



University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

BACHELOR'S THESIS

Study programme/specialisation:

Bachelor in Computer Science

Spring/Autumn semester, 2021

Open/Confidential

Author: Rostem, Alan Shakhewan & Babic, Nemanja

Programme coordinator: Erlend Tøssebro

Supervisor(s): Ahmed Adnan Aqrawi

Title of bachelor's thesis:

Digital e-learning platform for oil and gas sector in conjunction with The Open Group OSDU Forum

Credits: 20

Keywords:

user-centered design, e-learning platform, architectural design, minimum viable product, usability testing, prototyping, programming, front-end, back-end

Number of pages: 95.....

+ supplemental material/other: 14.....

Stavanger, 23. April 2021.
date/year

Digital e-learning platform for oil and gas sector in conjunction with The Open Group OSDU Forum

Alan Rostem and Nemanja Babić

Supervisors: Ahmed Aqrawi and Erlend Tøssebro

Bachelor thesis, Computer Science

University of Stavanger

Abstract

Several studies are targeting improving user interface and overall user experience. Various design principles are used during the development process and help shape the product in a wanted manner. One of the most popular design principles is user-centered design. The user is in the middle of the iterative development process. In this way, the users' experience gets significantly better, and user-centered software can improve their productivity.

In this thesis, we have used user-centered design as the foundation stone for an e-learning application. The target group was people working in the oil and gas industry. They were also the participants in our usability testing sessions. The first prototype made was a low-fidelity one, based on the researched design principles and examples of modern user interfaces with the same purpose. The low-fidelity paper prototype has helped us quickly realize what can be done differently and perhaps more effectively. We used the collected user feedback from the testing to create a functional high-fidelity prototype using popular solutions. It is the final prototype of this thesis and a more natural platform for usability testing.

The changes made from the first prototype and the features represented were very positive, with few exceptions and suggestions. That can be confirmed by looking at the quantitative analysis. Users have solved all the usability testing tasks successfully, without significant issues. They have rated the difficulty of the tasks, using a scale from one to five, with an average score of 1,4 in the first usability testing session and 1,31 in the second. The overall experience has been excellent, and the developers can use the prototype to build the e-learning platform further to accomplish more and go above the minimum viable product standards that we used.

Acknowledgements

This bachelor thesis results from a project assigned by the Faculty of Science and Technology at the University of Stavanger. It is written in a collaboration with Schlumberger and the production is meant for The Open Group OSDU Forum™.

We want to thank our supervisors Ahmed Adnan Aqrabi and Erlend Tøssebro, for remarkable guidance and valuable feedback throughout the entire thesis. Their understanding and insight into the field have helped us grow professionally and inspire future work. We would also like to express our gratitude towards Schlumberger, Carnain AS and the employees in the respective firms for taking their valuable time to help us perform usability testing sessions.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Goals and objectives | 2 |
| 1.2 | Contributions | 2 |
| 1.3 | Thesis outline | 3 |
| 2 | Background | 4 |
| 2.1 | Schlumberger | 4 |
| 2.2 | The Open Group OSDU™ Forum | 4 |
| 2.3 | E-Learning | 4 |
| 2.4 | Similar products | 5 |
| 2.4.1 | Udemy | 5 |
| 2.4.2 | LinkedIn Learning | 5 |
| 2.4.3 | Moodle | 6 |
| 2.4.4 | What OSDU Academy will do different | 6 |
| 2.5 | Business model | 7 |
| 2.6 | Software tools background | 7 |
| 2.6.1 | C# and .NET | 7 |
| 2.6.2 | ReactJS | 7 |
| 2.6.3 | JetBrains Rider | 8 |
| 2.6.4 | MongoDB Compass | 8 |
| 2.6.5 | Postman | 8 |
| 2.6.6 | Google Chrome | 9 |
| 2.6.7 | Microsoft PowerPoint | 9 |
| 2.6.8 | Ngrok | 9 |
| 3 | User-centered design principles | 10 |
| 3.1 | User-centered design | 10 |
| 3.1.1 | Process | 10 |
| 3.1.2 | The essential elements of UCD | 11 |
| 3.1.3 | User requirements | 11 |
| 3.1.4 | Prototype | 12 |
| 3.1.5 | Evaluation | 13 |
| 3.2 | Layout patterns | 13 |
| 3.2.1 | Gutenberg diagram | 14 |
| 3.3 | Emotional design | 15 |
| 3.3.1 | Cognitive aspects | 16 |
| 3.3.2 | Emotional states and cognition | 17 |
| 3.4 | Interaction design and user experience | 17 |
| 3.4.1 | Usability goals | 17 |
| 3.4.2 | Human-computer interaction | 18 |
| 4 | Implementation and testing methodology | 19 |
| 4.1 | Project planning | 19 |
| 4.2 | Defining personas | 20 |
| 4.2.1 | Persona perspective evaluation | 20 |
| 4.2.2 | Persona definition process | 20 |

| | | |
|----------|--|-----------|
| 4.2.3 | Persona categories | 20 |
| 4.3 | User stories | 22 |
| 4.3.1 | Jens' story | 22 |
| 4.4 | Requirements | 23 |
| 4.4.1 | The requirements change | 24 |
| 4.4.2 | Software requirements specification | 25 |
| 4.5 | Architectural design | 25 |
| 4.5.1 | Technical requirements | 26 |
| 4.5.2 | Choice of database management system | 26 |
| 4.5.3 | Front-end programming utilities | 27 |
| 4.5.3.1 | ReactJS | 28 |
| 4.5.3.2 | Flutter | 28 |
| 4.5.3.3 | Web page template libraries | 29 |
| 4.5.3.4 | The final choice | 29 |
| 4.5.4 | The back-end solution | 29 |
| 4.5.4.1 | ASP.NET Core | 29 |
| 4.5.4.2 | Flask | 30 |
| 4.5.4.3 | Spring Boot | 30 |
| 4.5.4.4 | The final choice | 30 |
| 4.5.5 | Design patterns | 31 |
| 4.5.5.1 | Service layer | 31 |
| 4.5.5.2 | REST | 31 |
| 4.5.5.3 | SPA | 32 |
| 4.5.5.4 | MVC | 32 |
| 4.5.5.5 | Middleware | 33 |
| 4.6 | Usability testing | 33 |
| 4.6.1 | Testing goals | 33 |
| 4.6.2 | Test setup | 34 |
| 4.7 | Prototype 1 testing | 34 |
| 4.7.1 | Demographic group | 34 |
| 4.7.2 | Testing environment | 35 |
| 4.7.3 | Scenarios and tasks | 35 |
| 4.7.4 | Testing platform | 36 |
| 4.7.5 | The method | 36 |
| 4.7.6 | The questionnaire | 37 |
| 4.8 | Prototype 2 testing | 37 |
| 4.8.1 | Demographic group | 37 |
| 4.8.2 | Testing environment | 38 |
| 4.8.3 | Scenarios and tasks | 38 |
| 4.8.4 | Testing platform | 38 |
| 4.8.5 | The method | 38 |
| 4.8.6 | The questionnaire | 39 |
| 5 | The Prototypes | 40 |
| 5.1 | Prototype 1 | 40 |
| 5.1.1 | The home page | 40 |
| 5.1.2 | Login and registration | 42 |
| 5.1.3 | Finding courses | 44 |

| | | |
|----------|---|-----------|
| 5.1.4 | Applying for a course | 45 |
| 5.1.5 | Course design | 47 |
| 5.1.6 | Certification | 49 |
| 5.2 | Prototype 2 | 51 |
| 5.2.1 | Home page | 51 |
| 5.2.2 | Profile page | 53 |
| 5.2.3 | Course front page | 54 |
| 5.2.4 | Course interface | 55 |
| 5.2.5 | Teach OSDU | 56 |
| 5.2.6 | Pre-certification page | 57 |
| 5.2.7 | Functional architecture | 58 |
| 5.2.7.1 | Services and components | 58 |
| 5.2.7.2 | Controllers and access routes | 60 |
| 5.2.7.3 | Sequence diagram | 61 |
| 5.2.8 | Database design | 62 |
| 5.2.8.1 | Course content management | 63 |
| 5.2.9 | Security baseline | 64 |
| 5.2.9.1 | User authentication and permissions | 64 |
| 5.2.9.2 | CSRF-secured yet vulnerable to XSS | 66 |
| 5.2.9.3 | Password encryption | 67 |
| 5.2.9.4 | SSL Certificate | 68 |
| 5.2.9.5 | Injection mitigation by design | 69 |
| 5.2.9.6 | Remaining vulnerabilities | 69 |
| 5.3 | Product limitations | 70 |
| 5.3.1 | Course editor | 70 |
| 5.3.2 | Registration methods and role selection | 70 |
| 5.3.3 | Ratings and forums | 71 |
| 5.3.4 | Search engine and course recommendations | 71 |
| 6 | Results and analysis | 73 |
| 6.1 | Prototype 1 | 73 |
| 6.1.1 | Scenarios and tasks | 73 |
| 6.1.1.1 | Scenario I: Registration | 73 |
| 6.1.1.2 | Scenario II: Finding a relevant course | 74 |
| 6.1.1.3 | Scenario III: Login and re-using the platform | 74 |
| 6.1.1.4 | Scenario IV: Continue with previous work | 74 |
| 6.1.2 | Quantitative findings | 75 |
| 6.1.3 | Qualitative findings | 78 |
| 6.1.4 | Additional participant remarks | 81 |
| 6.1.4.1 | Common behavior | 81 |
| 6.1.5 | Further insight | 81 |
| 6.2 | Prototype 2 | 82 |
| 6.2.1 | Scenarios and tasks | 82 |
| 6.2.1.1 | Scenario I: Revisiting a course | 83 |
| 6.2.1.2 | Scenario II: Apply for a petroleum geology course | 83 |
| 6.2.1.3 | Scenario III: Your certificate status | 83 |
| 6.2.1.4 | Scenario IV: Taking the petroleum geology course | 84 |
| 6.2.2 | Quantitative findings | 84 |

| | | |
|----------|--|-----------|
| 6.2.3 | Qualitative findings | 87 |
| 6.2.4 | Changes and improvements | 91 |
| 7 | Conclusion and future work | 92 |
| 7.1 | Conclusion | 92 |
| 7.1.1 | Design | 92 |
| 7.1.2 | MVP | 93 |
| 7.1.3 | Final conclusion | 93 |
| 7.2 | Future work | 94 |
| 7.2.1 | Search functionality | 94 |
| 7.2.2 | Supporting course instructors | 94 |
| 7.2.3 | Federated login | 94 |
| | References | 96 |
| | Appendix | 98 |
| A1 | Software Requirements Specification | 98 |
| A1.1 | Introduction | 98 |
| A1.2 | Functional requirements | 98 |
| A1.2.1 | Home page | 98 |
| A1.2.2 | Log-in | 99 |
| A1.2.3 | User profile | 99 |
| A1.2.4 | Certificating | 99 |
| A1.2.5 | Use of different media | 99 |
| A1.2.6 | Registration | 100 |
| A1.2.7 | Course management | 100 |
| A1.2.8 | Feedback and reports | 101 |
| A1.2.9 | User Roles | 101 |
| A1.2.10 | Forum | 101 |
| A1.3 | Non-functional requirements | 102 |
| A1.3.1 | Scalability | 102 |
| A1.3.2 | Security standards | 102 |
| A1.3.3 | Responsiveness (Performance) | 102 |
| A1.3.4 | Down-time (Serviceability) | 103 |
| A1.3.5 | Appealing design | 103 |
| A1.3.6 | Support access to external resources | 103 |
| A1.3.7 | Alternative log-in methods | 104 |
| A2 | Persona - Sofia | 105 |
| A2.1 | User profile | 105 |
| A2.2 | User story Sofia | 106 |
| A3 | Prototype 1 - testing session | 107 |
| A3.1 | Scenarios and Tasks | 107 |
| A3.1.1 | Scenario/task I: Registration | 107 |
| A3.1.2 | Scenario/task II: Finding a relevant course | 107 |
| A3.1.3 | Scenario/task III: Login and re-using the platform | 107 |
| A3.1.4 | Scenario/task IV: Continue with previous work | 107 |
| A3.2 | Questionnaires and surveys | 107 |
| A3.2.1 | Pre-testing survey | 107 |
| A3.2.2 | Post-task questions | 108 |

| | | | |
|----|--------|--|-----|
| | A3.2.3 | Post-testing questions | 108 |
| A4 | | Prototype 2 - testing session | 108 |
| | A4.1 | Scenarios and Tasks | 108 |
| | | A4.1.1 Scenario I: Revisiting a course | 108 |
| | | A4.1.2 Scenario II: Apply for a petroleum geology course | 109 |
| | | A4.1.3 Scenario III: Your certificate status | 109 |
| | | A4.1.4 Scenario IV: Taking the petroleum geology course | 109 |
| | A4.2 | Questionnaires and surveys | 109 |
| | | A4.2.1 Pre-testing survey | 109 |
| | | A4.2.2 Post-task questions | 110 |
| | | A4.2.3 Post-testing questions | 110 |
| A5 | | Prototype Testing Session Videos | 110 |
| A6 | | Source code | 111 |
| A7 | | Demo Video | 111 |

List of Figures

| | | |
|------|--|-----|
| 3.1 | A user-centered design process illustration. Inspired by User Centered Design [1] | 11 |
| 3.2 | An example of a flip-flop paper prototype. Source: [2] | 13 |
| 3.3 | The Gutenberg diagram [3] | 15 |
| 3.4 | Donald A. Norman's levels of emotional design | 16 |
| 3.5 | HCI - multidisciplinary field of study. Inspired by: Human-Computer interaction [4] | 18 |
| 4.1 | A Gantt diagram of the project plan | 19 |
| 4.2 | An example of a persona named Jens | 21 |
| 4.3 | Jens' user-story | 23 |
| 4.4 | The rating table | 27 |
| 5.1 | Overview of upper part of the home page | 41 |
| 5.2 | Overview of lower part of the home page - not logged in | 41 |
| 5.3 | Overview of lower part of the home page - logged in | 42 |
| 5.4 | The login page after pressing "sign in" | 43 |
| 5.5 | The registration page after pressing "sign up" | 44 |
| 5.6 | Searching for a course | 45 |
| 5.7 | Course page when applying for the first time | 46 |
| 5.8 | Course page when already applied for | 46 |
| 5.9 | The course interface showing a document | 47 |
| 5.10 | The course interface showing a playable video | 48 |
| 5.11 | Further down the course page | 49 |
| 5.12 | The test interface, part 1 | 50 |
| 5.13 | The test interface, part 2 | 50 |
| 5.14 | Certificate | 51 |
| 5.15 | Home page when the user is not logged in | 52 |
| 5.16 | Home page when the user is logged in | 53 |
| 5.17 | Profile page of a user named Jens | 54 |
| 5.18 | Course front page | 55 |
| 5.19 | Course interface | 56 |
| 5.20 | Teach OSDU | 57 |
| 5.21 | Certification choice | 58 |
| 5.22 | Sequence diagram of the login process | 61 |
| 5.23 | Diagram of MongoDB database | 62 |
| 5.24 | Example of a decoded token using jwt.io | 66 |
| 5.25 | Example of a user's hash and salt stored in the database (screenshot from MongoDB Compass) | 67 |
| 5.26 | Secure connection as seen using Google Chrome | 68 |
| 6.1 | An overview of time usage | 75 |
| 6.2 | Time median for every task | 76 |
| 6.3 | Rating overview | 77 |
| 6.4 | An overview of time usage for the second usability test | 85 |
| 6.5 | Time median for every task in the second usability test | 85 |
| 6.6 | Rating overview for the second usability test | 86 |
| A2.1 | User named Sofia | 105 |
| A2.2 | User story - Sofia | 106 |

Listings

| | | |
|---|--|----|
| 1 | React component example | 58 |
| 2 | Example of a service layer object in JavaScript (ECMAScript 6) | 59 |
| 3 | C# course controller code example | 60 |
| 4 | Course content example in XML format | 63 |
| 5 | C# authorization class attribute attached to controller | 64 |

1 Introduction

Employees among many petroleum businesses are required to learn new skills because of changing technologies in new assignments. Unfortunately, the task of learning and improving one's skill is time-consuming and not always available to everyone. At the time of this writing, a pandemic has struck the world, which has forced employees to be laid off or work from home. It is clear that the increasing demand for e-learning platforms results from various changes people have to adapt to. Hence, the opportunity of providing an e-learning platform to the oil and gas sector has become present in a partnership with Schlumberger.

Developing a wide platform for teaching and learning requires the developers to understand a range of topics that include studies on future consumers in order to conclude a suitable mission. The user interface of software products have changed rapidly, and standards have evolved into designing simplistic and comprehensive looks on the interface layout. Nevertheless, every application or service seen in the industry is unique and fulfills different purposes while maintaining a modern standard. How does one know for certain that an interface fits acceptable criteria without becoming a carbon-copy?

Modern stylistic approaches are very minimalistic. They are structured by rectangular shapes filled with a color from a limited palette. Elements like gradients and exotic shapes are not often present in such interfaces anymore. Additionally, text has become a lot less present, yet it is the primary source of information to the user. Only single words are used to describe functional elements and sentences or paragraphs are suited in informational areas.

As much as the interface design and visuals are important, the functional aspects and quality of the product cannot be explained from this type of design alone. As developers, it is also crucial to design the product's technological architecture as well. Things like response time, resource- and data management and security are all topics this thesis will focus on in the process of creating the platform. These aspects are essential to understand for improving the quality of any software product, yet there are many more steps involved.

Nothing is perfect and this project is no exception. Testing is called for in any software project and is never enough. This project will need to adopt iterative software development and usability testing methodologies in order to reach the best possible quality. Hence, the development of OSDU Academy will start with reasonable knowledge in mind.

1.1 Goals and objectives

The thesis' main goal is to learn and understand the various user-centered design principles to develop a prototype for an e-learning platform for the oil and gas sector. The consumers of the platform are people who want to expand their knowledge, along with those creating the learning content. An easy-to-use interface adjusted to multiple user profiles is a key objective. Furthermore, an important component of the design process is the product architecture. This implies the technical design aspects and the workflows in code, which will be achieved through various design patterns used in the software industry. Possible future users will test the interface to confirm the product's quality and ensure that current standards are followed.

1.2 Contributions

The contributions of this thesis consists of:

- An interface design for an e-learning platform sided in the oil and gas sector based on researched user-centered design principles
- Architectural design of a functional, extensible and open-source e-learning platform prototype
- Defining a minimum viable product to provide users in the respective industries to evaluate
- Delivering the minimum viable product to real life working scenarios for testing and validation of quality and usability metrics
- Documentation of data and analysis of e-learning prototype usability tests

1.3 Thesis outline

This bachelor thesis will have following chapters

Chapter 2 General description and background material

Chapter 3 An overview over different design principles, as well as key concepts for understanding the thesis

Chapter 4 Design and implementation process documented with respect to additional relevant knowledge and various testing methodologies post-development. Descriptions on usability testing, testing sessions and testing environment

Chapter 5 An overview of the prototypes created and used in the testing process. Documented architectural design and key aspects in the functional prototype code

Chapter 6 The results and analysis from the prototype testing. The section should evaluate the results for future improvements

Chapter 7 Conclusion and future work

2 Background

The following chapter describes the several entities present in the project's working and a background. The project's name is OSDU Academy, and it is a collaborative project meant for a specific demographic within The Open Group OSDU™ Forum. Additionally, the resources that helped this platform's development originate from a company named Schlumberger, with help from our supervisor.

2.1 Schlumberger

Schlumberger is an oilfield service company [5]. The organization is operating in more than 120 countries worldwide [6]. One of them is Norway, where they offer their expertise in sales, services, exploration, and production, to name a few. They also offer software solutions in the field, which this project will contribute to.

2.2 The Open Group OSDU™ Forum

The Open Subsurface Data Universe™ Forum is a trademark of The Open Group which aims to develop an open, standards-based data platform that will bring together exploration, development, and wells data [7]. The OSDU Forum is an international forum of oil and gas operators, cloud services companies, technology providers, suppliers of applications to oil and gas operators, academia, and other standards organizations [7].

The project's relation to the organization is that they are the target audience of the platform that will be developed. The e-learning application will be designed for oil and gas professionals in The Open Group OSDU Forum in an open-source manner.

2.3 E-Learning

E-learning, or electronic learning, is the form of studying using digital resources. Due to its virtual nature, this method of education is widespread and available everywhere. There are various means for teaching, such as videos and embedded e-books. Nevertheless, the learning methods within such platforms are not without drawbacks [8].

E-learning is a cost-effective way of learning and usually has no time restrictions. There is no need to wait for other people to start the class or even attend it at a specific place. This learning method improves productivity, and the multimedia approach can be quite engaging for the learner [8]. Nevertheless, this technique is health-concerning. Eyestrain, bad posture, and other physical problems may affect the learner after extensive use [8]. Furthermore, it makes the student less practically skilled, and both the teacher and students can feel less engaged since there is no natural personal connection between the two parts. This is fairly visible in today's pandemic with almost every school and university lecturing online. E-learning is practical and useful for individuals with a tight schedule, but its benefits come at a price that does not suit everyone.

2.4 Similar products

OSDU Academy will not be the first in the market. Despite the variants of e-learning platforms having the same type of clients, other platforms focus on providing content in different manners that fit different market requirements.

2.4.1 Udemy

The massive open online course provider, *Udemy*, is a type of platform that fits the criteria of an e-learning application with its own market of users across every field around the globe. The platform is designed for instructors to build courses around topics that interest them so that anyone who purchases them can study them and get a learning certificate. The platform has its own course development tools for the instructors, featuring a variety of media uploading and more.

Udemy is very similar to OSDU Academy as the platform will feature quick ways to apply to a course. The business model is yet to be decided so it is not certain that OSDU Academy will feature purchasing courses the same way Udemy does, but the platform will look similar in the general structure.

2.4.2 LinkedIn Learning

LinkedIn Learning, a subsidiary of *LinkedIn*, is an online video-learning website where experts in the industry host videos about business, creative and technology. The courses

work similar to *Udemy*. They can be purchased, offer a variety of learning mediums and allow for earning a learning certificate. What makes this platform unique is the fact that it is connected to LinkedIn, which is a platform that connects professionals together. LinkedIn Learning pairs well with it since other businesses can see the user's learning certificates on their profiles. This is beneficial for the user's reputation in the professional world.

Compared to Udemy and OSDU Academy, LinkedIn Learning is quite similar in structure too. The only exception is that the target audience is different, which is true among all three platforms.

