transparency before committing to a subscription. Furthermore, testers generally compared the subscription functionality to typical social media subscriptions, indicating a desire for a similar user experience. They also noted the absence of information about the owner of a subject when viewing it, which could impact users' trust and understanding of the content. Regarding im-

ported resources, testers appreciated the ability to import them as a way to avoid duplicating work. However, there were concerns about the lack of clarity regarding what happens to imported resources, particularly when unsubscribing from the original subject. Testers felt that imported resources should remain accessible even if the original subject is unsubscribed, but there were differing opinions on whether references to those resources should be removed upon subscription.

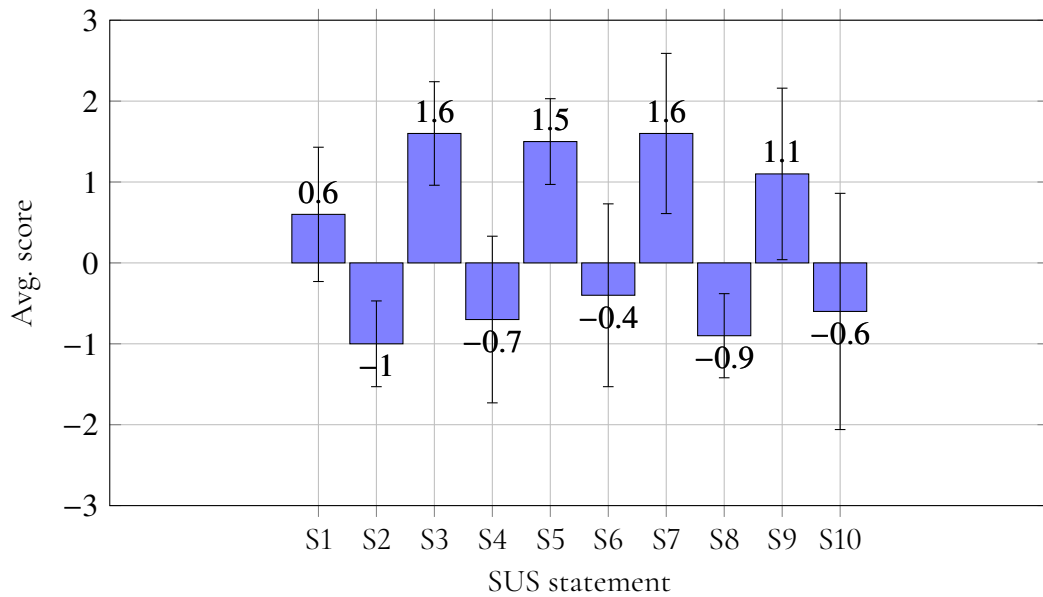
#### 4. How satisfied are users with the overall user experience and functionality of the ontology management tool?

Based on observations and interviews, we found that users appreciated the visual appearance and user-friendliness of the ontology management tool. They found the interface appealing and that desired features were available. Testers also expressed appreciation for the role of the import function in avoiding unnecessary duplication of work and explained that both hierarchical and graphical views were useful for different purposes. The graphical view, in particular, was praised for its clarity and simplicity, facilitating users' understanding of the data. However, users struggled to understand exactly what they were doing within the tool. From the SUS survey, table 8.4, we obtained a score of 75.3. We can, in figure 3.1, see that a score of 75.3 correlates to a "Good" and "Acceptable" application.

**Table 8.4:** Participants' average score on each statement from the System Usability Scale.

Nbr	Statement	Average
1	I think that I would like to use this system frequently	3.1
2	I found the system unnecessarily complex	1.5
3	I thought the system was easy to use	4.1
4	I think that I would need the support of a technical person to be able to use this system	1.75
5	I found the various functions in this system were well integrated	4.0
6	I thought there was too much inconsistency in this system	2.1
7	I would imagine that most people would learn to use this system very quickly	4.1
8	I found the system very cumbersome to use	1.6
9	I felt very confident using the system	3.6
10	I needed to learn a lot of things before I could get going with this system	1.9
<b>Total SUS Score</b>		<b>75.3</b>

This result demonstrates an overall positive perception of the tool, particularly regarding



**Figure 8.14:** Avg scores and standard deviations for the each statement.

its visual appearance and user-friendliness. Although users appreciated certain features, such as the ability to import resources and use graphical views, the SUS survey indicates that there is room for improvement in terms of user experience.