2.4.3 Moodle

Unlike the formerly mentioned products, *Moodle* is considered to be a learning management system rather than a standalone platform. It is designed to provide educators, administrators, and learners with a fast and secure system for personalized learning environments [9]. This system is distributed under the *GNU General Public License* [10], which means anyone can use it free of charge. Many schools and universities use Moodle to host their courses and enable a robust platform for their students.

While Moodle does not provide the same *system* as OSDU Academy, there is one key similarity: open-source. Just like Moodle, OSDU Academy will be created on the basis of keeping the structure as open-sourced as possible.

2.4.4 What OSDU Academy will do different

The products listed above cater to learning in general and have quite similar appeal as OSDU Academy will have. The platform will feature courses that anyone can apply to and a means for teachers, or instructors, to easily create a course.

With a currently limited business model, OSDU Academy will only do a few things differently than the aforementioned products. Firstly, it is targeted towards The Open Group OSDU Forum™ closely tied to the oil and gas sector. Lastly, it is the only open-source *e-learning platform* out of the three. Even though Moodle is open-source, it is not an e-learning platform in itself. These are crucial elements that distinguish OSDU Academy from the other three aforementioned products and will be the selling-point of

the platform. This means that OSDU Academy has the potential to be used internally among many other businesses with its open and extensible nature.

2.5 Business model

The business model for this project is not fully decided. It can be defined in later stages by Schlumberger. Only that the project is an open-source e-learning platform has been taken into consideration. Throughout this thesis there will be many decisions made based on this fact alone.

2.6 Software tools background

The following chapter will explain what software tools were used in the making of OSDU Academy. As this is a software project, popular software development and analysis tools were used in the process, such as integrated development environments. It is crucial to note that the following tools were chosen for use with thorough research at the early stages of the development process and it is all documented in chapter 4.5.

2.6.1 C# and .NET

C# and the .NET Core library were used for a variety of reasons discussed in chapter 4.5.4.4. C# is an object-oriented and compiled programming language developed by Microsoft. It is approved by European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). Anders Hejlsberg and his team has developed C# during the development of the .NET Framework. It is one of the most popular and widely used programming languages with a strong resemblance to Java.

2.6.2 ReactJS

As explained in chapter 4.5.3, ReactJS became the front-end web development utility of choice. It is an open-source library for JavaScript that adds a declarative and component-based application programming interface to the front-end development environment. ReactJS is also based on a subsidiary language of JavaScript, called JSX, which lets developers embed HTML-like code into JavaScript code to create interfaces. The purpose

of ReactJS is to create single-page applications and it features a wide variety of libraries created by other developers in its community.

2.6.3 JetBrains Rider

Due to using C# and ReactJS in the project, Rider has been a natural choice to go for. It is a fast and powerful, cross-platform .NET Integrated Development Environment [11]. It is based on the IntelliJ platform and ReSharper. It supports multiple .NET project types, including .NET Core and Mono based projects. It allows for developing both desktop and web applications, in addition to Unity games and Xamarin mobile apps. Additionally, the wide range of programming languages the environment supports for syntax highlighting and code-completion is a largely attractive trait in this utility.

2.6.4 MongoDB Compass

MongoDB Compass is a graphical user interface for managing databases created in MongoDB. It is possible to make queries, insert, delete and update data for any authorized database using the comprehensible interface. The software also provides debugging and optimization tools for such databases.

The choice of using MongoDB Compass results from selecting MongoDB as the Database Management System for OSDU Academy. Read more about it in chapter 4.5.2.

2.6.5 Postman

Postman is a tool used to test, share and create web applications. The most common usage is to make all types of Hyper Text Transfer Protocol, or HTTP, request calls to a web application for testing purposes. This tool provides users with a variety of usability tools to save previously used calls for later use, among other things.

Since OSDU Academy is an e-learning platform, it should be developed using a client-server model, as briefly mentioned in chapter 4.5.1. As a result, HTTP requests will become a common theme as part of the architecture and thus requires a proper testing tool for such functionalities. Therefore, Postman is our software tool of choice for the job.

2.6.6 Google Chrome

The popular web browser developed by Google Inc., Google Chrome, resulted in being a testing and development environment for the front-end website portion of OSDU Academy. Aside from being a feature-rich and extensible web browser, it also provides a variety of development tools for front-end web developers. There exists an inspector for viewing a website's source code, a developer console with real-time JavaScript interpretation, a network profiler and a performance monitor, among many other things.

2.6.7 Microsoft PowerPoint

Microsoft PowerPoint is a presentation creation software tool developed by Microsoft as part of the Microsoft Office family. It is featured as a desktop application and a web application with less features. The program allows the user to create informative, aesthetically pleasing and interactive slides for their presentations. The varying amount of features opens for many other use cases too, such as web design.

Paper prototyping can be an exhausting method of representing an ideal solution. Using PowerPoint, this process can be simplified and more effectively. How and why PowerPoint was utilized in this thesis can be found in chapter 4.7.4.

2.6.8 Ngrok

Ngrok allows developers to tunnel their local development server to the public with ease. This software saves developers time and money from setting up a domain only to test their experimental servers temporarily. The program also includes features for monitoring each request performed by both clients and the main server. Ngrok is free to use by anyone, but provides paid services that include premium features, such as improved tunneling capacity.

As part of the prototyping process, Ngrok became a useful tool that enabled the usability testing sessions for the dynamic prototype, as mentioned in chapter 4.8.4. The HTTP request monitoring and ability to create temporary links for the testing participants was very easy to do with Ngrok.

3 User-centered design principles

"Design is a plan or drawing produced to show the look and function or workings of a building, garment, or other object before it is made." [12]

In software engineering, the design is a process of converting requirements into a suitable idea that gives a programmer a blueprint for coding and implementation. It is not always simple to notice the difference between a well- and a poorly-designed product. Thankfully design studies have become more popular in the modern-day, and the result of continuous work arose in the form of various design principles and rules. By following these, one can end up with a high-quality result. One of the most important objectives in software development is to be user-friendly. Henceforth, it is important to have an understanding of the average user and their thinking, as it allows the designer to adapt the product to user requirements and their skills.

3.1 User-centered design

User-centered design (UCD) is an iterative design process in which the users and their needs are the main focus in every step of the development [1].

The significance of applying user-centered design to a software project is to include future users in the forming of the product through different research methods, such as surveys and interviews. A close relationship between the designer, the software engineer and the end-user makes the product more usable and closer to the user expectations and technical requirements.

3.1.1 Process

The aim of the project should be the whole user experience. To make the final product as close as possible to users' expectations, the product use cases must be clear at the very beginning of the process, even before the requirements are specified.

The next step would be designing the initial solution in the form of an early draft. As UCD is an evaluation-driven design process, it is important to include targeted user groups to give feedback on the draft. This way, the design team gets an insight into what could

and should be changed relative to the users' demands and technical requirements. Keep in mind, UCD is an iterative process, which means that the final steps repeat until the product is on a satisfying level. The whole process is illustrated in figure 3.1.

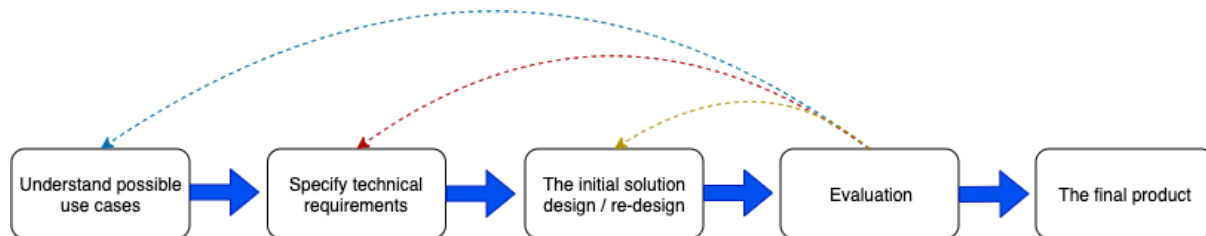


Figure 3.1: A user-centered design process illustration. Inspired by User Centered Design [1]

3.1.2 The essential elements of UCD

As it is stated in *User-Centered design: An introduction* [13] by *Ekatarina Novoseltseva* there are four essential elements of UCD:

- **Visibility:** It should be clear from the beginning what the product is about and in what ways can it be used.
- **Accessibility:** Information-access processes have to be as simple as possible. The design-team should implement multiple ways to make way for what a user is looking for, i.e. a menu, search box, action button, etc.
- **Legibility:** Users should be able to recognize, distinguish and understand various symbols and words without any difficulty.
- **Language:** Keep phrases and words simple and easy to understand.

3.1.3 User requirements

“User requirements, often referred to as user needs, describe what the user does with the system, such as what activities that users must be able to perform.” [14]

It is important to gather information and set requirements for the final product. The requirements are established based on, among other things, users' skills, understanding, goals, and the environment the product will be used in. As the user-centered design is an iterative process, the established requirements could easily be changed in the later phases

of testing and evaluation. The point is to establish requirements as a foundation of the system and make means for users to achieve their goals.

Establishing the requirements starts with creating a layout for the possible use cases. Popular solutions are stories that show how a user might use the system and how it should be used, in addition to the use case-driven approach¹. The additional data can be collected in the later development stages by using a prototype as a center of discussion. After there is enough information available to analyze and acknowledge the data, one can establish the requirements and move on to the next phase of designing.

3.1.4 Prototype

“A prototype is the first example of something, such as machine or other industrial product, from which all later forms are developed.” [15]

Prototype development helps the designer answer questions about the requirements or user-experience, among other things. Consequently, it is a crucial part of the design process as it impacts greatly how the final product is going to look and function [16]. A prototype can vary from a low-fidelity to a high-fidelity prototype. The former one is a less expensive version that focuses solely on the look, and the latter is more interactive and concentrates on the live usability. A paper prototype example is shown in figure 3.2.

¹Steps user might take to accomplish a goal

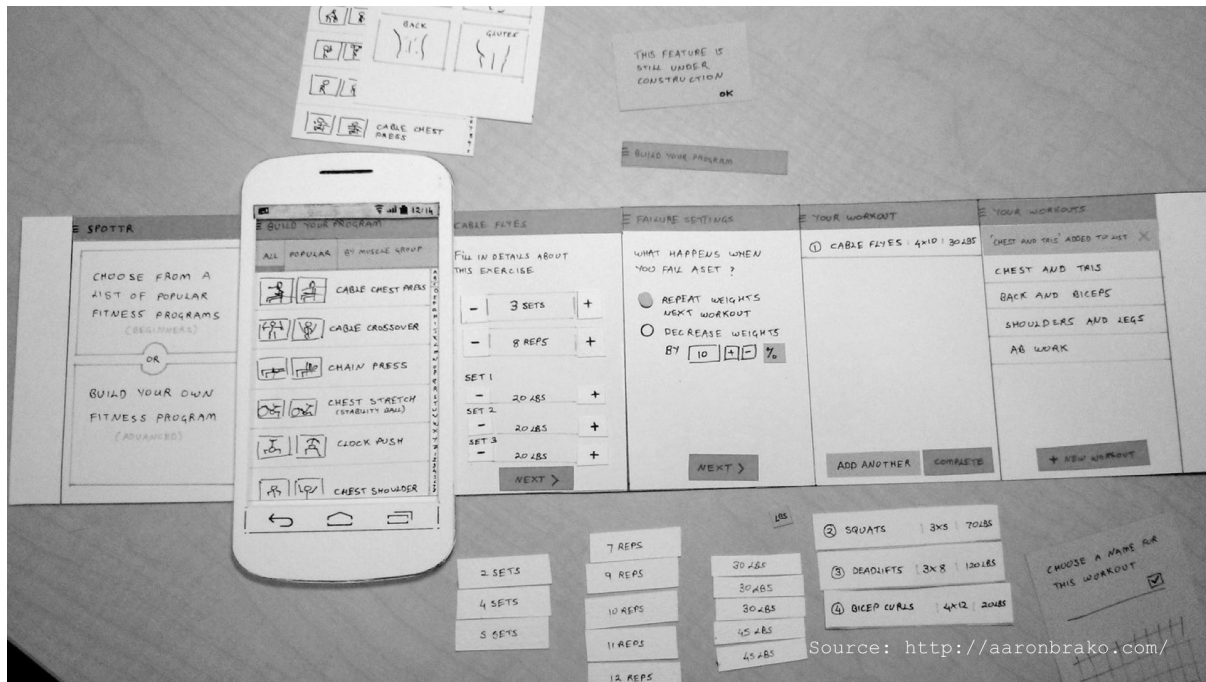


Figure 3.2: An example of a flip-flop paper prototype. Source: [2]

3.1.5 Evaluation

When the prior stages of the development process are done, they need to be evaluated. One needs to know if the product satisfies the users' needs and expectations, as well as if requirements that were established earlier in the process are implemented in the desired way. This is done by letting users test the prototype and give their feedback. The user evaluations have to be analyzed, so the design can be modified accordingly. This type of evaluation is commonly referred to as a formative evaluation. Henceforth, when the development is finalized, summative evaluation is utilized. The method involves collecting data for assessing the quality of the product [17].

3.2 Layout patterns

As part of the design process, an established pattern for defining the design's layout is needed in order for the design to function accordingly. Every design layout element must have its purpose, and there needs to be a way to formulate the possible effects they have on how the user observes the design. Fortunately, there exists a famous diagram that can help understand the workflows, as described in figure 3.3.

3.2.1 Gutenberg diagram

The *Gutenberg Diagram* [3] is an illustrative description of how the user observes design elements visually when looking at evenly distributed information on, for instance, a web page. The diagram is very direct, in the sense that what is being illustrated is essentially the pattern in which the user will move their eyes.

The pattern expresses that the user moves their eyes downwards with slight horizontal movements. The origin of every movement gradually shifts to the right as the user moves down. Consequently, the eye-movement moves from the **primary optical area** toward the final **terminal area**. Again, see figure 3.3. This behavior is similar to how one would read a book. Simply read every sentence from left to right while going down a line and repeat the process.

With the concept of reading patterns in mind, the Gutenberg diagram reflects upon how some areas in a particular interface may fall short by simply going unnoticed. Important information in the **weak fallow area** will almost always go unnoticed and result in a flawed design. Yet, it is important to consider that these concepts apply greatly to designs with low *typographic hierarchy* ² and does not work for all designs.

²When design elements share great similarity [18]

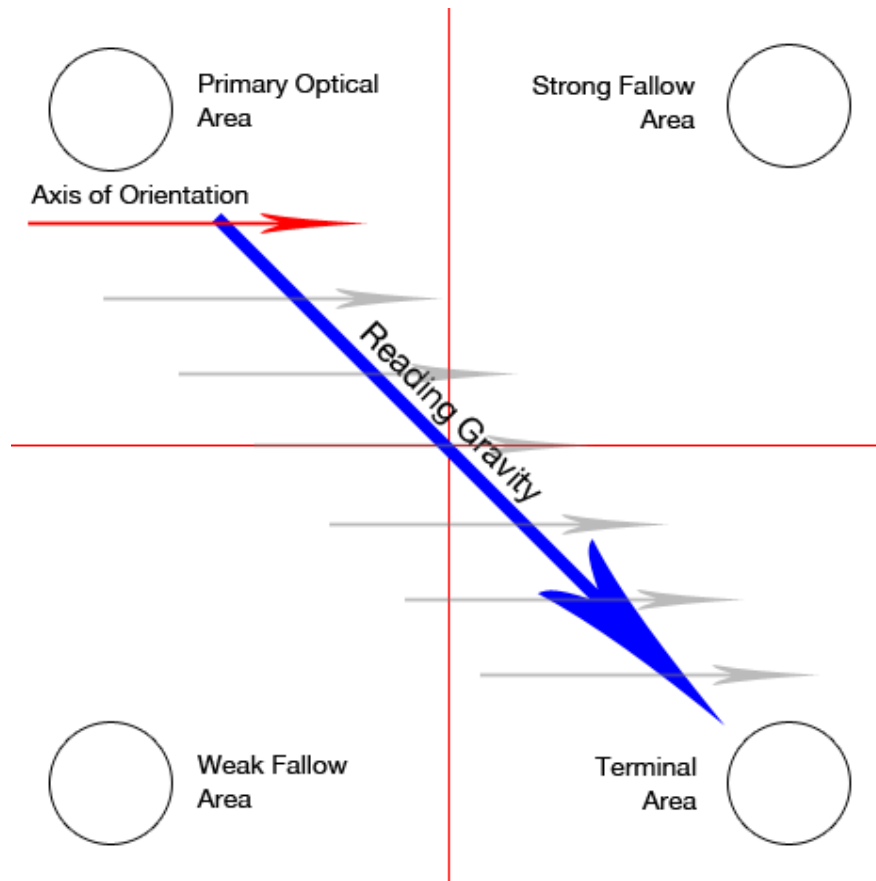


Figure 3.3: The Gutenberg diagram [3]

3.3 Emotional design

“Everything has a personality: everything sends an emotional signal. Even where this was not the intention of the designer, the people who view the website infer personalities and experience emotions.” — Don Norman, Grand Old Man of User Experience [19]

With respect to the user-centered design being a user-requirement-focused design process, it is crucial to recognize that the users are humans, who are emotional beings.

Emotional design is the act of creating designs that bring forth emotions in the user when interacting with a product. Responses from any design are predominantly emotional, thus opening the possibility to utilize design techniques that reach the users on a cognitive level.

3.3.1 Cognitive aspects

A design needs to be visually and functionally appealing. This allows users to more easily rationalize the product. Such concepts are represented in Donald A. Norman's "Three Levels of Emotional Design" [20]. The model consists of three contiguous categories, also referred to as **levels**, that describe how a design should influence the user.

The top-level, **visceral** design, describes visual aspects of the product. This helps distinguish what the product's capabilities and purposes are on a superficial level, which separates it from its competitors. This means that it lays the ground for the identity of the product, which is why it is commonly associated with branding. Visuals are closely tied to **behavioral** design, the practical and functional aspects of a product since this can change how a product feels to the user. When users interact with a design, they want to achieve a goal within the product's usage in a satisfying fashion. Ease of use and the feeling of a product can heavily impact how the user perceives it too, and this must be prioritized in the design. This is known as **reflective** design. This entire model can be visually interpreted in figure 3.4

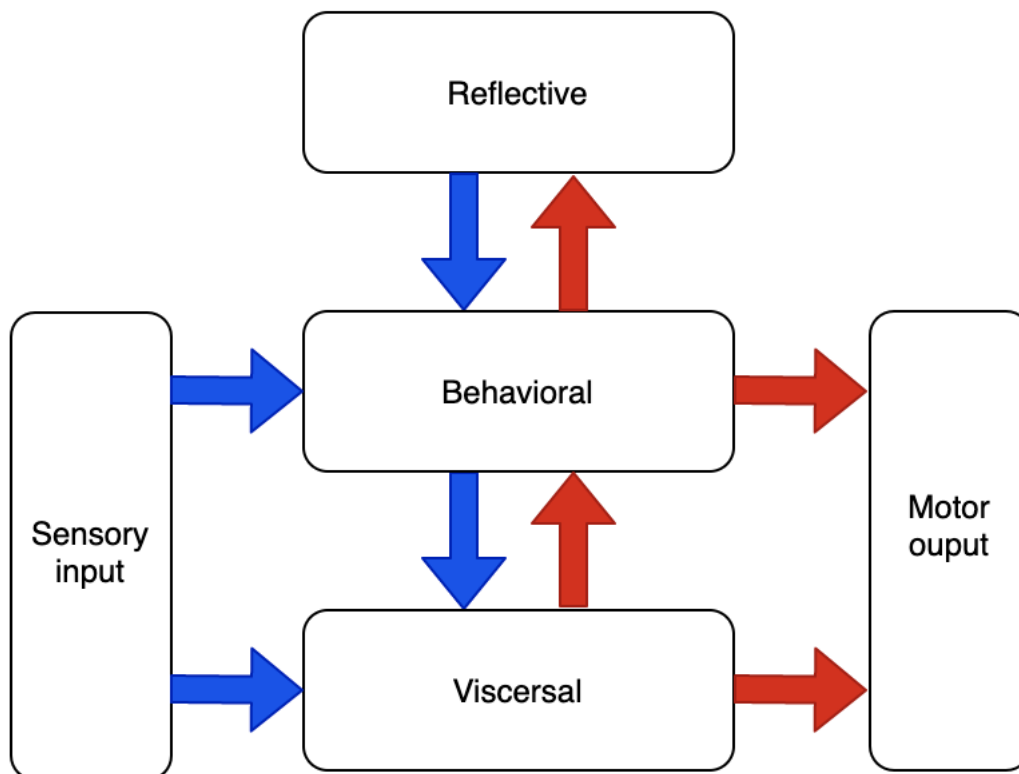


Figure 3.4: Donald A. Norman's levels of emotional design

Users actively judge the product's performance and benefits in accordance with their nuanced and rational thinking. Designing around user perception is a powerful tool that opens the possibility of users identifying with the product, rendering the shortcomings of the product less problematic to the user.

3.3.2 Emotional states and cognition

A brief clarification on the relationship between human cognition and emotional reaction is necessary to understand the possible outcomes of interaction. Psychological elements like anxiety, anger, hope, and relief [21] heavily impact how the user interacts with the product sequentially. If the product is subject to cause negative emotion in the user, then the concentration is lowered, thus rendering necessary elements in the interface disregarded. As follows, the user may develop further negative reactions such as stress if those features were the key to progress to a different section in the interface. Another important element is attention to detail. If positive reactions are evoked in the user, they are less likely to judge smaller elements within the interface.

3.4 Interaction design and user experience

Interaction design describes, in a broad sense, the practice of designing interactive software products. Most processes of interaction design have already been described in this chapter, such as prototyping and evaluation. This section will explain the design of interactions with respect to the users.

3.4.1 Usability goals

Being clear about the objective of developing an interactive product is the primary part of understanding the users [22]. This particular process involves finding certain goals associated with the user experience, which are described in the first chapter of *Interaction Design, beyond human-computer interaction* [22]. This set of goals is referred to as **usability goals**. As the name implies, it is directed towards ensuring that the product is usable. When setting these goals in the design, the following elements must be taken into consideration [22]:

- Effectiveness

- Efficiency
- Safety
- Utility
- Learnability
- Memorability

In short, the interactive product should be easy to learn, effective to use, and enjoyable for the user. Emphasizing these characteristics when designing results in an optimized interaction. The higher quality of interaction enables users to achieve their goals associated with the product with ease.

3.4.2 Human-computer interaction

Human-computer interaction is a study about how humans and computers interact. It is a multidisciplinary field [4] which intersects with computer- and cognitive science, as well as human factors engineering. Figure 3.5 will always be a relevant model in the process of interaction design.

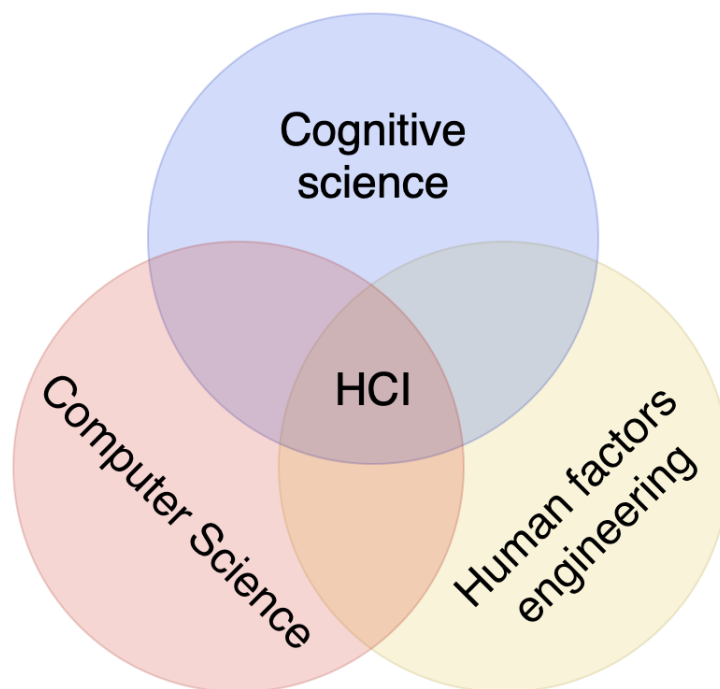


Figure 3.5: HCI - multidisciplinary field of study. Inspired by: Human-Computer interaction [4]

4 Implementation and testing methodology

Implementation of ideas is a process that requires a set of knowledge about the path of the desired design. One cannot simply start designing without a basis for the product's usage, and that itself is not a simple concept to dive into. Finding a way to constrain ideas to a reasonable limit is a matter of learning where, how, and why the product will be used, and especially who will be using it. It all comes down to finding a constraint, which is a very positive trait that supports planning.

4.1 Project planning

When this thesis was planned, it was divided into three main phases. The process is illustrated in figure 4.1. The first one is the research and understanding of the various design and testing principles, followed by generating the user requirement document and evaluation. The second part includes technical decision-making and designing the software architecture, which will result in another, more functional, prototype. Besides, technical requirements and architecture are prone to changes and will be adapted to needs along the implementation process. In this phase, we will use the agile method of using *sprints* to have an overview of what has been done and the next steps. Once again, the testing with real users will be conducted. The documentation of the process will be written from the very beginning and will be finished in the third phase. More detailed information is presented in the continuation of the thesis.

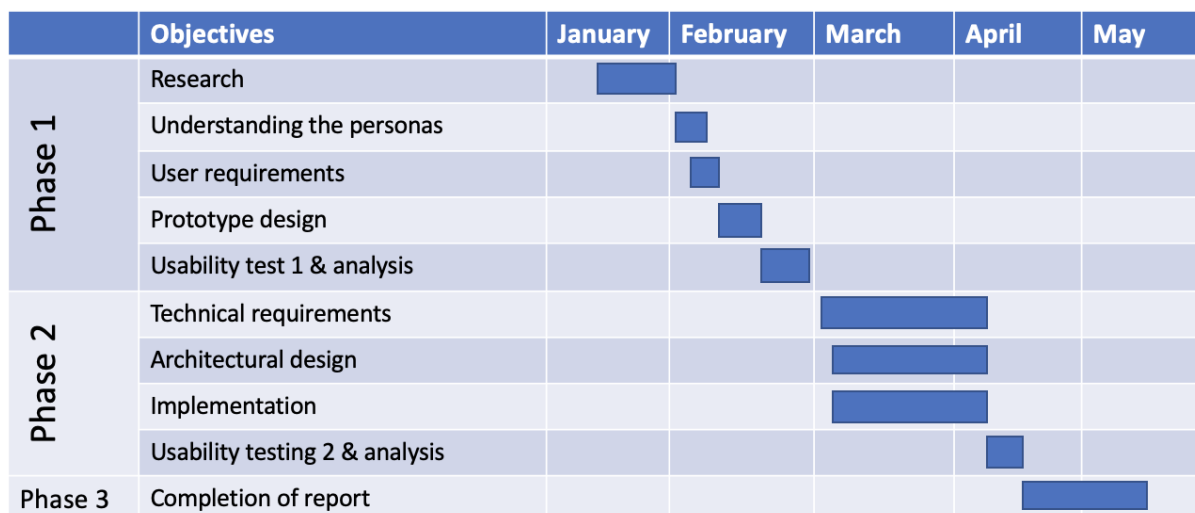


Figure 4.1: A Gantt diagram of the project plan

4.2 Defining personas

Personas are not real people. They are quite the opposite - fictional characters created by designers based on real user data and observations. Personas help one understand the various user groups and their needs, experiences, behaviors, and goals [23]. It gives the development process a human touch one can empathize with and shift their perspective to the user.

4.2.1 Persona perspective evaluation

User personas can be evaluated in a variety of fashions that describe the users from different perspectives. A common one involved in this project is user-story-, or goal-oriented personification. The idea is to identify a user that wants to fulfill a narrative in a certain scenario, and use that thought-out information to note requirements for the product. Within goal-directed personification, the user's role is taken into consideration. In the previously mentioned example, the user is a student, but it is not clear that the product in question will only involve students as users.

Viewing a persona from the user's role mediates the understanding of the product's usage in a wider or global sense. The following project focuses on two user groups that utilize the product in two different ways that both have similar traits and inequalities which fit into the category of the product. Roles encapsulate business objectives and the impacts the product will have on the user.

4.2.2 Persona definition process

A persona can be defined by first identifying who the character is, what conflicts they may undergo, and the goals they want to achieve. This project uses a method that notes down the character's background, situation, and goals related to the product, along with what the character expects from it.

4.2.3 Persona categories

There are two main users of the platform: the teachers and their students. The teacher is a type of facilitator of the platform. They are supposed to provide material to students,

administrate and explain it. Furthermore, a teacher should arrange a form of evaluating and certificating students. They will also function as the primary advisor for the courses.

The second user type is the student. Its profile is represented in figure 4.2. These people are the main consumer of the platform since they have no responsibility for the published content, only to find and consume it and possibly provide feedback as a contribution to the platform. In addition, the students must get a learning benefit from the material along with a learning certificate.

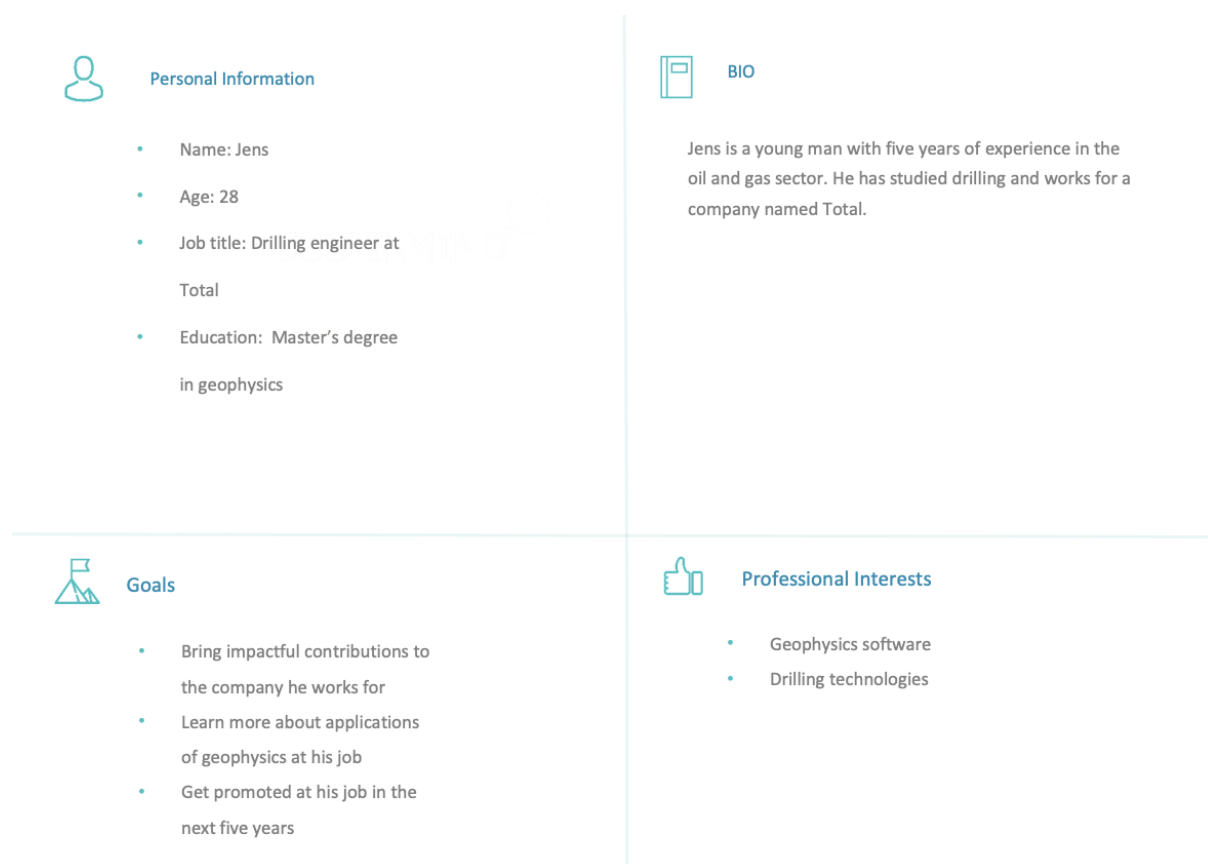


Figure 4.2: An example of a persona named Jens

4.3 User stories

As opposed to a persona, a user story is a natural description of the system functionalities from the user's perspective. User stories are useful for determining software requirements as well as the designer's understanding of the user in practice.

Creating a user story can be achieved in a multitude of fashions, such as having a real user write it or just make it up. The *Connextra Format* [24] is a useful method to represent a user's wishes in accordance with the product. Due to the limitations of resources, this project will make up a user story using a flow chart that covers all possible scenarios from a user perspective and that particular user's choices. In chapter 4.3.1, an example of such a user story is described, along with an illustration in figure 4.3.

4.3.1 Jens' story

Following the *Connextra Format* [24], Jens' story can be interpreted as follows:

“As a drilling engineer I want to learn how to use geophysics software so that I can use those skills in my future work.”

As Jens has a full-time job, the best possible way for him to achieve his goal is to engage in e-learning. He got a recommendation from a friend about e-learning specialized in the oil and gas sector, which might fulfill his interests. Before Jens starts looking for a course, he needs to register for a user account. Following up, Jens starts searching for a course teaching Techlog Wellbore. He sees that there are many positive reviews on a particular one, so he applies for it. As he learns the course, he needs help from time to time which is why Jens used the course forum to clear up some confusion about the material. At long last, when he is finished with the course, he gets a certificate proving his understanding of the geophysics software Techlog Wellbore.

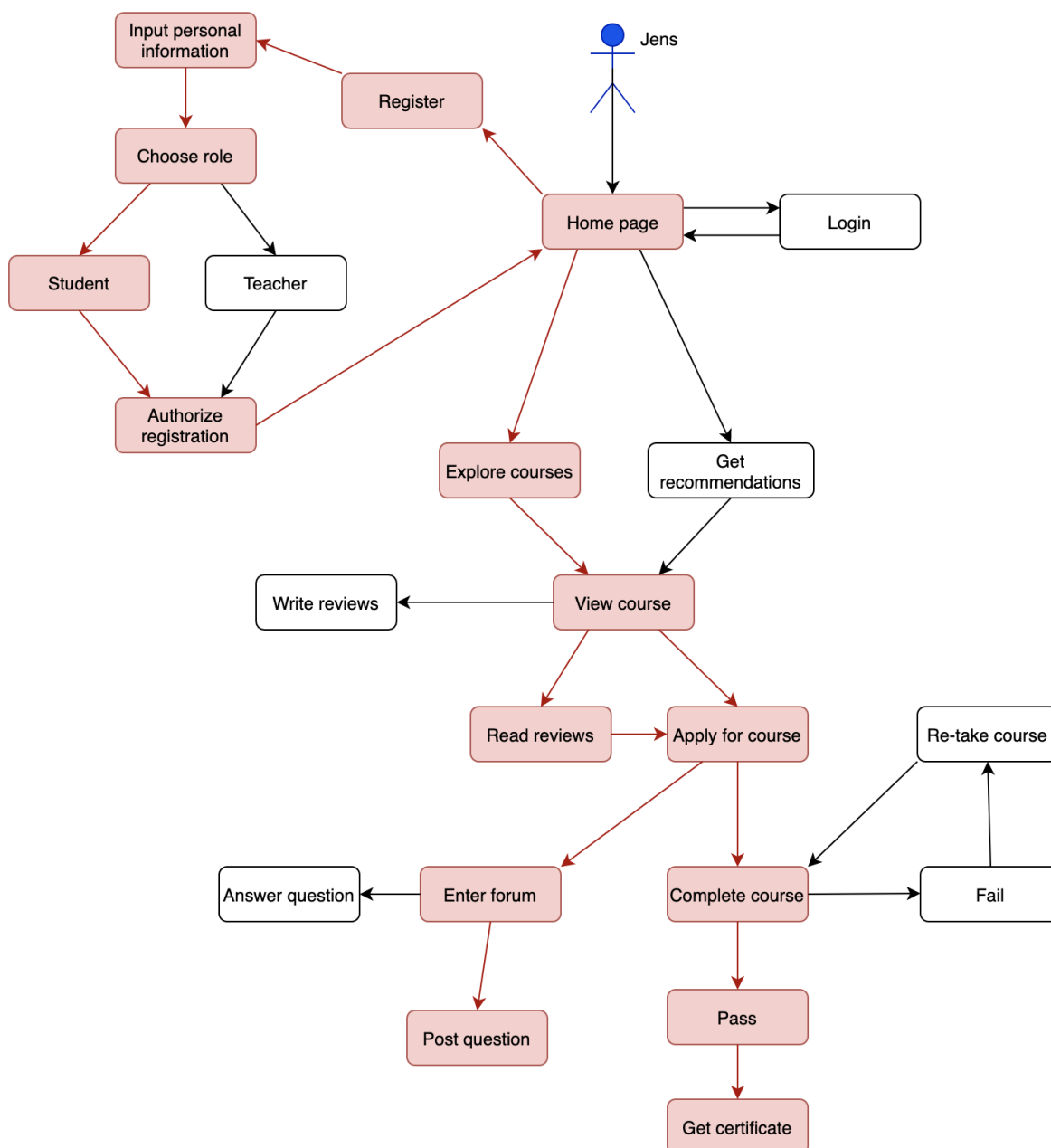


Figure 4.3: Jens' user-story

4.4 Requirements

"Requirements are a specification of what should be implemented. They are descriptions of how the system should behave or of a system property or attribute. They may be a constraint on the development process of the system." [25]

It is essential to establish a formal agreement about the product's outcome between the client and the designers [26]. Software requirements is a term that implies the software's

needs on an obligatory level, yet commonly throughout the software development life-cycle, business decisions always vary. Therefore, the term can be interpreted in a magnitude of ways; however, the thesis will focus on two essential categories of software requirements that define functional- and user needs.

The two categorizations of requirements are formally referred to as functional and non-functional requirements. The former describes the exhibited behavior of the system under certain conditions. In contrast, the latter describes properties or characteristics that the system must exhibit or a constraint that it must respect [25]. Functional requirements are associated with **what** the system does, and non-functional relates to **how** the system does it.

4.4.1 The requirements change

As cited in the first chapter of *Software Requirements, Third Edition*, by Karl Wieggers and Joy Beatty [25], requirements development is subdivided into elicitation, analysis, specification, and validation. These are essential principles, especially in user-centered design, since elicitation and analysis are most relevant and are discussed further in chapter 4.6.

An iterative process accounts for and is subject to constant changes caused by multiple factors. Firstly, users are prone to changes when it comes to expectations and requirements. Commonly, clients of software products rapidly change their perceptions and requirements associated with the product and are often unsure about what they want as the project progresses, making the software validation a difficult yet crucial part of the development. Henceforth, customer and stakeholder feedback is the most valuable form of information that helps discover a possible otherwise overlooked requirement. This also counts for the development process itself. The team has to note down even more needs for the software as they discover further problems, such as code that requires additional performance solutions.

Following up, the requirements must be managed accordingly. The goal is to anticipate and accommodate drastic changes to minimize the disruptive impacts of such changes, which is fundamental in an *agile* software methodology. All of the processes discussed in this chapter have to lead up to the very reason discussed here: change. In the next

sub-chapter, a document referred to as a *software requirements specification* is mentioned in depth; thus, it is crucial to consider that the information contained within this document will not be constant.

4.4.2 Software requirements specification

In order to keep track of every requirement in a software project, some form of documentation must be put in to use. The industry standard is to write a *Software Requirement Specification*, also known as an SRS, which is a document containing information about the requirements. In the tenth chapter of *Software Requirements, Third Edition*, by Karl Wieggers and Joy Beatty [25], there is a description of how such a document is structured. The structure is presented with a lot of depth, but it can be generalized into the following principles:

- Product description
- System features
- Data requirements
- External interface requirements
- Quality attributes

These principles can easily be translated to functional and non-functional requirements, which is easier to follow since this project is directed to developing a functional prototype with proper usability. The document of requirements used in this project can be found in Appendix A1. The product description is essential to the document introduction. System features and data requirements go under functional requirements, while external interface requirements can be a mix of both functional and non-functional requirements. Lastly, quality attributes are somewhat by definition non-functional requirements.

4.5 Architectural design

Architectural design is defined as the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system, as described by the Institute of Electrical and Electronics Engineers

[27]. The design has various architectural styles, depending on the needs. There are data-centered-, data flow-, call and return-, object-oriented and layered architectures. Each style is set to structure components of the system. One has to think of components that will perform required actions by the program and how to implement those components. Furthermore, the set of connectors that will help in coordination, communication, and cooperation between the components is a crucial factor [27]. Last but not least important, there has to be semantic models that help the designer to understand the system.

4.5.1 Technical requirements

For a successful project, it is important to think about various requirements, some of them being software testing and deployment. A critical aspect is to evaluate the different technologies to be put into use. This project is divided into two parts: the front-end and the back-end.

The front-end part aims to make a user interface concerning the research done and feedback from the real users. The second part is to use an existing data management system to store relevant material and make a proper connection to its primary source. With these two components in mind, a back-end that manages the application's core logic is also required.

4.5.2 Choice of database management system

Choosing the appropriate database management system, or DBMS, can be delicate. One has to consider several factors and find a way to select the best. The method used to determine which DBMS will be used in OSDU Academy is based on a rating system. Five different systems were considered and rated by eight criteria on a scale from one to five, where five is the best. One of the measures tested was how familiar we are with the particular database system, as it would make the job easier. Furthermore, other relevant criteria were multimedia support and scalability. These two measures were present as it is important to have a variation in the learning process media, i.e., video and text along with a healthy way to extend the system. Some other criteria considered were:

- **Compatibility** - How compatible the system is with various programming languages and operating systems

- **Open-source quality** - Whether the DBMS is free to consume or perhaps licensed. If latter is true, is there a way to get a license for academic purposes?
- **Usability** - How hard it is to learn and use the system.
- **Support and development** - Is there enough documentation and help provided, and how frequently does the organization come with new and updated features.
- **Security** - General data security compatibility.

In figure 4.4, there is a table of ratings given to each popular database management system we found online. Note that the ratings are very subjective to our requirement specifications.

| DBMS Engine | Familiarity | Multimedia support | Scalability | Compatibility | Open-source quality | Usability | Support and Development | Security | Average score |
|----------------------|-------------|--------------------|-------------|---------------|---------------------|-----------|-------------------------|----------|---------------|
| MySQL | 3 | 3 | 3 | 5 | 5 | 3 | 4 | 4 | 3,62 |
| Oracle Database | 1 | 4 | 4 | 3 | 1 | 3 | 2 | 3 | 2,62 |
| MongoDB | 1 | 3 | 5 | 5 | 5 | 3 | 3 | 5 | 3,75 |
| Microsoft SQL Server | 1 | 3 | 3 | 4 | 5 | 4 | 3 | 4 | 3,38 |
| PostgreSQL | 1 | 3 | 5 | 5 | 5 | 3 | 3 | 5 | 3,75 |

Figure 4.4: The rating table

After evaluating each DBMS engine using a table of subjective standards, there were two systems with the same highest average score. The two competitors ended up being PostgreSQL and MongoDB. In the end, we chose to use MongoDB as the main DBMS for OSDU Academy, which was done due to two important factors. Firstly, the system uses a familiar query language similar to JavaScript, which we think makes using the database easier. Secondly, MongoDB is a non-relational database, which means that issues that a relational database, like PostgreSQL, would otherwise have will not be present. Issues like these include normalization, which is a large topic.

4.5.3 Front-end programming utilities

Programming utilities are better known as software development tools. These are used to make the development process as simple as possible. Besides, there are several purposes for utilities, such as maintenance and optimization. Other software utilities are the frameworks and software development kits.

In the process of planning the development, we have thought of assorted frameworks and software development kits for the front-end. We tried to find programming utilities that were familiar to us among the following three categories of utilities:

- **JavaScript libraries:** Angular, React, Vue
- **Cross-platform mobile SDKs:** Xamarin, Flutter
- **Web page template libraries:** Razor, Jinja2, Thymeleaf

In the end, the shortlist consisted of ReactJS, Flutter and any web template library depending on the back-end solution. Narrowing down to these three libraries made it easier to pick which category to bring into development since the most familiar and suitable solutions were already ready to be picked.

4.5.3.1 ReactJS

ReactJS is a front-end web library for JavaScript. It is built to put together the HTML and JavaScript portions of a front-end web project together using a language called JSX, making it easier to develop interactivity for the interface. Furthermore, ReactJS is a declarative framework, meaning every part of the user interface is defined separately as components. It is also very robust due to the easy ways to update the interface live. Lastly, React has a wide variety of extensions commonly used in the field.