Figure 8.14 presents the avg statement score for each statement. The values are shifted so that the statements that expect a lower score are displayed as negative numbers. From the diagram we can see that the average scores for statement 1, 4, 6 and 10 had the worst scores.

The standard deviation was largest for statements 4, 6 and 10 which can suggest that the participants did not understand the statement or had a hard time to apply the statement to the product. Although these three statements have great deviations, they concern areas that are of interest. Technical help, inconsistency and learning are all areas that allude to our previous issues with the conceptual model.

Statement 1 had a lower deviation, which indicates that the participants believe that they won't use the system frequently. This is expected as the participating test group wasn't the expected users.

### 8.3.2 Evaluation with Stakeholders

Together with Pinteg, the test results were evaluated. During this evaluation, it was early concluded that the findings from the third research question, addressing the user's comprehension of ontology management logic, provided a clear insight into the current issues in their back end. Solutions to these problems will be discussed in chapter 9.

Additionally, the specific challenge of applying ontologies in the domain of data protection was highlighted, in contrast to other fields like biology and medicine, where established tools are available. It was noted that differences in the definitions of concepts across various subject areas could impact the implementation of ontologies in data protection.

# Chapter 9

## Discussion

---

*The discussion cover general takeaways, what went well and what could have gone better. It will also discuss where we ended up, the final prototype, and evaluate the findings from the final usability test, described in section 8.3. Finally, we will compare our findings to the research questions of the thesis.*

### 9.1 Overview of Design Process

In this overview, we will explore the different phases of the process and analyze insights discovered along the way. By examining each phase and its significance, an understanding can be obtained of how the design process has shaped and refined our work to meet the needs and expectations of users.

In the first phase of the design process, personas were created to gain a better understanding of user needs and goals. It provided a structured starting point, but it was challenging to identify user needs because there was a lack of a clearly defined target audience. This limited the ability to create relevant personas, which in turn affected the initial design choices. With perspective from the later stages of the design process, it can be concluded that the limited understanding of users in the conceptual design phase may have negatively impacted the product's development. Without a clear target audience, there can be a misunderstanding of what knowledge a user would have, which may have resulted in a less user-friendly product. This underscores the importance of carefully researching and defining the target audience at the beginning of the design process to ensure that the product effectively meets users' needs and expectations.

During the implementation of the first prototype, usability tests were conducted with an early lo-fi prototype to quickly iterate and improve the design. Conducting tests early provided valuable feedback and the opportunity to identify areas of concern before progressing too far into the design process. However, only four tests were conducted, all with women from the same institution. This lack of diversity in the test group could have affected the ability to uncover all potential user needs and challenges. During usability testing, several

strengths and weaknesses of the prototype were identified. A good user flow and clear navigation were noted as positive aspects. However, participants struggled to understand the difference between an ontology and a concept, indicating the need for clearer explanations of these terms. Because the test participants did not belong to the intended user group, their lack of understanding was assumed partly to be related to this factor.

In the third phase of the design process, focus shifted to more conceptual design, given the analysis of the previous evaluation. As discussions with Pinteg were ongoing throughout the project, the stakeholders agreed that it was in their interest to evaluate the existing backend solution. This shift in focus of design work meant that there was a need to reassess previous assumptions and adapt design choices to better integrate with potential changes that could occur on the backend side. A positive aspect of this change was that it allowed for a deeper exploration of the underlying logic of the system. By understanding and testing the logic, a more coherent and user-friendly design could be created that better-matched users' needs and expectations.

In the final phase of the design process, a comprehensive usability test was conducted to evaluate the entire application thoroughly. By testing different types of user data, i.e. qualitative/subjective, qualitative/objective, quantitative/subjective, and quantitative/objective, a good understanding of both the user experience and the application's functionality was gained. Despite the tests being comprehensive in evaluating the entire application's usability, there were negative aspects to the selection of participants. With only eight test participants, there is a risk that the validity of the test was affected, especially since the test participants did not truly belong to the intended target user group. This may have affected the ability to uncover all potential user needs and challenges.