4.5.3.2 Flutter

Flutter is an SDK developed by Google and features the programming language Dart for programming user interfaces. It is one of the most popular choices for building cross-platform applications as it gives a chance to develop relatively fast and allows for re-using code. Another positive trait with Flutter is that it is possible to use it for the web, yet it is difficult to set it up. Besides, Flutter is quite new and is developing rapidly. One can imagine how difficult it is to find relevant documentation and help, not to mention that the maintenance in the future would be challenging. Otherwise, the biggest issue at this moment is that Flutter is not a very typical SDK which makes it more difficult to handle.

4.5.3.3 Web page template libraries

Many back-end solutions already come with, or support, web-page rendering using a template pattern. Templates are essentially HTML-documents that usually support embedded code written in the language the back-end is coded in. These libraries make it very easy to communicate with the back-end since the embedded code will always be contextual to the data sent to the client. Most template libraries are very similar and function the same way, only that the syntax is different. Examples are Razor, Jinja2 and Thymeleaf and are all supported by the back-end solutions discussed further in this chapter respectively.

4.5.3.4 The final choice

Considering everything, the final decision ended up being **ReactJS** as part of building OSDU Academy. The main reason is that we are familiar with the technology to some extent. Coding in pure JavaScript requires us to account for supporting different web browsers, and ReactJS already takes care of that. Additionally, it is a very consistent single-page application framework, so transferring the design to other platforms will be considerably easier.

4.5.4 The back-end solution

The main application will rest in the back-end, meaning that the choice of framework will be a delicate decision to make. This is because the underlying foundation of the application's structure and functionality will be built from a programming interface that could possibly be limiting depending on the choice. In many cases it is wise to use limited frameworks if time is a factor. Other occasions that require demanding calculations and algorithms may require something more powerful and performance-oriented.

4.5.4.1 ASP.NET Core

ASP.NET Core is a collection of libraries packed into a framework for building web applications written in C# by Microsoft. It is the successor of ASP.NET and is open-source. ASP.NET Core is commonly used for developing cross-platform cloud-based web applications. Other frameworks such as ASP.NET MVC and ASP.NET Web API are

packed into a into a single programming model in ASP.NET Core. Since it is written in C#, there are many opportunities for middle- to high-performance computing in larger applications. In addition, it comes with web-page template rendering using Razor, but there are other options.

4.5.4.2 Flask

This is a Python-based micro web framework for the back-end. Flask's advantage is the front-end capabilities it has integrated with its back-end technology, namely that it uses Jinja2 as its template rendering engine. Moreover, Flask is flexible, meaning that many parts of the framework can be changed. As a result, it has many extensions provided by different developers. The variety of extensions consist of many essential solutions such as security. Yet, Flask alone is not very scalable and handles requests one at a time, making it a fairly unsuitable candidate for real-time applications. In conclusion, the development using Flask is easy and straight-forward, but it has its limitations performance-wise.

4.5.4.3 Spring Boot

Spring Boot is an framework written in Java by the Pivotal Team. This application programming interface is used to develop microservices. A microservice is an architecture that allows developers to create and deploy services independently. Spring Boot features web-page rendering using Thymeleaf, similar to Razor using ASP.NET. It also features frameworks for back-end web development design patterns like the model-view-controller pattern.

4.5.4.4 The final choice

In the end, we chose to use **ASP.NET Core**. Despite knowing little about the business model for OSDU Academy, it is plausible to take a highly capable back-end written in C# since we expect that the platform will be scaled a lot in the future. It is not the framework alone that interested us, but also the language C# since it is very powerful in many other ways than performance, like the extensive language features. The wide variety of possible design patterns that ASP.NET Core uses is also a deciding factor since this availability is a very good trait to have when accounting for changes.

4.5.5 Design patterns

After choosing suitable software development technologies, several design patterns for implementing the product should be established. The design patterns sometimes come in the API designs themselves and developers should stick to the norms where it is needed. Despite using several design patterns in a project, it is still possible to define the architecture using one core pattern with other sections defined by their respective ones. An e-learning platform is provided through one central server that contains the underlying application logic. The user is provided with a specific client application that is architecturally designed to follow the logic of the server in a user-friendly form factor. This model, independent of application type, is generally known as a **Web API** and is the conventional way to refer to a website application or cloud service. This section will describe each design pattern brought into the implementation process with its advantages. We wanted all the design patterns to assist in achieving multiple goals for the final product.

4.5.5.1 Service layer

At a very basic level, the service layer is the most common interface to an application logic that different clients, like a web interface or command line tool, can use [28]. A service layer can provide the necessary functions that any other client can use to communicate with the application server from the client code. These functions should make calls to the server and its database to perform the desired logic. This design pattern offers several advantages that makes the application easy to extend, test and port to other forms of client applications. It is also easier to create documentation since the abstracted functionality does not concern itself with internal logic.

4.5.5.2 REST

Representational State Transfer, or REST, is an architectural design pattern that defines a set of constraints for creating Web APIs [28]. It is most often used with Hyper Text Transfer Protocol, or HTTP, but does not define it. Instead, it defines a client-server architecture managed by HTTP in a stateless³ fashion. REST is a layered system that implements a uniform interface that should be easy to comprehend due to its simplicity

³No client information is stored between HTTP requests. Each request is separate [29].

and standard nature. Since REST uses HTTP to communicate, it allows for it to be consumed by any end-user device or application client. Again, this pattern opens the possibility to communicate with any type of client application, making platform portability much easier. This pattern also complements the service layer pattern.

4.5.5.3 SPA

Single-page application, or SPA, is a common design pattern for client applications that communicate with the server side. For most web-based applications, navigating to different sections requires the client to load an entirely new page. SPAs do not perform this action, meaning that the user stays on one page as long as they're using the website. This pattern is both advantageous for the server side and client application design. Firstly, the server does not need to send data that contains information about the page layout each time the user navigates. It is done only once and the rest of the workflow is handled by the client while it requests data and receives payloads⁴ when needed. Secondly, the client application design will be easier to transfer to other platforms than just running it in a web browser.

4.5.5.4 MVC

The model-view-controller, or MVC, pattern is a three-part architectural pattern that separates an application into three interconnected parts: model, view and controller. This setup is used to separate internal data representations from the user interface, thus making the architecture uniform and more comprehensible. The first part is the model, which is the central component of the application managing all logic and data. The view is a representation of said data in the form of a graphical user interface. Lastly, the controller is what accepts user input and turns it into commands for the model or view. All of these components interact with each other which results in the full application.

This specific design pattern is a crucial one in OSDU Academy. Nearly all logical aspects of the program is handled through the MVC pattern interconnected heavily with SPA and the service layer pattern. Each aspect in the back-end is separated into a controller where the required logic is handled. More about this is described in chapter 3.

⁴A batch of data sent across networks

4.5.5.5 Middleware

The patterns covered so far have described most of the core architecture of a Web API, but there some important aspects that the Middleware can take care of in the back-end. A Web API needs a system that can manage user permissions and validate data, which the Middleware pattern is used for. This pattern defines a component which modifies each request and response in the application in order to perform a certain requirement. Normally these requirements would be for data validation and permission management, as discussed earlier. Using one or more Middlerwares for actions like validation makes the code more uniform and eliminates repeated issues when implementing a specific request protocol. The Middleware pattern complements many design patterns about requests and data management across networks.

4.6 Usability testing

A product is meant to help complete a task. Knowing how well the task can be completed requires testing the product's overall usability. User interfaces are, more or less, complex systems that require an extensive setup, thus an equally quantifiable test. Usability tests identify flaws with the design and where the users may have trouble with it, which in turn mediates the product's improvement. Such a test consists of real users who are given a task using the product in a controlled environment. The designers then observe the user's actions and take notes on their performance. The more conducted tests, the more data can be collected, hence a better product.

4.6.1 Testing goals

Identifying the goals and needs of the users is the way to find areas to test in a product. Using that information helps developing scenarios and tasks for the testing participants. Generally, designers look into who the users of the product would be, as well as their goals with it. As follows, one can use that information to create hypothetical scenarios and tasks for the test subjects. The goal is to evaluate the interaction between the product and the users, which is a crucial feature of usability testing. With all that in mind, the design elements that cause problems become clearer. In addition, it becomes easier to know how to extend the product with additional features.

4.6.2 Test setup

There are two main contributors under a testing session, the facilitator and the tester. The facilitator is the person who controls the test and observes behavior and is in no way supposed to help participants solve problems. They are supposed to ask neutral questions and motivate them to find their answers. The facilitator should have at least one observer who can help note the data, comments, and common behavior. The testers should try to solve problems presented in the scenarios and tasks and give their opinions when it comes to different aspects of the product. Additionally, everyone who can benefit from observing a test, including the developers, managers, and technical support, can be present without having an active role in the session.

Usually, a usability test has two or three questionnaires, one which takes place before the tester starts with tasks, one after every task, and one which summarizes the overall experience. The pre-test survey collects general demographic information about the participant, and the post-task survey gathers data such as satisfaction level of use for each task [30].

4.7 Prototype 1 testing

The first prototype is a low-fidelity one created in Microsoft PowerPoint [31]. Based on the previously discussed testing principles, we will introduce the ways the prototype coincides with them and their important components and methods.

4.7.1 Demographic group

The participants collected for this usability test consisted of five employees of Schlumberger. Our supervisor contacted them and convinced them to participate in the usability test. As mentioned, these are employees of Schlumberger, where most of them were engineers. Two of them are software developers. The rest are leaders in other fields, so the slight variations in their demographic allow for some interesting results considering a few already have some experience in design.

4.7.2 Testing environment

The place where the test was conducted could not be done in person due to the COVID-19 pandemic. Instead, it had to be done over a video call since personal interaction was prohibited to some extent. Each participant would be invited to a Microsoft Teams call via email and enter it when it was their turn. For the sake of analysis, the calls were also recorded. After introducing the test to a participant, it would be disclosed that the call is under-recording, which everyone agreed to.

The key to the usability test was to elicit comfort in the participants. No matter where the test is conducted, the tester must feel at ease to comment as much as possible on certain elements of the design. A psychological element that mediates this is to be chatty and open the conversation with lightheartedness. Since this is a test and the participants are aware of it, they can possibly expect formality. This is especially important since the test was being recorded. One has to start slow, give an introduction and describe what is going to happen. In this way, the facilitator gets the credibility and the participants become more relaxed.

For the test's data to be analyzed, it had to be noted by an observer that sat next to the facilitator of the test. As the facilitator guided the participant through the prototype when they made a choice and thus responded, the observer had to take notes on their behavior as follows.

4.7.3 Scenarios and tasks

Scenarios are believable assignments the user should accomplish by using the given prototype. Every scenario has a particular task. These are used to uncover difficulties and issues within the design. Moreover, we wanted to find out the users' common behaviors and test the learnability of each task. Before beginning the testing session, each user got an email containing a description of every task. The complete scenario document is included in appendix A3.1. To achieve the best results, we did not disturb the participants while doing tasks, except when they would be stuck on one. The users did not get any detailed explanation about the prototype other than how to interact with it. This helped to bring out their natural reactions since there was no expectation set.

4.7.4 Testing platform

Low-fidelity prototypes are very cheap to produce, especially on a time-crunch. Due to COVID-19 regulations, the designers of this project could not meet the testing participants in person, so different measures had to be taken. Normally, low-fidelity prototypes are made by pencil and paper and are interacted with step-by-step by the tester in person. Nevertheless, the same result can be achieved using digital programs. Using Microsoft PowerPoint [31], this can be achieved since the program not only supports placing shapes and text but also hyperlinks that lead to other slides and pop-ups. The extended tools allow for creating a prototype that looks more like the end product. The slides substituted papers in a traditional paper prototype and were displayed to the participants through video calls. This makes the experience slightly more interactive since it is closer to how a user interface operates.

4.7.5 The method

The method which was used is called "Think Aloud." It is a brilliant and easy-to-learn testing process [32]. The users were encouraged to comment as much as they can while performing their tasks, in addition to answering post-task surveys and a post-testing questionnaire. Before the test was conducted, the users were asked to answer a couple of demographic questions that will be taken into consideration in later analysis. This includes:

- **Name:** This was an optional question.
- **Gender:** The users were not obligated to answer this one either.
- **Age range:** Affects how the user learns the software.
- **Occupation:** Occupation might have an impact on how they perceive the design.
- **Highest education:** Distinguish how good their potential skills are in the subject of software design.
- **Use of e-learning:** Familiarity the users have with similar products.

These questions exist as an ice-breaker for conversation. It makes participants more comfortable and open to talk about how they feel and think during the session.

Furthermore, using this information beforehand means that one can determine common behaviors in certain demographics. For instance, someone who works in software engineering will, most likely, figure out how to use the interface faster than the average person. Nevertheless, it is key to have a varied demographic actually to see the differences in conduct.

4.7.6 The questionnaire

For additional information, the participants were asked some questions after certain tasking. As previously discussed in chapter 4.7.3, after each task was completed, we would ask them one or more questions about their experience, expectations, or other questions related to the task. Some of the questions were also improvised as certain user responses ended up going against the planned questions' need. Additionally, we would set up questions related to things we would expect the users to find lacking. This way of asking questions reassured and disproved our design decisions and saved us the discussion amongst ourselves. Furthermore, once all scenarios and tasks were done, there would be a post-testing discussion with the participant to ask general questions. Many of the planned questions would already be answered by the user when they thought aloud during the task, so some were improvised there.

4.8 Prototype 2 testing

The second prototype is a high-fidelity prototype created based on the user feedback from the first prototype with further utilization of the design principles discussed in the previous chapters. It is a minimum viable product, meaning that the prototype has limited features but is entirely functional and can serve its purpose.

4.8.1 Demographic group

The participants for this usability test consisted of four individuals who work in the oil and gas industry. The first two testers have also tested the first prototype. They will be subject to testing the memorability and learnability. The other two are new as we wanted to see how new users experience the new prototype. The participants' demographics vary once again, consisting of two project managers, a software engineer, and a consultant.

Considering their jobs, we get to see how different users interact with OSDU Academy from varying fields and mindsets.

4.8.2 Testing environment

The environment where the test was conducted is the same as in the previous prototype due to the current pandemic restrictions, namely done over a video call. Each participant got an invitation to a Microsoft Teams meeting via email and agreed to be recorded under the testing session.

4.8.3 Scenarios and tasks

As the previous usability testing sessions were successful, we returned to the same practices and used the same formula to provide relevant scenarios and tasks. The testers were supposed to complete those and help us find possible issues and inconveniences. We prepared the scenarios to cover the most critical features that need to be tested. All scenarios are documented in chapter 6.2.1. Moreover, each task's main objective was to test the various elements of the user testing, such as efficiency, learnability, and satisfaction.

4.8.4 Testing platform

As mentioned, this prototype is of high-fidelity and fully functional. It is hosted from one of our local computers and tunneled through the service called ngrok. Ngrok is a cross-platform application that allows the development team to host the locally hosted web server on a subdomain of ngrok.io. The participants were provided with a tunneled link that re-routes to our local machine. In this way, the testers get to navigate and control the product by themselves, and we get to analyze their behavior more closely and naturally in a fully functional product that we could monitor. In other words, it is much more interactive than the paper version from the first prototype.

4.8.5 The method

The method remains the same as in the previous testing session. The testers were encouraged to think aloud and provide both positive and negative criticism. As the two testers were also a part of the first usability test, they did not have to answer demographic

questions in the pre-test survey but were updated with the project's progress. Furthermore, as the other two participants were new, we collected the demographic data about them and used the survey as an ice-breaker, once again.

4.8.6 The questionnaire

To collect qualitative data for analysis purposes, the testers were asked additional questions both at the end of every task and after completing the test. Those included everything from how difficult a particular task was to their overall experience. The post-task questions were simple ones with a clear objective to confirm some design choices and understand what could be improved. The post-testing survey serves as a conclusion to the testing session, as it provides more general information for the analysis.

5 The Prototypes

The preliminary versions of OSDU Academy were designed with empirical knowledge about similar products. As prototypes, they are not meant to be complete products and are instead opportunities for trial and error. The first prototype was a low-fidelity prototype that opened the path to the core design. As a result of the feedback and analysis of the first prototypical design, the second prototype became an evolved high-fidelity version of the first one. Each prototype's structure will be explained with reasons for each detail.

An important detail is that these prototypes' design was inspired greatly by Schlumberger's official website [33]. The dark-blue color scheme and a couple of navigation features are very similar to their official website.

5.1 Prototype 1

This prototype is a low-fidelity prototype, meaning that it is not interactive and essentially is a mock-up version of the final product. It was developed using Microsoft PowerPoint [31] by using slides as the different sections of the interface. The design process took into account the various essential elements of user-centered design, such as visibility, accessibility, legibility and language.

As indicated, this prototype is a simplified model of the planned product. Essentially, it serves as a foundation stone for further making, and not all the features stated in the Software Requirements Specification (Appendix A1) are included. Examples are forum and review pages. These two were not crucial aspects of the prototype and testing them was not needed in this iteration.

5.1.1 The home page

The first slide in the prototype displays the home page. See figure 5.1. It consists of three parts. The first and most obvious one is a banner showing the name of the application and is assorted with a chevron. The background image for OSDU Academy is meant to target the consumers in the oil and gas industry.

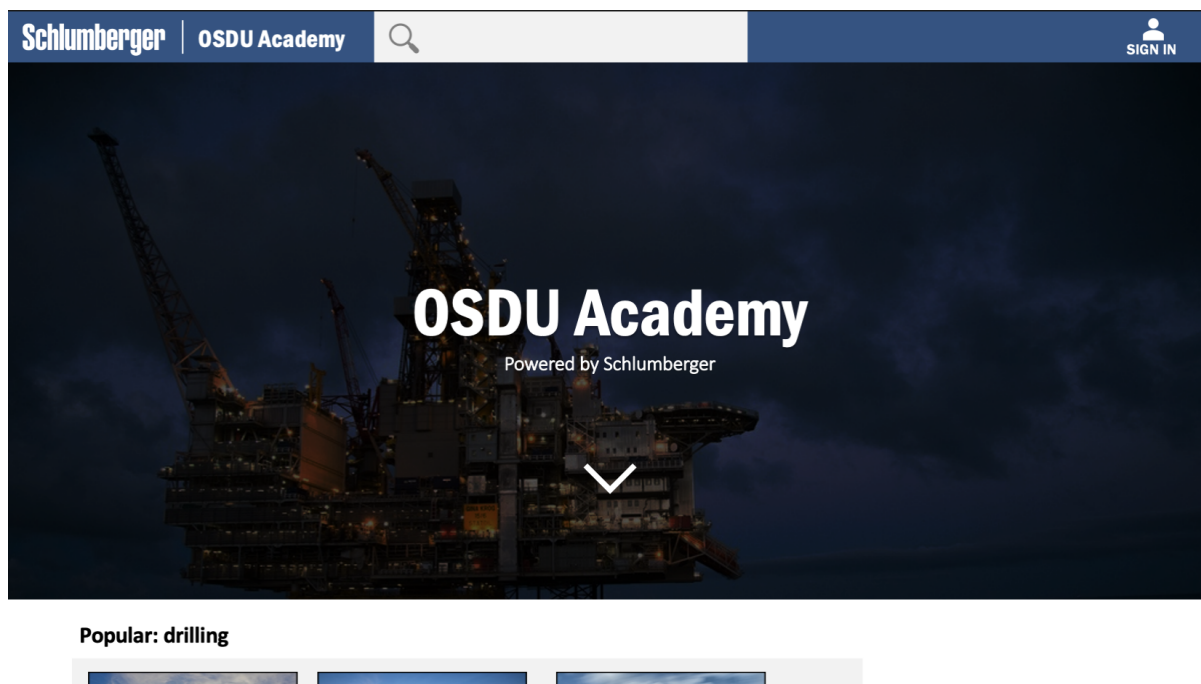


Figure 5.1: Overview of upper part of the home page

Clicking on the chevron would take the user to the lower part of the page. The reason for having courses on the home page was to give consumers a chance to get an idea of course categories. It features courses in popular topics located in an area with navigation possibilities. This only applies to the home page when a user is not logged in; see figure 5.2.

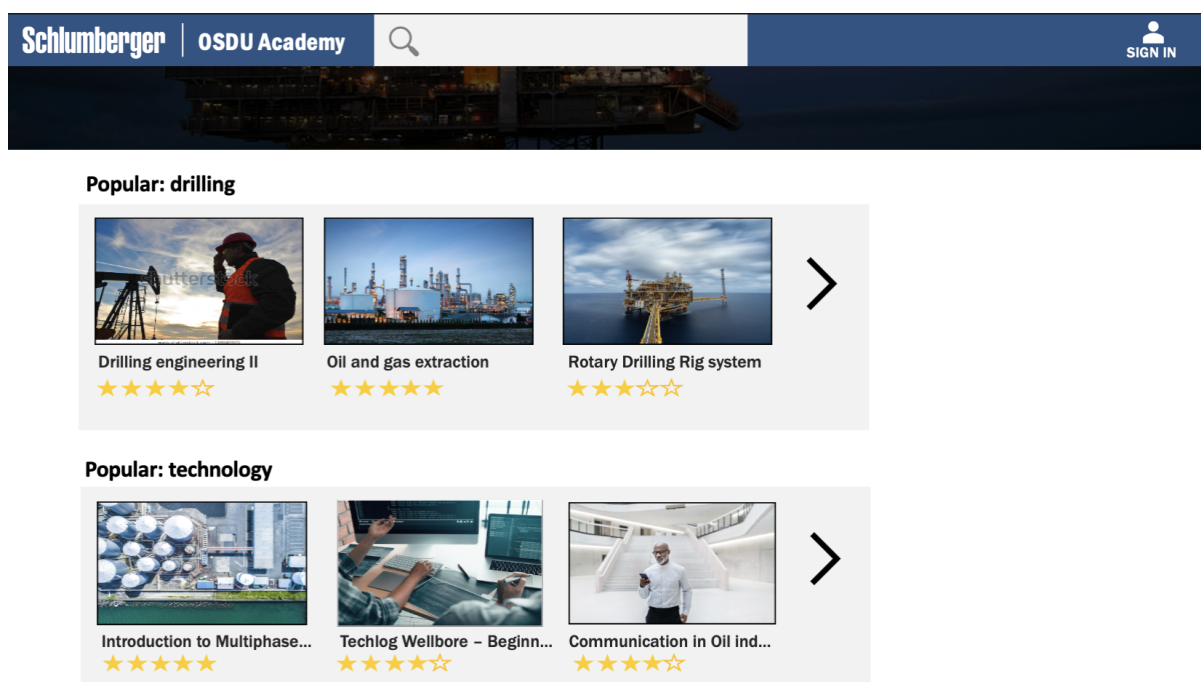


Figure 5.2: Overview of lower part of the home page - not logged in

If a customer has a profile and is logged in, the structure is a bit different, and the content is based on the particular user's choices and interests. Instead of generally popular choices, one can find already started courses under "Course history" and suggestions based on already applied ones, see figure 5.3.

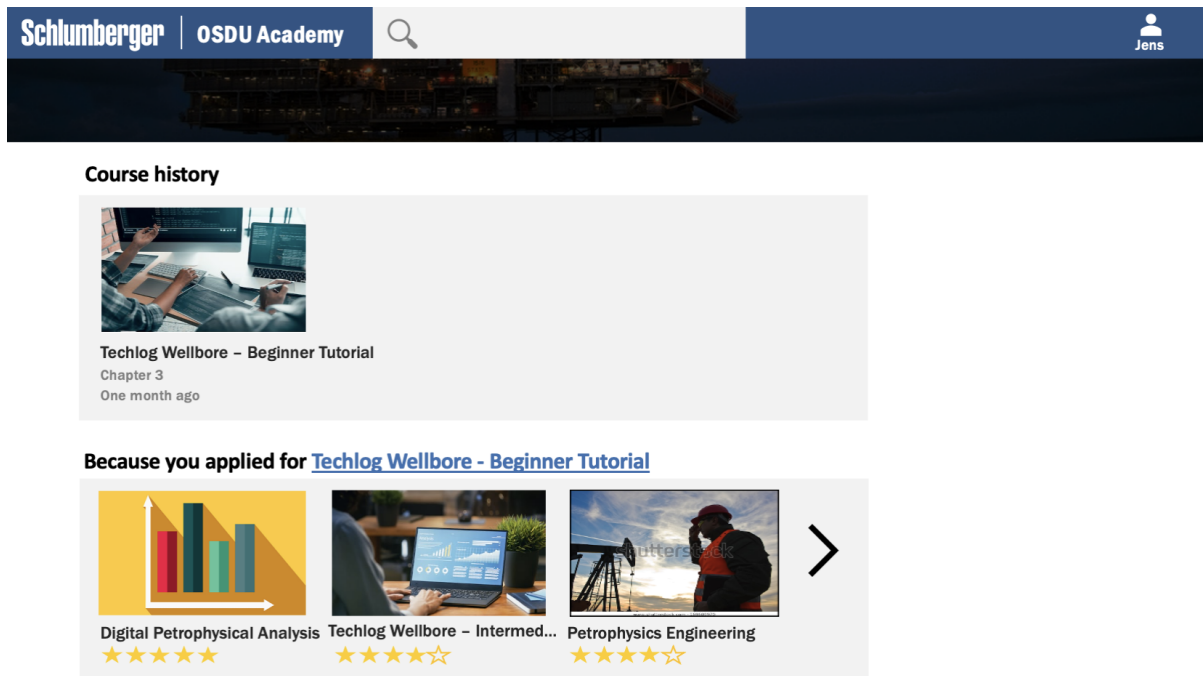


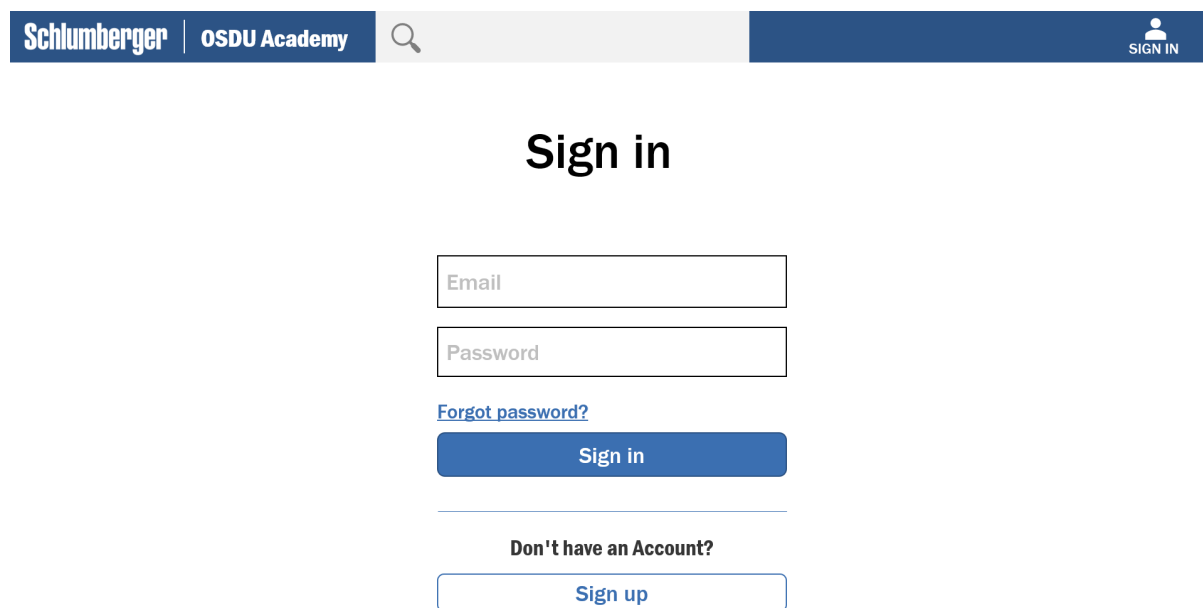
Figure 5.3: Overview of lower part of the home page - logged in

Lastly, there is a top bar on the home page with a navigation bar on it. This is common for every page of the prototype, with a slight difference in the design of, among other things, the course interface as described in chapter 5.1.5. This bar's position is fixed to the top of the screen and does not scroll with the rest of the page. On the far left side, there are the Schlumberger logo and the name of the application, OSDU Academy. There is a visible search field in the central part, and on the right side is a relatively small "sign-in" button that takes the user to the login and register forms. The search field and the "sign-in"-button will be used frequently and are designed accordingly.

5.1.2 Login and registration

Following up from the site's frontal page, pressing the top-right "sign-in"-button would lead to a plain page with sign-in and registration options. See figure 5.4 and figure 5.5 for reference. The layout features a standard set of input fields for email and password. The hyperlink with the text "Forgot password?" is underneath the two input fields in case the

user had forgotten their password. Under that is a blue "sign-in"-button that leads to the home page again with the user signed in. Besides, in order for the user to register an account, under a line and bold text saying "Don't have an Account?" is a white outline "sign up"-button. This design is heavily inspired by Schlumberger's official website [33], as the method of signing up is the same: the user has to access the sign-in section to find the registration page.

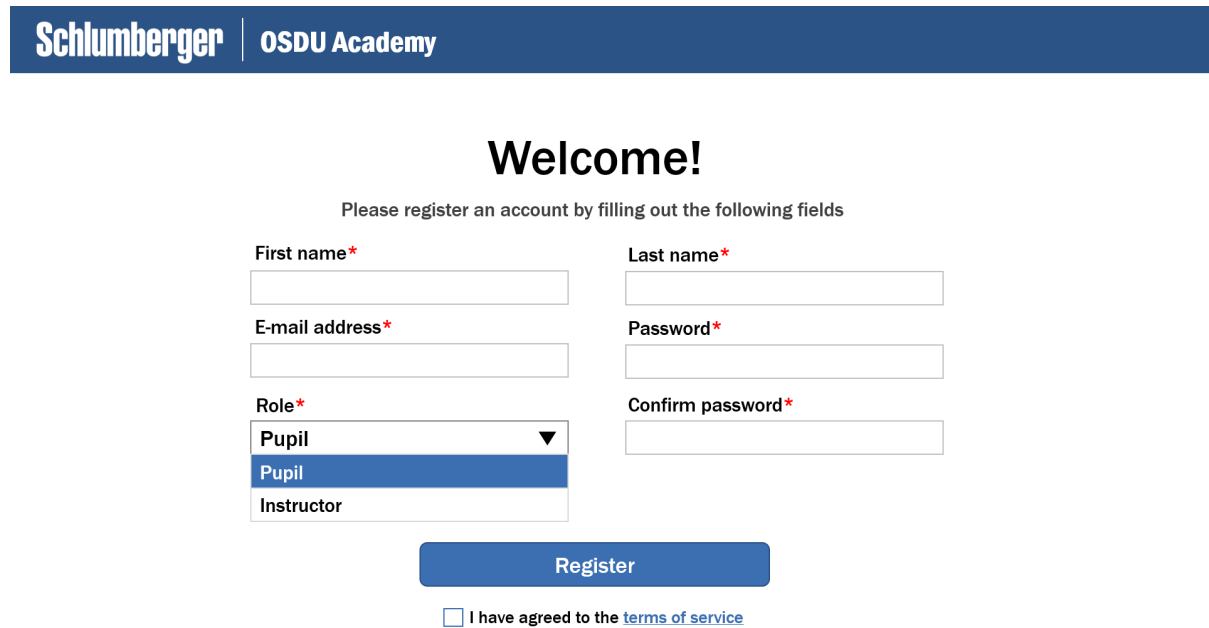


The image shows a web page header with the Schlumberger logo and OSDU Academy text on the left, a search icon in the center, and a SIGN IN button with a user icon on the right. Below the header, the page title "Sign in" is centered. The main content area contains two input fields: "Email" and "Password". Below the "Password" field is a blue link "Forgot password?". Underneath is a solid blue "Sign in" button. A horizontal line separates this section from the registration section below, which features the text "Don't have an Account?" and a white outline "Sign up" button.

Figure 5.4: The login page after pressing "sign in"

After finding the "sign up"-button and clicking it, the user should be presented with a registration page with similar traits as the log-in one. Some subtle changes have been made to the top menu bar this time: the search bar and "sign-in"-button is no longer present, and the bar itself is slightly taller. Similar to Schlumberger's website [33], there is some large text welcoming the user and a smaller subtext instructing the user on how to register. Input fields for standard user information like name, email, and password are presented in the same format as the log-in page. The difference here is that each field has a bold text label above it with a red asterisk at the end, indicating that the input data is required. A key feature alongside these is a drop-down list for the user's role on the platform. They can either be a student or a teacher on the platform, which in theory should change some further options. Furthermore, there is a large blue "Register"-button with a checkbox under it that must be checked since the user must agree to the platform's

terms of service.



The image shows a registration form for Schlumberger OSDU Academy. At the top, there is a dark blue header with the Schlumberger logo and 'OSDU Academy' text. Below the header, the word 'Welcome!' is centered in a large, bold font. Underneath, a subtitle reads 'Please register an account by filling out the following fields'. The form consists of several input fields: 'First name*' and 'Last name*' (text boxes), 'E-mail address*' (text box), 'Password*' (text box), and 'Confirm password*' (text box). A 'Role*' dropdown menu is open, showing 'Pupil' (selected), 'Pupil', and 'Instructor'. A blue 'Register' button is centered below the fields. At the bottom, there is a checkbox labeled 'I have agreed to the [terms of service](#)'.

Figure 5.5: The registration page after pressing "sign up"

5.1.3 Finding courses

The user can find a course by scrolling down on the home page or by using the search alternative at the top navigation bar shown in figure 5.6. The search page is quite standard and modern. The largest part of it is reserved for the search results. Each of those results has a clear heading, a short description, and a rating scale from one to five stars. Furthermore, one can filter the results by using the menu on the left side. There is a drop-down list for general filters, and below are more specific choices, like the domain and ratings. Additionally, there is a large text between the top bar and the page's body, detailing the number of results.

The screenshot shows a search results page for the Schlumberger OSDU Academy. The header includes the Schlumberger logo, OSDU Academy, a search bar with 'techlog tutorial', and a user profile icon for 'Jens'. The main heading is 'Your search gave 3 results'. On the left, there are filters for 'Most Relevant' (selected), 'Domain' (Geophysics, Business, Analysis, Drilling, Software), and 'Rating' (4+ stars, 3+ stars, 2+ stars). The search results are displayed as three cards:

- Techlog Wellbore – Beginner Tutorial**: Integration of all wellbore-centric data types into multidiscipline workflows. Rating: 5 stars.
- Fundamentals of Petroleum Geology**: It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. Rating: 5 stars.
- Techlog Wellbore – Intermediate Course**: Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit. Rating: 5 stars.

Figure 5.6: Searching for a course

5.1.4 Applying for a course

As a result of pressing the course buttons on either the search page or anywhere else featuring such buttons, the user will be presented with a page including information about the course in question, as seen in figure 5.7. A large part of the page is covered by the course description and a large banner at the top with an image to the right. The banner is the most eye-catching part, as the title is there along with a small description subtext. Important elements that make the course more attractive are a bright green "Apply"-button and up to five stars on the left next to a "Read reviews" hyperlink. At the bottom-right quadrant of the page, there is a course structure as well, featuring all the chapters.

Schlumberger | OSDU Academy

Search: multiphase meters

User: Jens

Introduction to Multiphase Flow Meters

Learn about the principles and applications of Multiphase Flow Meters

★★★★★ Read reviews

Apply

Description

A multiphase flow meter is a device used to measure the individual phase flow rates of constituent phases in a given flow (for example in oil and gas industry) where oil, water and gas mixtures are initially co-mingled together during the oil production processes.

The oil and gas industry began to be interested in developing MPFMs in the early 1980s, as measurement technology improved, and wellhead separators were costly. Depleting oil reserves, (More water and gas in the produced oil) along with smaller, deeper wells with higher water contents, saw the advent of increasingly frequent occurrences of multiphase flow where the single-phase meters were unable to provide accurate answers. After a lengthy gestation period, MPFMs capable of performing the required measurements became commercially available. Much of the early research was done at the Christian Michelsen research center in Bergen, Norway, and this work spawned a number of spin off companies in Norway leading to the Roxar / Emerson, Schlumberger, Framo, and MPM meters.

Course structure

- Introduction
- What are Multiphase Flow Meters
- Multiphase Flow Meter Principles
- Applications of Multiphase Flow Meters
- Test

Figure 5.7: Course page when applying for the first time

Figure 5.7 is what it would look like for a user who has yet to apply for that particular course. If they already applied for a certain course and are viewing its front page, the difference would be a blue "Continue"-button instead of a green one for applying to the course as shown in figure 5.8.

Schlumberger | OSDU Academy

Search: Techlog Wellbore

User: Jens

Techlog Wellbore – Beginner Tutorial

Integration of all wellbore-centric data types into multidiscipline workflows.

★★★★★ Read reviews

Continue

Description

Data acquired in wellbores is intricate, varied, and represents a significant investment. It is imperative to capitalize on that investment, whether you are working on your first deepwater exploration well or delivering a comprehensive drilling program in a shale gas play. With the Techlog platform, you can bring all of your wellbore data into one highly intuitive application to carry out analyses. In addition to offering advanced modules for domain experts, the Techlog platform provides a solid foundation for generalist users to review and analyze the data. It supports complex workflows across multiple disciplines.

The Techlog platform dynamically incorporates the latest data. It is streamed directly from the well site for real-time processing and instantaneous analysis as the well is drilled, aiding decision making during operations.

Course structure

- Introduction
- Lithology Management
- Production Logging
- Palette Editing Tool
- Test

Figure 5.8: Course page when already applied for

5.1.5 Course design

Each course follows a general format where only a certain area will vary based on what content is shown in each chapter. Up until now, the fixed navigation bar at the top has seen no changes. This time, the bar has two new hyperlinks: giving feedback and tracking course progress. There is a bar with two directional buttons that navigate to the previous and next chapters on the very bottom of the course content. In this bar is the name and numbering of the viewed chapter. See figure 5.9.

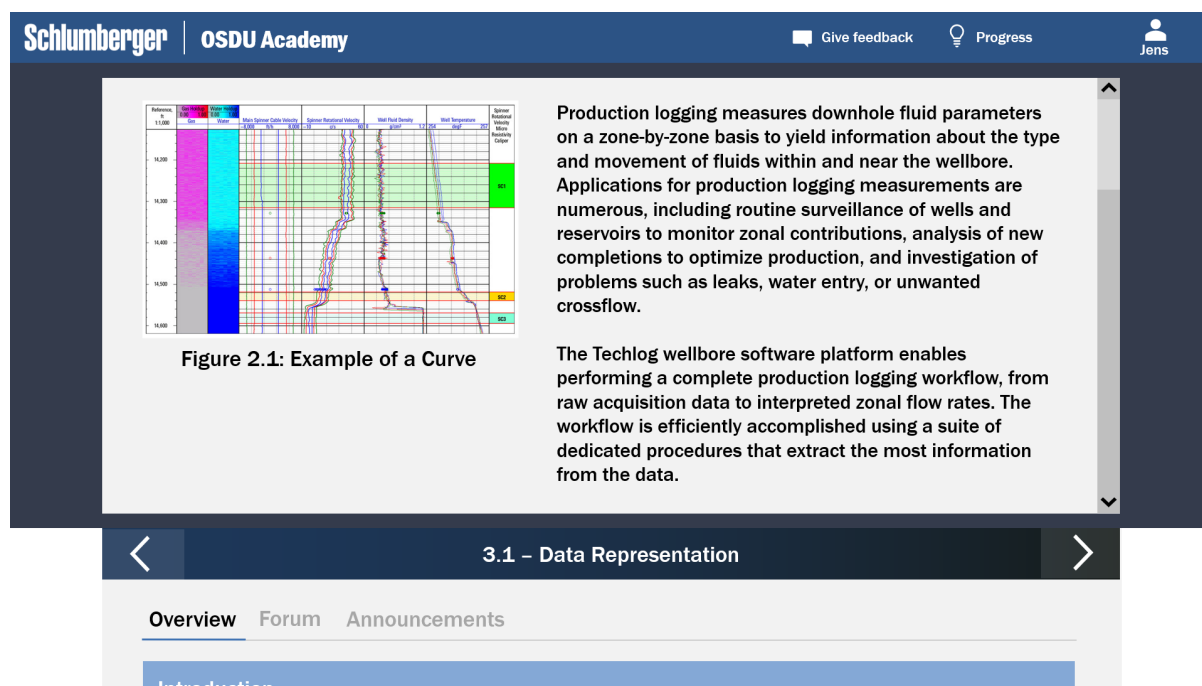


Figure 5.9: The course interface showing a document

Pressing the right chevron leads to the next chapter that, for example, would show a video, as seen in figure 5.10. Here, the chapter numbering and title changes chronologically.

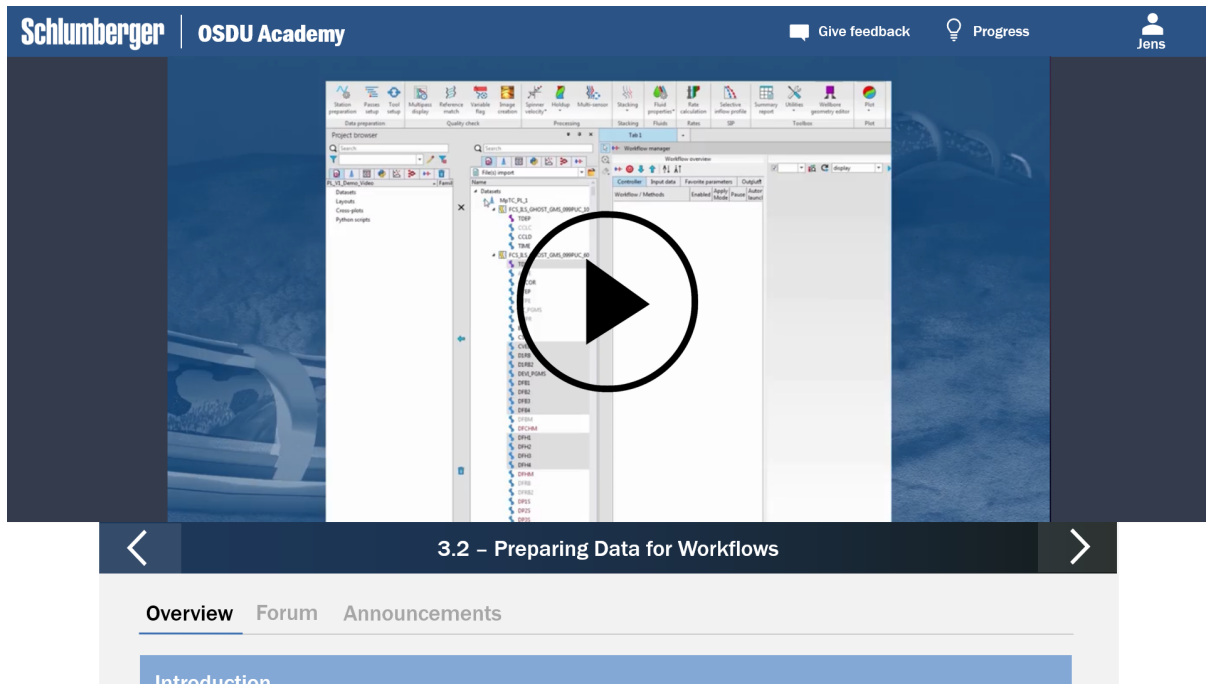


Figure 5.10: The course interface showing a playable video

Last but not least important, at the very bottom of the page, an entire menu exists, which should help the user navigate through the course. The user would have to scroll to it to see the rest of it, hence why it is cut off at the bottom in figure 5.10. Figure 5.11 illustrates the rest of this somewhat apparent area. Each chapter can be collapsed and opened in order to view its subsections and navigate into them. Additionally, the user can press the above title bars to reveal a forum or a page where the instructor has published their announcements for the course.