## 9.2 Result and Analysis

To address the project's research questions, we will first discuss the third question:

### **How does one construct an ontology management tool to ensure comprehension and accurate utilization by users lacking semantic expertise?**

During the process, both we and Pinteg started to understand that there was more to it than designing a simple web application. The usability testing of each phase confirmed that users expect certain things, independent of the use case of the web application. For example, we can't change what subscriptions mean for someone. Instead, we had to adapt to certain existing conceptual models and figure out how we could implement these concepts into the application.

Because the aim was to create an editor and a community for creating and sharing definitions and processes in the field of Data Privacy and Protection, the initial idea was for the tool to handle ontologies. This meant that we had to define concepts that weren't generally known, concepts such as ontology. During the evaluation parts of all the phases, we observed that even though we changed the definitions test participants didn't understand *what* the purpose of the tool was. We concluded that the tool probably would have to have another purpose, but the underlying element is an ontology editor. For the user, an ontology editor doesn't tell them anything but a tool for "the creation and management of company processes that handles sensitive information", may give the user a better context.

When we came to this conclusion, the design of the tool meant less for us than previously, and more emphasis was put on what features this application needs. The current implementation of the backend enabled us to come to a solution that satisfied that implementation, but as the analysis shows in the last phase, some concepts need to be improved.

As ontologies are collections of concepts and their relations, we believe that the definitions used in the hi-fi prototype were reasonable for this purpose, but for future iterations with the conclusion described previously, the tool should be tested with different definitions and given context. **Create Subject** could become **Create process** and **Create resource** could become **Create concept**, which could give the user a better understanding of the purpose of the tool. The ontology visualizations like the hierarchy and knowledge graph could still exist but work as complementary features.

Depending on the direction the application takes, requirements and needs may vary. The first research question was:

### **What specific demands and requirements must be addressed in the development of an ontology management tool?**

During the course of the project, specific requirements that the tool needs to meet were defined. These include the ability to create, edit, and delete ontologies, subscribe to other users' subjects in order to import and clone concepts, and have clear visualization of ontologies. However, as previously mentioned, not all requirements will be needed depending on the direction the tool takes. For example, clear visualization of ontologies is not necessary if the general public is going to use the application.

From the usability testing of the hi-fi prototype, it was interesting to see how the users interpreted the subscribe feature. As subscriptions are common among social media platforms nowadays, the testers had strong conceptual models of subscriptions. Test participants thought that it was unreasonable not to see what they were subscribing to as the subjects were supposedly public. Other services usually give the user the possibility of viewing *what* they are choosing to subscribe to, and the subscription works more as a notifier than an access token. We agreed that it is probably better to have public projects visible at all times, as instead of using the subscription as a key to opening the subject it can be used when users want to have more information from a certain subject.

Changing this will also solve other issues encountered during the evaluation of the final prototype. The import feature gives the user the possibility of copying work and adding it to their subject/s. As this is one of the core ideas of the tool, imports must be handled correctly. Some testers highlighted that when a resource is imported into their work, they expect the resource to stay even if they unsubscribe. That way, the imported resource should reference the original work, to give credit to the author.

For imports to work, there has to be a version control of resources, and a possible way of handling this is to have resources create new resources and reference their previous version. For example, if a resource is imported it will create a new resource for the importing user, called the same thing but having different id's. The new resource can have a version or reference property that points to the id of the original resource. If the new resource is finalized and also published, there will be a chain of references if someone imports the new resource. If one of the original resources is edited or deleted, it won't change the new versions, but change the references.

That brings us to another takeaway from the usability testing of the hi-fi prototype; all

testers couldn't understand why resources were locked when published. There are many reasons as to why they shouldn't be, and the best one is that the owner should be able to choose what they want to do with their resource. Also, a simple error may have to be corrected. The reason why it wasn't possible to change a published resource was that they may be referenced in other resources, and upon modifications may create errors in the backend.

Regarding the second question:

#### **What design choices are considered optimal for such a tool?**

It has been continuously discussed throughout the process. As previously mentioned, we realized that this question was not as relevant to explore at this stage as the other two. This is because the results of each usability test showed that users understood how to perform the intended tasks and found that the application had a smooth user flow and clear navigation.

Two specific design choices that proved to be important for this type of application were the colors and shapes in the graph and hierarchy. By maintaining consistency in colors and shapes for different relationships and properties, users could more easily distinguish between them.