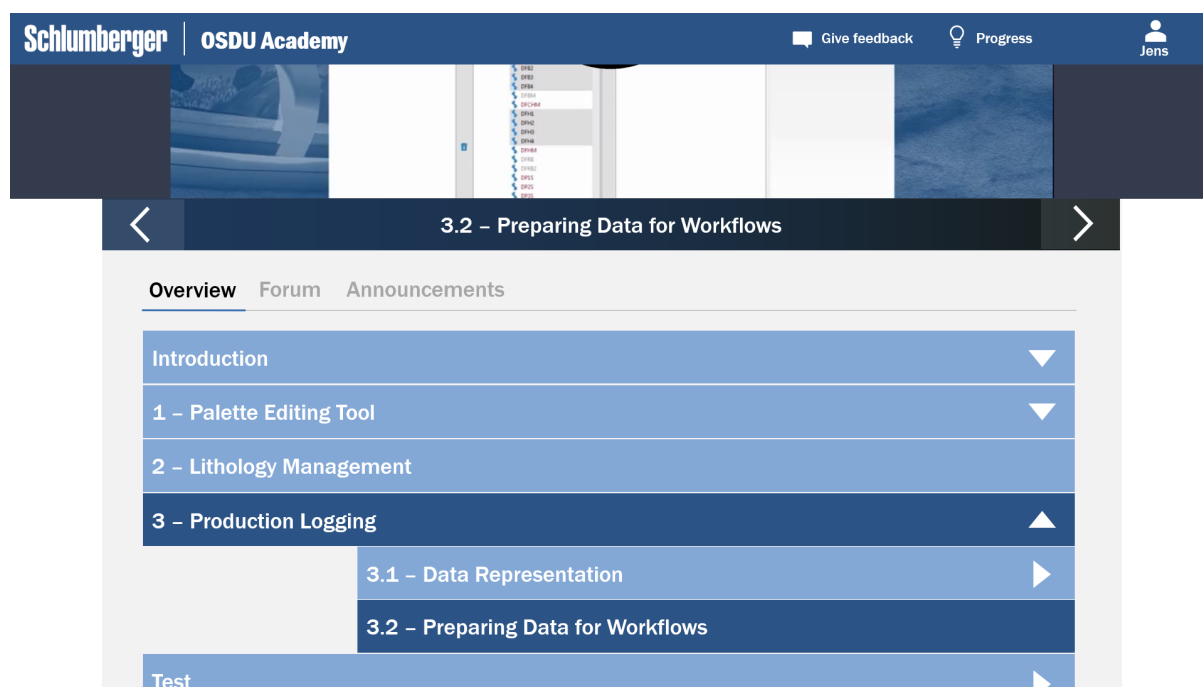


Figure 5.11: Further down the course page

5.1.6 Certification

The test and certification are the two last pages of Prototype 1. See the figures 5.12, 5.13 and 5.14. The top bar at this point does not have a search field. Under it, one can find another banner containing a subtle arrow that takes the user back to the course page. Within this large banner is also a large heading with the text "Certification."

Moreover, the page consists of a field with information about the test. For instance, the user gets to know the duration and the form of the test, among other things. Below, there is a test in the mentioned format, in which case is a quiz featuring multiple-choice and true or false questions. At the end of the quiz, there is a bright green submit button which can be clicked to end the test. The user can then check answers and download a certificate if the result is satisfying.

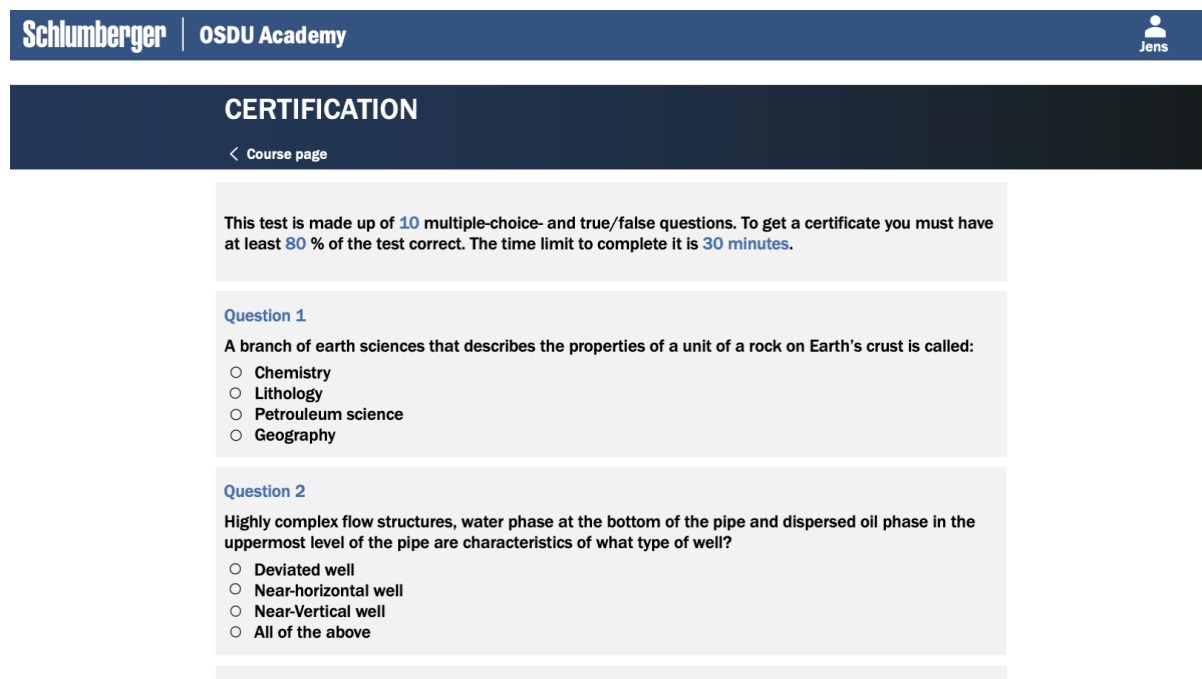


Figure 5.12: The test interface, part 1

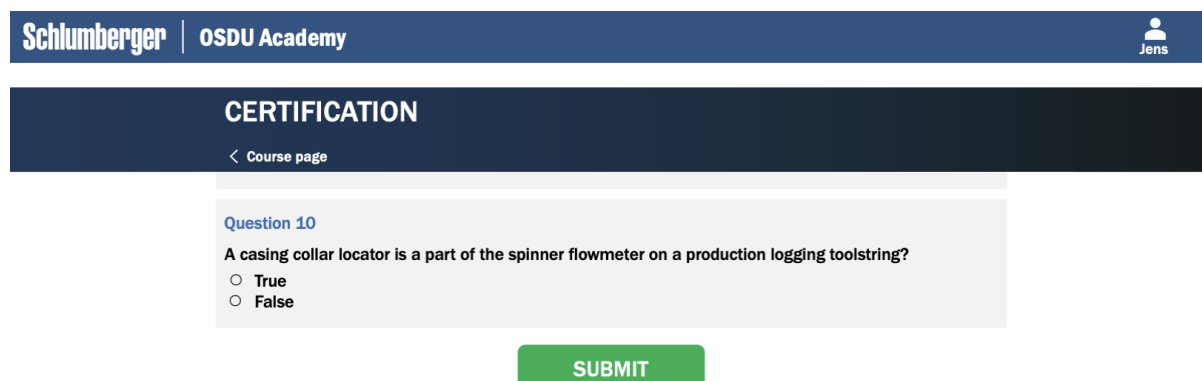


Figure 5.13: The test interface, part 2

Lastly, the certificate is a simple golden brown rectangle confirming that the test is passed and providing information about the score and a link to check the test results in detail. There is a green button at the very bottom of the rectangle that allows the user to download the certificate. When the test is passed and the banner is shown, the user can

rate the course under if they choose to.

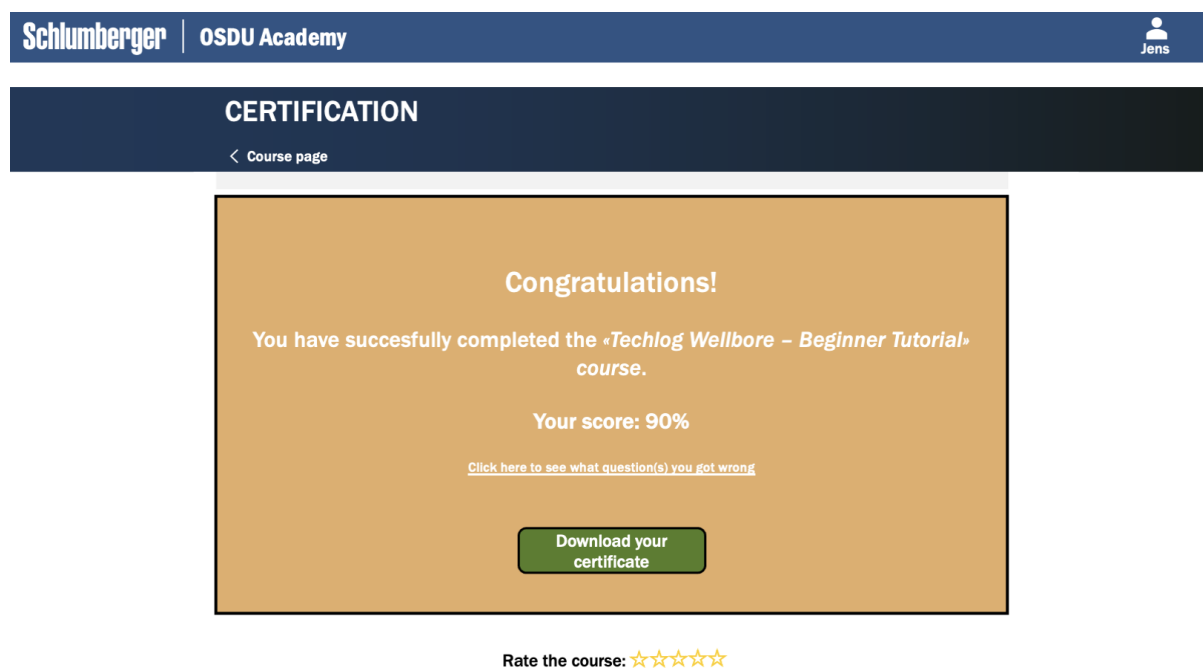


Figure 5.14: Certificate

5.2 Prototype 2

This prototype is a high-fidelity prototype, meaning that it is fully functional. It was developed using ASP.NET Core and ReactJS. The design process and development took into account the different elements of user-centered design and feedback received from the actual users. This chapter will be a little bit different, as this prototype has more to cover. First, the new interface design and its differences to Prototype 1 will be covered, then there will be more information on the technical aspects and architectural design choices.

5.2.1 Home page

The first image shows the re-designed home page in figure 5.15. It still consists of three parts: the top navigation menu, the banner and the content. There have been changes in all of those. The banner is more subtle than in the first prototype but contains more information about the application.

Below the banner, there is a carousel that features trending courses and a category navigation bar. As the application is a minimum viable product, the navigation bar has a

more significant role, namely to be used as a filter for courses in different domains. The various courses are represented with an image, the name of the course, a short description, and a difficulty tag. Further, the stars have been removed as the rating system is no longer a part of the product. This is due to the fact that a minimal viable product does not require this feature to be implemented. One important thing to note is the "Trending Courses" title. In contrast to the name, it does not feature actual trending courses based on statistics. Instead, they are simply pulled from the database by their domain name in order to simulate displaying popular courses.

The top bar on the home page has also been changed. It is no longer fixed to the top of the screen and is standard for all pages. The logo and icons have been more centered and horizontally aligned with the main page content. Also, there is an additional icon targeted towards the teaching community of OSDU Academy. Read more about this in chapter 5.2.5. The "sign-in" button is aesthetically unchanged, but it has new functionality when the user is logged in; namely, it re-directs to the profile page. The most significant difference is that the search bar has been taken away due to following minimal viable product standards.

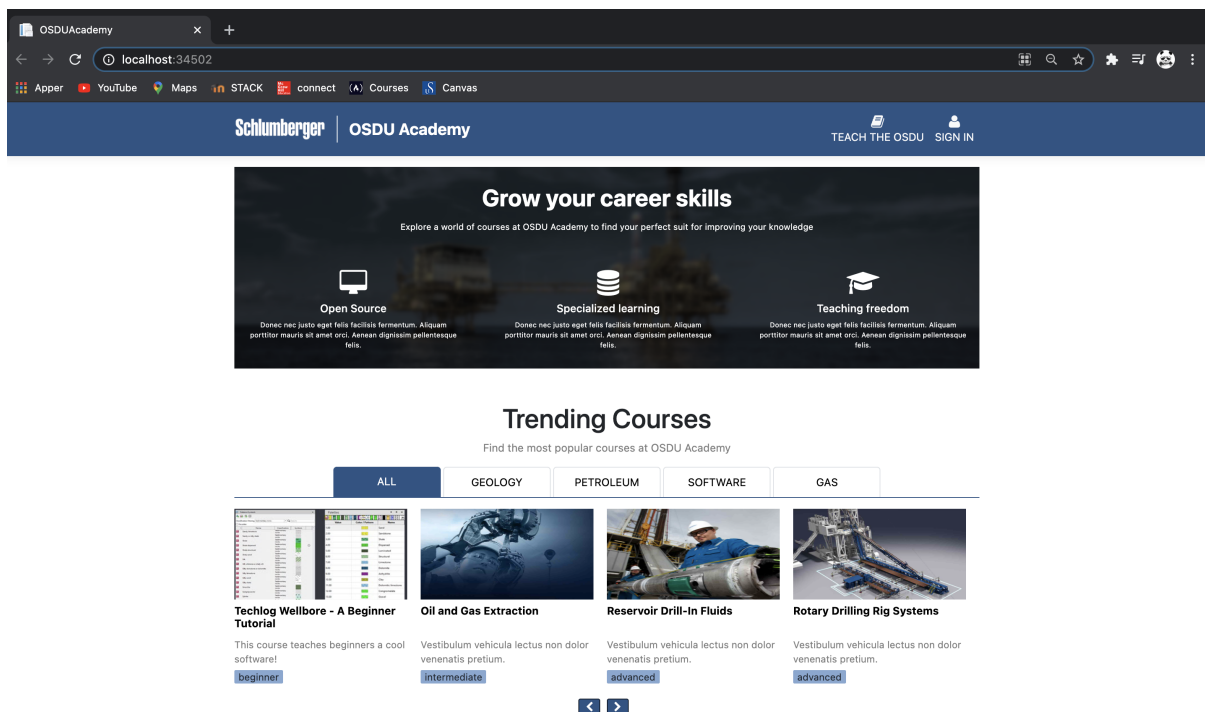


Figure 5.15: Home page when the user is not logged in

The testers have been critical towards the home page when the user is logged in, see figure 5.16. As a result of the feedback, the page has been re-designed accordingly. It includes the same features as the home page when the user is not logged in, with slight changes in the banner's design and the addition of a new section. The banner is slightly smaller and with a different message. The "Course history" from Prototype 1 has been replaced with a more visible and direct "Catch up on your learning" segment. It contains all the courses the user has applied for and has not finished.

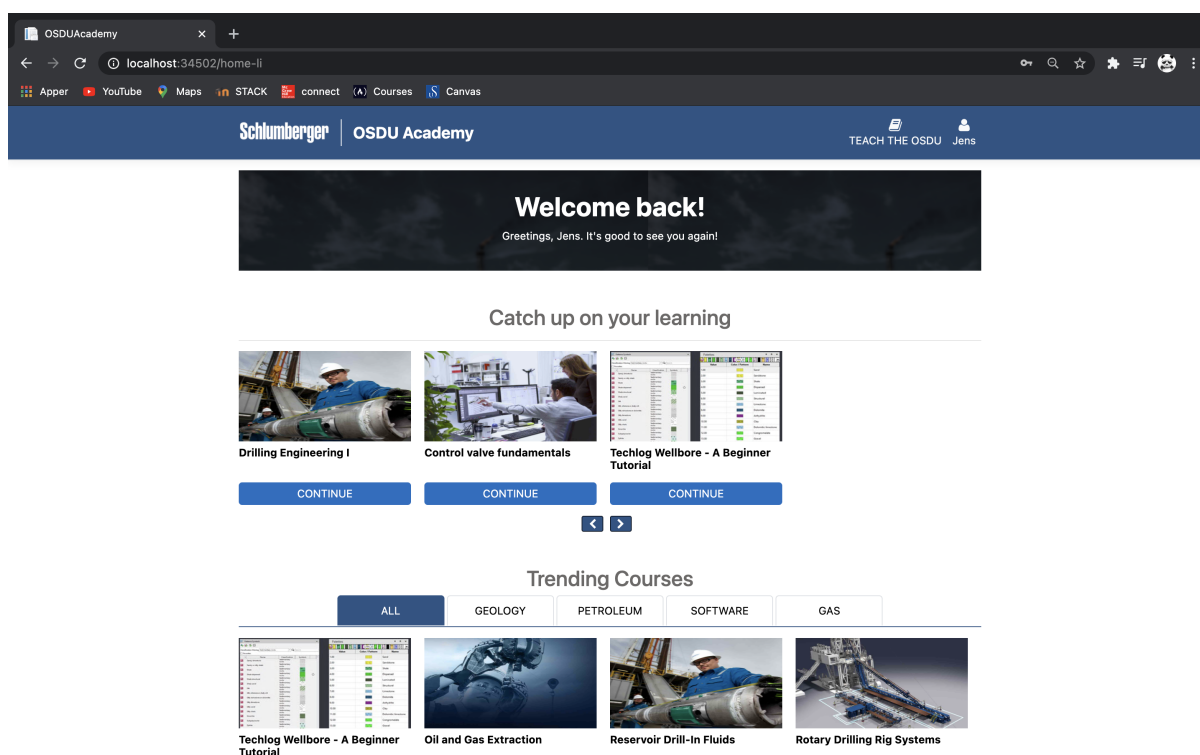


Figure 5.16: Home page when the user is logged in

5.2.2 Profile page

In contrast to the previous prototype, there is a profile page in this one. It includes the user information on the left side and two sections containing enrolled and certified courses. Via the profile page, it is possible to log out. The main functionality linked to this page is that it lets the user have an overview of their account information and taken courses.

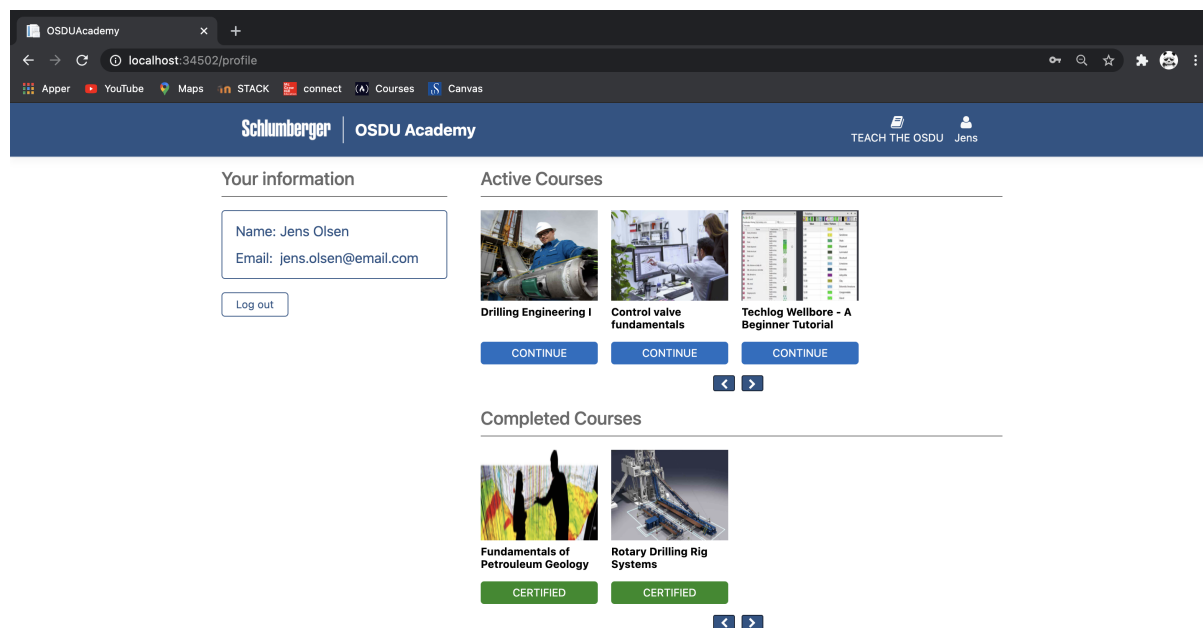


Figure 5.17: Profile page of a user named Jens

5.2.3 Course front page

One can see one of the most significant differences in the design on the course front page, as shown in figure . The relocated course description is still a crucial part of the page and is positioned under the course chapter overview. The overview has been taken from the right side to the more central location, and the design is more minimalistic. The banner with an image to the right has been slightly changed, as the information provided is not the same. The star rating and the "read reviews" option have been replaced with more descriptive data: a difficulty tag, and the approximate amount of time it takes to complete the course. Under the course image, there is a list of prerequisites one should take into consideration before taking the course. The apply button is no longer placed in the banner but under the image and is blue.

When a user applies for the course, the "Apply"-button gets deactivated, and an "Enter"-button, is shown next to it.

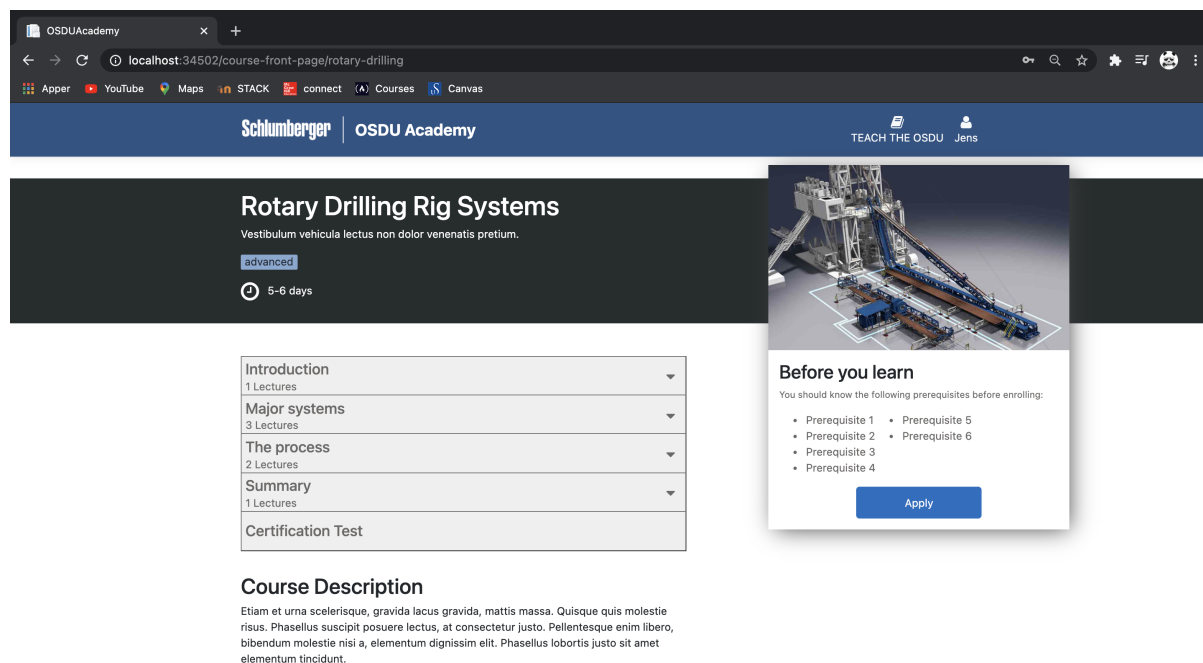


Figure 5.18: Course front page

5.2.4 Course interface

The course interface is completely redesigned. The top navigation bar contains only two buttons. The first one is the left-located home button which returns the user to the home page and the second one is an overview button. By clicking it, an overview with all the course sections and lectures is shown. The overview menu corresponds to the one on the course front page. The central part of the page is reserved for the course content. The page has a fixed size, and that is why the user can scroll the content section. There are chevrons on both sides of the segment that can take the user to the previous or next chapter. The user can skip to a specific chapter by selecting it in the overview menu. All other features that were not mentioned and were a part of the first prototype have been taken away due to minimal viable product standards. The new design of the course interface is shown in figure 5.19.

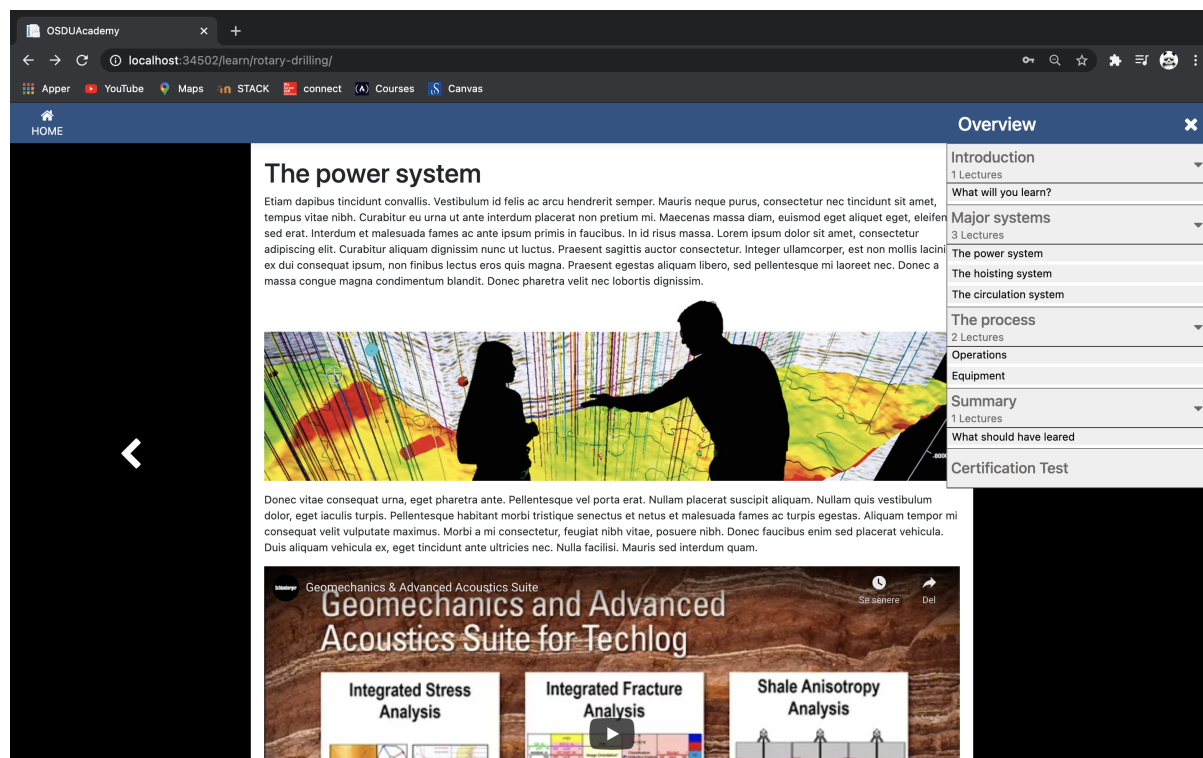


Figure 5.19: Course interface

5.2.5 Teach OSDU

Even though the focus is on the student, there is still a way to create courses for the OSDU Academy. Namely, by choosing to click on the "TEACH THE OSDU" button in the top menu, one gets redirected to an instruction page. Here lays the information on making a course, contact information, and a template for the course. Figure 5.20 represents this. The teacher can download the template by clicking on the hyperlink under the "Template" section.

The premise behind this way of contributing to the platform's content is to keep up with the standards of a minimum-viable product. Our focus was directed towards the consumers of courses in OSDU Academy, so we decided to keep content creation as basic as possible. The idea is that a teacher must follow the instructions inside the template for creating a course and send the content to the administration of OSDU Academy. Then, the administrators will be responsible for including the teacher's content in the platform.

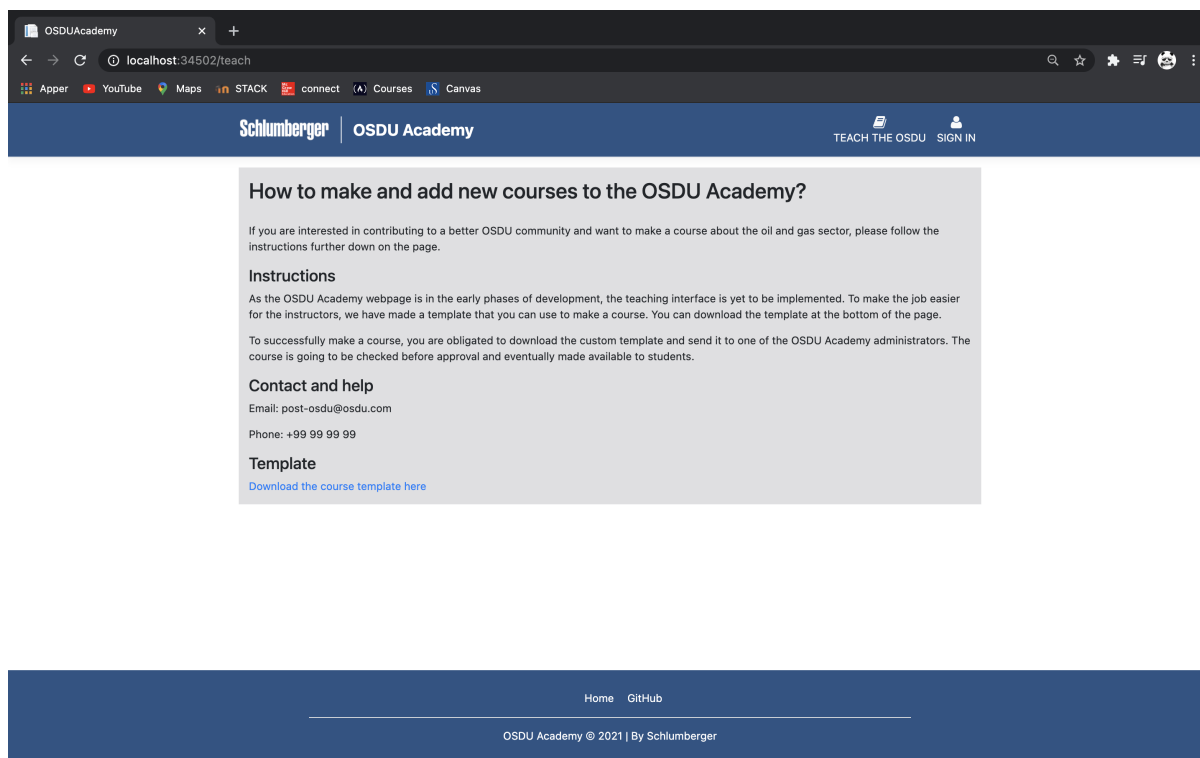


Figure 5.20: Teach OSDU

5.2.6 Pre-certification page

The overall certification process has not been changed drastically, see figure 5.21. Before taking the test itself, the option to take it immediately or later exists. This page is an addition in comparison with the previous prototype. Under the top navigation bar, one can find two banners, one containing a subtle way to go back to the course and another with information about the certification quiz. The user gets to know the form of the test and how much it takes to pass it. Below the information banner, a green "Take the test" button exists to access the test.

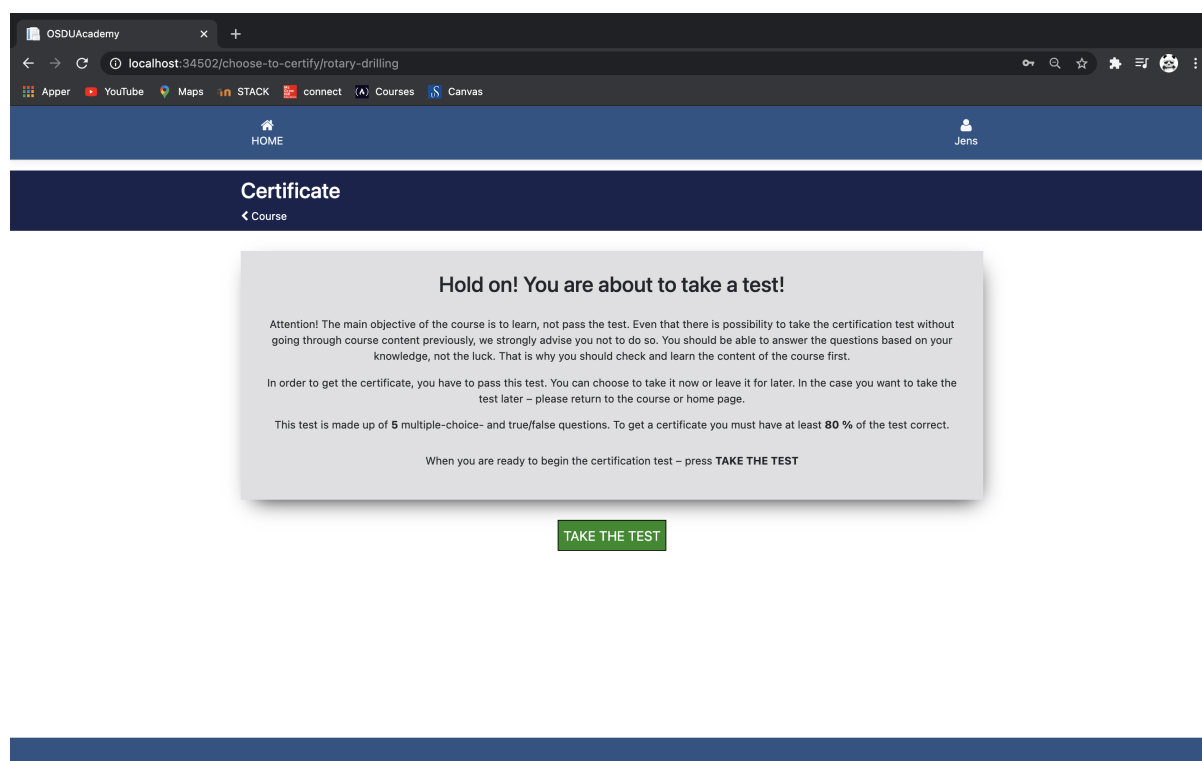


Figure 5.21: Certification choice

5.2.7 Functional architecture

This section will focus on the functional architecture's implementation. OSDU Academy was implemented with the many design patterns covered in chapter 4.5.5.

5.2.7.1 Services and components

The first part of OSDU Academy implemented was the front-end using React JSX on the web. Since the front-end is a single page application in React, a declarative and component-oriented pattern was employed to create the varying sections and widgets in the interface. Every widget, such as buttons, drop-down lists and even a full page is considered a component when using React.

Code snippet 1 shows an example component's structure. It consists of a JSX class that inherits the base class for React components and overrides essential methods such as "render" to function correctly. Code like this will easily be re-usable across the application since JSX allows the programmer to include the given component in any "render" method. This is done by referencing it using the class name wrapped in a HTML-like tag.

Code snippet 1: React component example

```
export class ChapterDrop extends Component {
  state = {dropBar: false};

  dropBar() {
    this.setState({
      dropBar: !this.state.dropBar,
    });
  }

  render () {
    return (
      <div className="course-drop">
        <button
          className="drop-button slightly-dim"
          onClick={this.dropBar.bind(this)}>
          ...

```

One last important pattern used for the front-end was the service layer model. Read more about it in chapter 4.5.5. In OSDU Academy, it was done by creating read-only objects containing asynchronous functions that abstracted the usage of Asynchronous JavaScript And XML, also known as AJAX, to communicate with the back-end. These "service objects" were treated like name-spaces for these functions corresponding to their respective usage, yet they also contained other helper functions for the front-end. Code snippet 2 shows an example of a service layer object that handles the retrieval of basic course information.

Code snippet 2: Example of a service layer object in JavaScript (ECMAScript 6)

```
const CourseService = {
  fetchFrontPageCourseData(courseRoute, callback) {
    fetch('course/' + courseRoute)
      .then(response => response.json())
      .then(callback);
  },

```

...

5.2.7.2 Controllers and access routes

Following the MVC pattern that was described in chapter 4.5.5, we were able to robustly communicate the platform's back-end with its front-end. The plan was to combine this pattern with the service layer pattern that we used to handle front-end connections, both to organize connection-based code in a uniform fashion and for compatibility regarding other client application types such as mobile. Controllers, similar to service layer objects on the front-end, were separate handlers for communication methods configured to a specific usage. Controllers can handle database queries, authentication, routing and sending data to the client.

In code snippet 3, one can find an example controller's structure with a routing method. It contains functionality for retrieving basic course data from the database and sends it to the client application.

Code snippet 3: C# course controller code example

```
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using System.Linq;
using MongoDB.Driver;

namespace OSDUAcademy.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class CourseController : ControllerBase
    {
        [Route("trending")]
        public IEnumerable<Course> GetTrending()
        {
            ...
        }
    }
}
```

5.2.7.3 Sequence diagram

Code and application structure is important in a software project, but designing the functional architecture does not end there. Developers in the software industry utilize diagrams using the Unified Modeling Language, or UML for short. This tool is used to visually model both the logical and structural aspects of a program in many different fashions, such as class diagrams, relational diagrams and sequence diagrams. Without cluttering this chapter with an excessive information, figure 5.22 will show a simple sequence diagram of the login process, among other things.

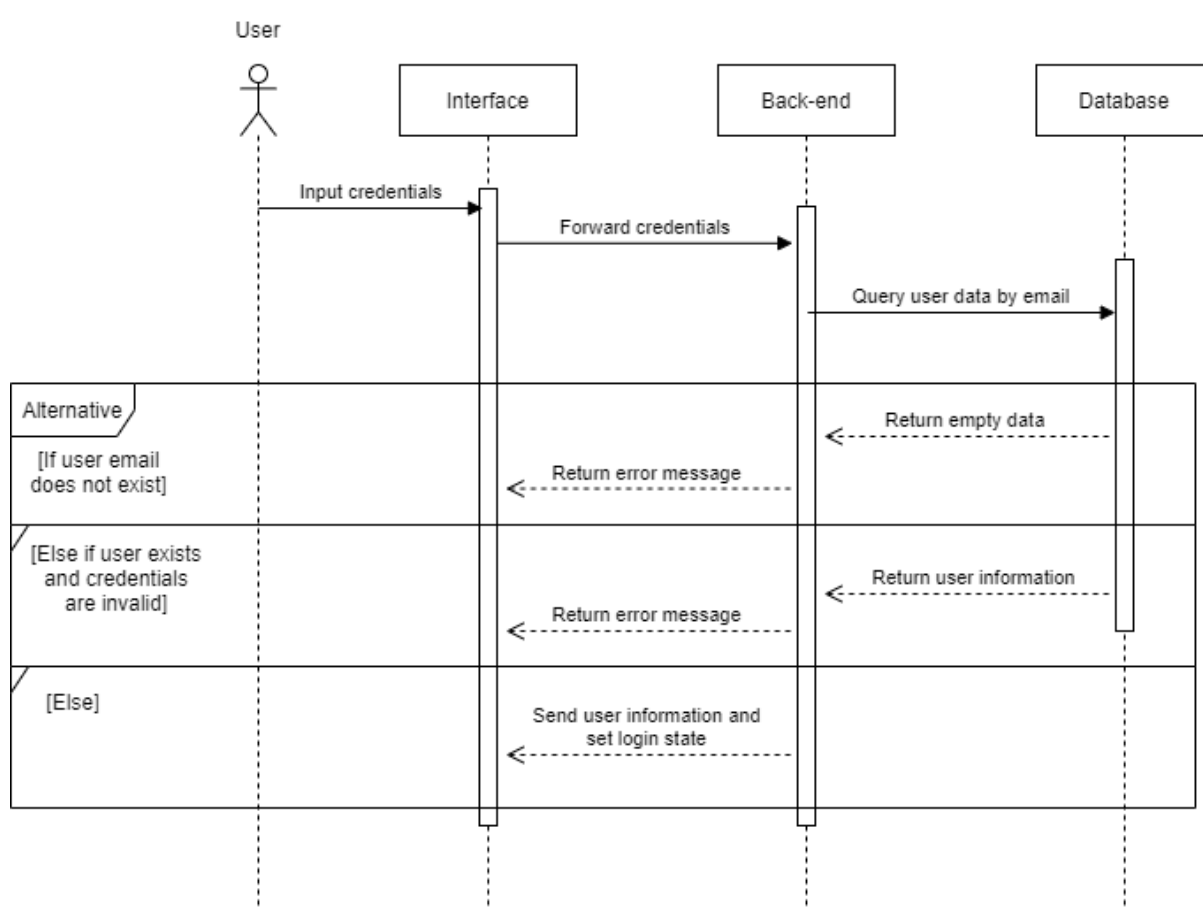


Figure 5.22: Sequence diagram of the login process

This three-component structure is sequentially present in all of OSDU Academy's functional architectural areas. It follows the client-server model with a database as its main source of information.

5.2.8 Database design

Before the database was implemented, we created a diagram that represented its layout. This process helped uncover and solve other long-term issues, such as the course data structure. Since we used MongoDB as the database management system for OSDU Academy, there were no strict constraints that a model for a relational database would require. Issues like normalization were not present in the making of a non-relational database, like MongoDB. In figure 5.23, there is a model for the database used in OSDU Academy. There is some custom syntax that needs to be covered with this diagram. Firstly, it is essential to note that MongoDB supports embedded documents, which are essentially nested entities. These are denoted by an arrow on top of the box in the diagram. Secondly, as supported by most database management systems, foreign keys are denoted using an arrow pointing to the "id"-field of an entity box. This way, it is possible to see the distinction between a referenced entity and a nested one. Lastly, each field possessed by an entity has its data type covered in parentheses. The data types are represented with the same names as they would in MongoDB.

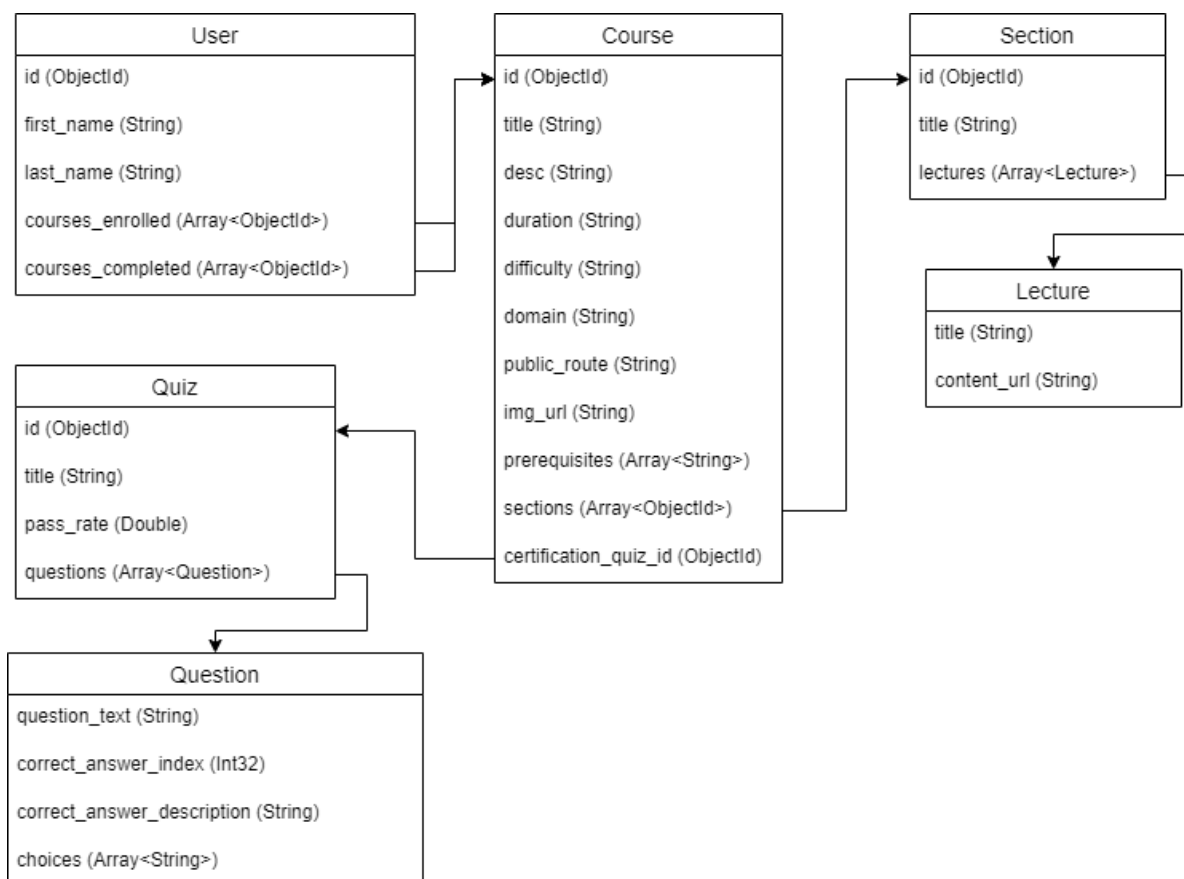


Figure 5.23: Diagram of MongoDB database

This structure was not decided without reason. As diagram 5.23 was being created, several issues popped up along the way, which we needed to account for. The first one was query performance. We expected that basic information about courses, such as title, description, and duration, would frequently be used on the front-end. This means that the basic course information is stored as a separate entity from its actual content, and the content of a course is a reference from a different table. This reduces the amount of data fetched when querying for, for instance, popular courses since the essential data is already structured accordingly in the database. We applied this construct for all entities in the database.