## **9.3 Future work**

What happens next comes down to what Pinteg wants to do with the tool. If the aim is to use it as an internal ontology editor, the current implementation would work fine, and there would be no need for features like subscriptions. But, if they want to use the web application as a community for creating subjects and sharing them, the tool needs to evolve into something else using the semantic data modeling underneath the user interface. As semantic data models gives the computer better understanding and context of relations, it could be used to build powerful search engines, recommendation systems, and knowledge management from the knowledge obtained in the community. The visualisation methods, the hierarchy and the graph, might be overwhelming for users, but can be utilized internally for Pinteg employees, and/or as a feature that can be unlocked for interested users.



# Chapter 10

## Conclusion

---

Adding semantic data modeling to a service is a powerful technique when utilized to its potential, but applying it to a service requires necessary research to decide whether the service benefits. The concept of ontologies is difficult to explain to users with a limited background in semantic data modeling. Even when given a context, there is no natural conceptual connection to the concept. Given the aim to create an ontology management tool, the applications' purpose can be taken in different directions. One natural direction was to use the tool as a straightforward ontology manager, similar to existing alternatives, which requires the user to be familiar with concepts. Another direction could be to make the tool similar to other community-driven platforms, such as GitHub and StackOverflow<sup>1</sup>, and add more social media-like features. In this case, it could be beneficial to use subscriptions. In either case, version control would be necessary to handle different resource versions deriving from the same concept, and to keep track of history. This also enables resources to be changed even though they have been shared to the community for others to use in their work.

---

<sup>1</sup><https://stackoverflow.com/>

# References

---

- [1] Panos Alexopoulos. *Semantic modeling for data*. O'Reilly Media, 2020.
- [2] Pinteg. About us. <https://pinteg.eu/about-us/>, s.d.
- [3] United Nations. Sustainable development goals, 2024. 29-05-2024.
- [4] Lund University. Lubsearch. <https://emedia.lub.lu.se/db/search>.
- [5] Emhimed Alatrish. Comparison of ontology editors. *ERAF Journal on Computing*, 4:23–38, 01 2012.
- [6] RG et al. Côté. The ontology lookup service, a lightweight cross-platform tool for controlled vocabulary queries. *BMC Bioinformatics*, 2006.
- [7] Stack Path. What is a web application? <https://www.stackpath.com/edge-academy/what-is-a-web-application/>. [Accessed: April 4, 2024].
- [8] Sun Microsystem. Distributed application architecture. <https://web.archive.org/web/20110406121920/http://java.sun.com/developer/Books/jdbc/ch07.pdf>, 2011. [Accessed: April 4, 2024].
- [9] Henrik Nielsen, Jeffrey Mogul, Larry M Masinter, Roy T. Fielding, Jim Gettys, Paul J. Leach, and Tim Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, 1999.
- [10] Tim Bray. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259, 2017.
- [11] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaktionsdesign: bortom människa-dator-interaktion*. Studentlitteratur, 2016.
- [12] Donald A. Norman. *The Design of Everyday Things*. Basic Books, [New York], 2013.
- [13] John D Gould and Clayton Lewis. Research contributions: Human aspects of computing designing for usability. *Communications of the ACM*, 28(3):300–311, 1985.

- [14] Jakob Nielsen. *Usability Inspection Methods*. John Wiley Sons, Inc., 1994.
- [15] Nielsen Norman Group. *Thinking aloud: The 1 usability tool*, 2019.
- [16] John Brooke. *Sus - a quick and dirty usability scale*. *Redhatch Consulting Ltd*, 1996.
- [17] Alana Chinn. What's the system usability scale (sus) how can you use it? <https://blog.hubspot.com/service/system-usability-scale-sus>, 2022.
- [18] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley, Hoboken, NJ, 2008.
- [19] Kp J Sharma S. Security analysis of oauth 2.0 implementation. *2023 Innovations in Power and Advanced Computing Technologies (i-PACT), Innovations in Power and Advanced Computing Technologies (i-PACT), 2023*, 2023.
- [20] X. et al. Fang. Effects of different interface color modes and textbox design on users' reading efficiency and accuracy. In *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 0102–0106, 2022.
- [21] Maja Nowak. *Vue vs. react in 2023 - comparison of two most popular js frameworks*. <https://www.monterail.com/blog/vue-vs-react>, s.d.
- [22] Joshua D. et al. Scarsbrook. Typescript's evolution: An analysis of feature adoption over time. In *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*, pages 109–114, 2023.

# Appendices

# Appendix A

## Personas

---

### **Sophia the Information Seeker**

Sophia is working within an area that handles complex information. She is studying on the side to further climb the company hierarchy. She seeks information in the area for an essay, and is browsing the internet for the essay background.

#### **Needs and Goals**

- Sophia needs a tool that facilitates the exploration and discovery of relevant information within her domain.
- She seeks a user-friendly interface that allows her to quickly search and navigate ontologies to find specific concepts or relationships.
- Sophia wants the tool to provide visual representations of data relationships, aiding her in understanding complex information structures.

#### **Pain Points**

- Sophia finds it challenging to navigate traditional databases and search engines to locate specific information buried within large datasets.
- Lack of visualization tools makes it difficult for Sophia to grasp the connections between different data entities and their attributes.

#### **Behavior**

- Sophia values simplicity and ease of use in software tools, preferring intuitive interfaces that require minimal training.

- 
- She relies heavily on search functionality to locate information quickly, so the tool should offer robust search capabilities with filters and advanced querying options.
  - Sophia appreciates visual aids such as graphs, charts, and diagrams that help her visualize data relationships and identify patterns more effectively.

## **Emma the Domain Expert**

Emma works at a mid-sized company that deals with customer data regularly. She has a background in legal compliance and understands the basics of data protection laws like GDPR.

### **Needs and Goals**

- Emma needs a tool that simplifies the process of managing ontologies within knowledge graphs.
- She wants the tool to be user-friendly and intuitive since she doesn't have extensive knowledge of semantic modeling.
- Emma aims to ensure her company's compliance with data privacy regulations while efficiently managing personal data.

### **Pain Points**

- Lack of familiarity with semantic modeling concepts makes it difficult for Emma to contribute effectively to ontology management.
- Current tools are either too complex or too rudimentary for her needs, resulting in inefficiencies in data management.

### **Behavior**

- Emma is open to learning new tools but prefers solutions that streamline her workflow and don't require extensive training.
- She appreciates clear documentation and user support to guide her through the ontology management process.

## **Oscar the Project Manager and Backend Lead Developer**

Oscar is a project manager at Pinteg, responsible for overseeing the development of the ontology management tool. Additionally, he serves as the lead developer for the backend aspects of the tool. He has a background in software development, particularly in backend systems, and extensive experience in project management.

---

## Needs and Goals

- Oscar needs a tool that aligns with the project objectives and meets the requirements outlined by stakeholders, including domain experts and clients.
- As the backend lead developer, he aims to ensure that the technical aspects of the tool are robust, scalable, and efficient.
- Oscar understands the importance of user-friendliness and seeks a solution that can be easily understood and utilized by end-users with limited knowledge of semantic modeling.

## Pain Points

- Balancing the technical requirements of the backend development with the usability needs of end-users can be challenging.

## Behavior

- Oscar is focused on delivering a high-quality product that meets both technical and user needs.
- He collaborates closely with frontend developers and user experience designers to ensure that the tool's interface is intuitive and user-friendly.
- Oscar leverages his technical expertise to address backend development challenges efficiently while keeping the end-user experience in mind.
- He actively seeks feedback from end-users to iterate on the tool's design and functionality, ensuring continuous improvement.

# Appendix B

## Terms used in Usability Testing, Phase 4

---

### **Ontology**

A structured description of concepts, relationships, and properties within a subject area. For example, we can create an ontology for Pets.

### **Semantic model**

A systematic representation of the meaning of information, facilitating understanding and interoperability between systems.

### **Resource**

A unit within the ontology and/or a semantic model. In this case, it can be an instance or a concept.

### **Concept**

A general class of entities within the ontology. For example, Dog in our Pets ontology.

### **Properties**

Attributes or characteristics that describe resources or concepts within the ontology. For example, *Name*, *Age*, and *Breed*, for the concept of Dog.

### **Instance**

A specific occurrence of a resource. For example, *Sally, 2 years old, German Shepherd*.