5.2.8.1 Course content management

Managing the course content was not as simple as storing it in MongoDB. It required that we had to organize it separately from the MongoDB-based database and design our own method since the amount of such content is too excessive. As seen in figure 5.23, lectures, which are a part of a course section, hold a "content_url" attribute. This is a path to a file which contains the structure and format of a lecture, written in Extensible Markup Language, also known as XML. This file is stored on the main disk where the operating system running the back-end is contained. Using XML opened many doors for current needs and potentially future development of OSDU Academy. XML is a markup language similar to the one we used for writing the front-end user interface, but as the name implies, it is extensible and can be customized to any liking.

An example of a lecture written in XML that contains a title, paragraph and an embedded YouTube video can be seen in code snippet 4.

Code snippet 4: Course content example in XML format

```
<Content>
  <Title>Motory motion valves</Title>
  <Paragraph>
    Integer vitae ultricies sem. Donec pharetra sollicitudin elit vel
      mattis. Morbi vehicula nec urna id dignissim. Cras velit massa,
      facilisis at feugiat a, ...
  </Paragraph>
  <YouTube link="https://www.youtube.com/embed/..." />
</Content>
```

Since the front-end of OSDU Academy is written in React JSX and is quite similar to XML, we decided to parse the XML code directly with a React library called "jsx-parser" and keep the XML code as JSX-friendly as possible. It saved a lot of time from configuring a specific XML parser, however if OSDU Academy is to ever be ported to other platforms, then specific XML parsers might be required for some systems or environments.

5.2.9 Security baseline

Since the prototype is meant to be a minimum viable product, security needs to be taken into account in the corresponding limited constraints. As for storing user data, the application only features logging in to predefined users in the database, and registration is not possible. The reason for this and other feature compromises will be discussed further in chapter 5.3. With that in mind, the security of Prototype 2 covers the limited functionality yet is extensible enough for future work.

5.2.9.1 User authentication and permissions

To achieve the best possible form for security, the most powerful thing one can do is limit access to certain data. This concept is not complete with just limiting data access permissions and limiting what is sent to the end-user in certain cases. For instance, users who have not logged into the system are granted basic information about a course, either on the home page with simple course cards or when accessing more information on the front page. In this prototype, users who have logged in and have applied to a course are the only ones granted access to the course's main content. This incomplete model makes it easy for the developers to account for future permission management, since such a logical structure is taken into account from the beginning.

Using ASP.NET Core, there exists a `C#` class attribute that tags a controller class with the requirement to be authorized, as seen in code snippet 5.

Code snippet 5: `C#` authorization class attribute attached to controller

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
```

```
namespace OSDUAcademy.Controllers
{
    [Authorize] // <-- Require authorization for this controller
    [ApiController]
    [Route("[controller]")]
    public class LearnController : ControllerBase
    {
        ...
    }
}
```

This authorization is handled by a middleware configured in the core application code. The API used in this security method uses something called JSON Web Tokens, or JWT. JWTs are encrypted strings of text containing information about its encryption algorithm, verification signature, and a payload of data based on its usage. In this case, the payload consists of the user's email and token expiration date. When a user has successfully logged in, this type of token is randomly generated and stored at the user client. With this token, the user does not need to log back every time they access the page since the token will be sent to the server each time they attempt it. If the token is valid, then the user is authenticated and is granted the corresponding data access. Before wrapping up, it should be important to cover that each token's verification signature is also encrypted using a secret key stored in the server-side application. This key is also randomly generated and should be stored safely outside of the application source code.

To understand how a JWT works, it must be visualized accordingly. Using the official client-side tool for token decoding at jwt.io, one can easily see how such a token structured. In figure 5.24 each section in the token separated by a dot is color-coded and decoded into the corresponding information.

The screenshot shows the jwt.io interface. On the left, a long JWT token is pasted into a text area. On the right, the token is decoded into three sections:

- HEADER: ALGORITHM & TOKEN TYPE:**

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```
- PAYLOAD: DATA:**

```
{
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name": "jens.olsen@email.com",
  "exp": 1618221416,
  "iss": "http://localhost:5000",
  "aud": "http://localhost:5000"
}
```
- VERIFY SIGNATURE:**

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  This key is not meant
)  secret base64 encoded
```

Figure 5.24: Example of a decoded token using jwt.io

5.2.9.2 CSRF-secured yet vulnerable to XSS

As discussed in the previous paragraph, JWT is not only used for controlling user permissions, since this can already be done by a simple login method done each time authentication is required. However, this process is very tedious and leaves the platform quite unusable. Having a web token, stored alongside the user, handle authentication automatically makes the user experience smoother. Nevertheless, this comes at a cost. Attackers from external sources can take advantage of the fact that the token is stored locally and perform a cross-site scripting attack, also known as an XSS attack. This type of attack is done through the scripting language in the web browser and any user who accesses a malicious website will have their local storage exposed. The authentication token our product uses is stored in that local storage too and this is where the security becomes weaker. Local storage should not be used for storing sensitive information, only settings and other convenience-related information shall remain alongside the user.

Nevertheless, this method for automated authentication protects against cross-site request forgery attacks, or CSRF attacks for short. As the name implies, attackers are able to forge requests to a website that a user would otherwise not be willing to do, through an

external source. The external source is key here, since it can vary how each request is forged by the attacker. Most commonly, requests are forged through fake websites. Fake websites are easy to create since front-end source code is always public to the attacker due to the nature of web browsers, therefore one can swiftly forge a fake and similar website to the desired one. By using constantly changing and generated web tokens to authenticate the original user, in theory, it would be impossible for attackers to succeed with cross-site request forgery. In other words, the validity of the real user will always be fresh using this system.

5.2.9.3 Password encryption

Passwords in OSDU Academy should be stored somewhere along with corresponding user data. However, this process is not as simple as keeping them in the database. In the event of a data breach and each user's raw password is stored in the database, attackers can easily access their passwords that are potentially used in the users' accounts for other services. To prevent this, passwords need to be encrypted in the database as well. This is typically done by hashing⁵ the password using a salt⁶. Both the salt and hash are stored in the database, and each salt is unique. The hash is created using the user's raw password input and the salt, which is randomly generated from the beginning. Now, the password is hidden, and all that is needed when a user needs to log in is to use the password they wrote, hash it with the salt, and compare it to the hash stored in the database as seen in figure 5.25.

```
  _id: ObjectId("60534a3d1a5ac19fe109ddbc")
  first_name: "Jens"
  last_name: "Olsen"
  > courses_applied: Array
  email: "jens.olsen@email.com"
  password: "aWNMaNTU8T2y/Mpnb0Tqof5C8gC3zxIIcx/SVQY3dSY="
  > courses_completed: Array
  salt: "ophbzzpAGKvdajwyXIIJbw=="
```

Figure 5.25: Example of a user's hash and salt stored in the database (screenshot from MongoDB Compass)

⁵Mathematical algorithm that maps data to a bit-array of a fixed size

⁶Random data that is used as an additional input to a one-way function that hashes data

5.2.9.4 SSL Certificate

Without something called an SSL certificate, all data exchanged between OSDU Academy and the end user is completely visible to any attacker. This means the application is commonly vulnerable to what is known as a "man in the middle"-attack. In the event that an attacker is able to modify the communications between two parts, it is possible for them to eavesdrop the communication, hence the name "man in the middle". The SSL certificate is essentially the key to securing a connection by encrypting everything that is exchanged. SSL does not only protect against eavesdroppers, but also serves the purpose as a protection against data breaches, phishing scams and many more attacks. Nevertheless, an SSL certificate is never enough. As covered earlier, there are other security concerns we accounted for and there are many more to come later in this chapter.

To prove that OSDU Academy has a valid SSL certificate, it is enough to look at the top-left corner of the web browser when accessing the OSDU Academy website. In figure 5.26, there should be a padlock icon present that, when clicked, reveals that the connection is secure and that there is a valid certificate.

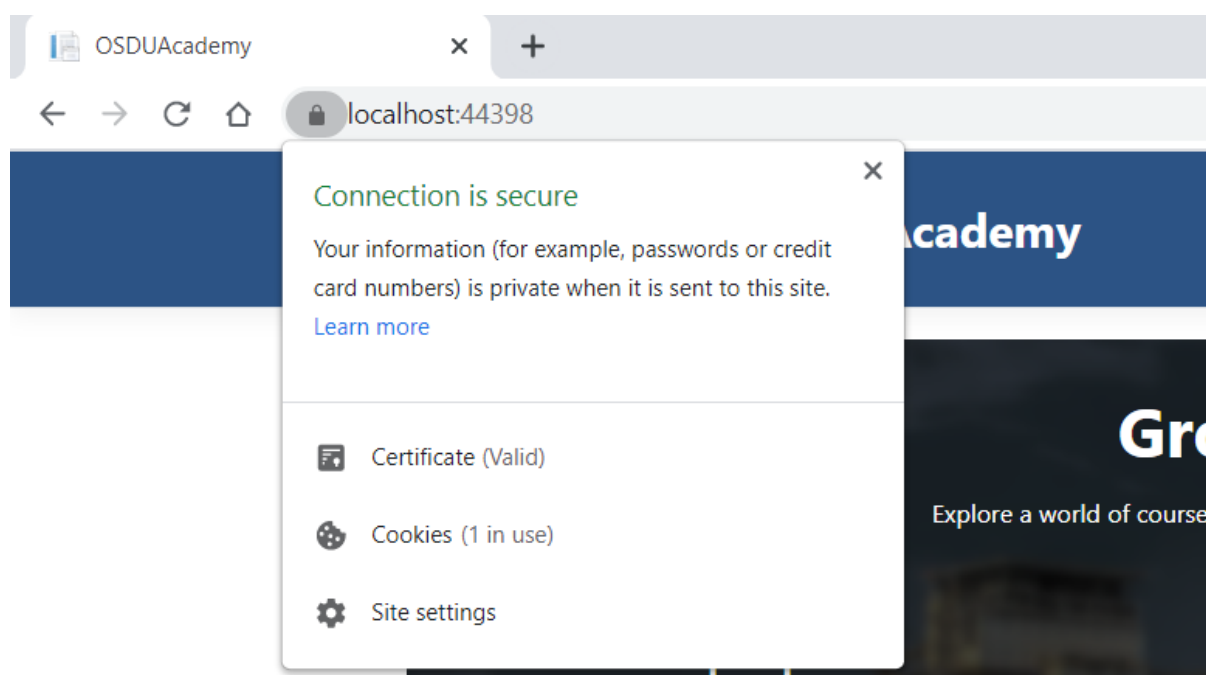


Figure 5.26: Secure connection as seen using Google Chrome

5.2.9.5 Injection mitigation by design

Code injection is an attack that back-end developers can sometimes overlook when implementing a communication method to the database. When creating an abstraction layer for database communication, one must use the database management engine's API and query language to make commands accordingly. In code, these commands are usually written directly and concatenated with user input where it is needed. This is a large vulnerability, since attackers can simply input certain characters to modify the command and retrieve sensitive data depending on the query language's syntax. Preventing this type of attack is mostly a code design choice and tools to mitigate this is commonly provided by database management engine driver APIs. In C#, there exists a driver for MongoDB that provides C# classes and methods which performs all the needed commands without needing to manually concatenate them with user input. Additionally, these functions perform input validation which is also key to preventing code injection.

Here is a C# code example of how such a function is used:

```
var cars = _carsCollection.Find(c => c.Name == "Mitsubishi").ToList();  
Console.WriteLine(cars);
```

And here is what MongoDB command the code corresponds to:

```
db.cars.find({name: "Mitsubishi"})
```

Notice how there was no trace of any MongoDB query language in the C# code snippet. All input validation is done through the "Find"-method and there is no possibility for an attacker to inject MongoDB code and succeed, since the command is handled safely through the driver.

5.2.9.6 Remaining vulnerabilities

While there was a magnitude of security concerns covered in the making of Prototype 2, the system safety is not complete. Attacks such as distributed denial of service and brute-force attacks can easily affect this product since there is no mitigation system implemented. Additionally, XSS attacks, as described in paragraph 5.2.9.2, was also an uncovered security concern. Due to our minimum viable product standards, time and

resources was put into user experience related topics and areas. The purpose of the usability tests was to test usability, not security. Nevertheless, security was considered for the sake of scalability and to simply notify about possible vulnerabilities. The larger a software project becomes, the more vulnerable it will be too.

5.3 Product limitations

We had several requirements set for Prototype 2 as part of the Software Requirement Specification in Appendix A1. In this particular development process, the requirements changed a lot over time and we ended up switching the priorities for each requirement in the document. In the end, we decided to implement as many P1 priority requirements as possible, disregarding all P3 priorities and only including a few P2 ones. This was supposed to make the development process more achievable and narrow down the amount of features in order to make the tests less overwhelming with respect to qualitative analysis. All the features that *were* implemented in Prototype 2 was already described in this chapter, so the following paragraphs will analyse the features that were removed.

5.3.1 Course editor

E-learning platforms are largely supported by its content creators. Thus it is important to serve a comprehensible and usable system for creating courses. Our initial idea was to create some form for a course editor directly in OSDU Academy's interface, yet this feature was too difficult to design and implement. It required an entirely different target audience from the students in mind. This is why we ended up shifting our focus towards designing and implementing the application for the students, while simultaneously creating a very simplified temporary method for teachers to create content for OSDU Academy. The temporary method allowed the application to fulfil the requirements of a standard e-learning platform.

5.3.2 Registration methods and role selection

Initially, teachers would have the same role as the students in addition to creating courses. However, as mentioned, we did not implement the course creation interface, and as a result, the role selection when registering was not a necessity. In the final product, the

user registration itself is not permitted either, as the only accounts one can access are predefined in the database for testing purposes. A vital part of many web applications is a user authentication system, and OSDU Academy is equipped with a limited one. Scrapping registration resulted in being a positive security decision since usability testers would not have their sensitive information subject to potential attacks. Furthermore, the preference was to implement federated login, which could have alleviated the security concerns on our part. Nevertheless, due to the time limit and unfamiliarity with the technology, different measures had to be taken.

5.3.3 Ratings and forums

The forum feature was almost as large of a system as the entire prototype. The amount of work required to implement course content storage would be nearly the same. Additionally, there would be a need for a reply system and a way to rate each question, depending on the application design. This large system simply did not convey the purpose we had planned for a minimal e-learning platform where the main selling point is to take courses. The same reason applies to scrapping the review and rating system. The user comments and reviews need some form for moderation, thus causing several other safety concerns. Additionally, the rating system was meant to complement the search engine and a course recommendation feature. While these features do indeed improve the quality of the product and serve good purposes on the side, they do not show what an e-learning platform is supposed to do.

5.3.4 Search engine and course recommendations

The search function was removed, as well. Despite this being a useful tool for growing platforms with large amounts of content, a minimum viable product like Prototype 2 did not need it. We already made sure there was a way to find desired courses based on domains in the home page. On another note, course ratings were removed and this resulted in the search engine becoming less useful since filtering based on ratings would not be possible.

The last and most complicated feature to be removed was the course recommendation system. While this is a good quality-of-life feature for the users, the implementation process

is too large. This requires understanding several statistical algorithms and optimization topics which was not the focus of this thesis.

When all is said and done, this is a minimum viable product, and heavy limitations are added to enable tests rather than production value.

6 Results and analysis

So far, almost every iterative user experience and user-centered design principle have been displayed in practice. The last vital part of UX and UCD is to analyze the feedback and draw certain conclusions from them. We will show the composed data from the two prototypes and reflect upon their outcomes in this section.

6.1 Prototype 1

The low-fidelity prototype's feedback was collected by several surveys conducted from the usability test as discussed in chapter 4.7. The first survey queried the testers about their general background information, which will show some relevance further down this chapter. We asked questions after each task during each test and then some more general ones after the entire test per person. Additionally, we tracked their performance with numerical values like completion time and noted their behavior on each task.

6.1.1 Scenarios and tasks

Before diving into the information we gathered, the scenarios and tasks the participants had to complete should be explained. Further down is a list of all the scenarios and an explanation of why we chose to create them and what parts of the prototype they should help test out.

6.1.1.1 Scenario I: Registration

You are a student named Jens who finds our platform intriguing and decide to be a part of it. Find a way to register an account.

The first scenario should be the easiest and quickest one. The point is to have the user find a way to navigate to the registration page when they are greeted with the first thing they see: the home page. This should let us know how they feel about accessing the platform's fundamental parts which is very crucial for attracting new users.

6.1.1.2 Scenario II: Finding a relevant course

You get an email from your boss about the next assignment involving multiphase flow meters. You are not very knowledgeable about the topic and want to learn more. Explore the platform until you find a course that suits you.

After the user has found a way to register an account, they should be back at the home page and logged in. This scenario tests two ways a user can look for a course: using the search bar or by looking at trending courses at the further down the home page. In addition, it tests how the user reacts to applying to a course since they should navigate to its front page in order to apply.

6.1.1.3 Scenario III: Login and re-using the platform

You have used the platform before, but have not come back for a month. You need to catch up on learning how to use Techlog Wellbore, a wellbore analysis software. Log in and try to find the course.

Similar to Scenario 1, the user is presented with the home page and logged out. This should test the learnability of logging in and the course history tab further down the home page.

6.1.1.4 Scenario IV: Continue with previous work

You have only one chapter left of your course named "Techlog Wellbore - Beginner Tutorial". You wish to read up on the last few pages and then take the test, Since you are already logged in on the website, try to continue with the course, find the test and take it.

This last scenario will test the core feature of OSDU Academy: learning in courses. It is important to keep the learning of the course short, since we only want to test the interface and not eventual course content. Lastly, the user will try the certification test and be greeted with the display a certificate. We wanted them to rate the certificate's design in this last part of the scenario.

6.1.2 Quantitative findings

Analysis is a part of the usability testing. We gathered quantitative data for every user and task. By analyzing it and studying statistics, one can discover common behavior, possible design changes, and inconveniences.

Some of the measures we used in the testing process are time on a particular task and task success. The latter could be described as a complete success with a few minor difficulties. Every participant did outstanding work and solved every task. When it comes to the former one, the results were a bit blended, as two of the testers used a very long time to solve tasks. For instance, the last participant used almost eighteen minutes to finish the fourth scenario 6.1.1.4, which is thrice as the task average, see figure 6.1.

As the first two scenarios (6.1.1.1 and 6.1.1.2) were about registering and finding a course, one can compare those with the third task 6.1.1.3, since it requests the participant to sign in again and continue with another course. Here we can test the learnability ⁷. From the "Time overview" figure, it is clear that users needed less time to complete the third task than the first two. It means that they got used to the product quickly and were able to repeat the action swiftly.

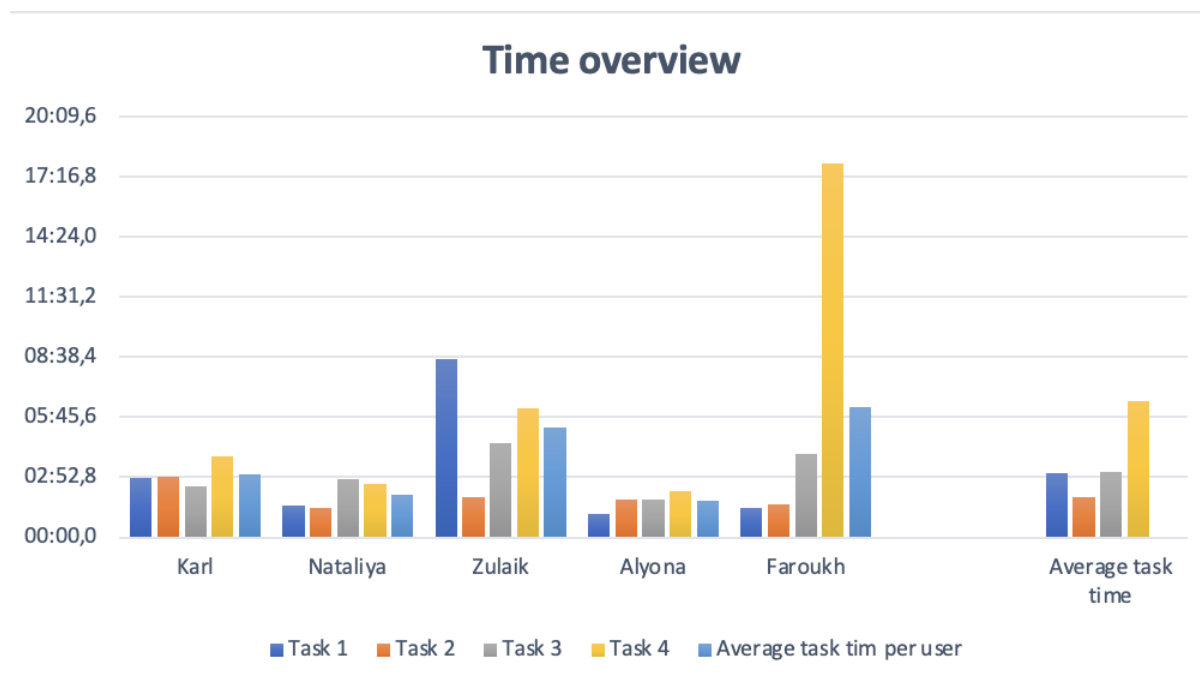


Figure 6.1: An overview of time usage

⁷The quality of an interface that allows users to quickly become familiar with them [34]

However, it is not hard to notice that the third participant used eight and a half minutes to accomplish the first task 6.1.1.1, read about the reason in the sub-chapter 6.1.3. We only had five testers, which means that the high timing had a big impact on the average task time. As a result, we decided to sort the time used in ascending order and find the median to get a better representation which is shown in figure 6.2. Indeed, the participants used a shorter time to repeat the actions from the first two tasks, but the difference is not drastic, as represented in figure 6.1. Furthermore, the results show us that participants with an age range between twenty-five and forty-eight are more comfortable with the product and use less time to complete the objectives. The same can be said for their occupation, as the second and fourth participants had the lowest average time per task. Those two are two software engineers working as, respectively, lead architect and project manager.

| Median | | | | | |
|--------|---------|---------|---------|---------|---------|
| Task 1 | 01:06,2 | 01:24,9 | 01:31,4 | 02:50,4 | 08:32,2 |
| Task 2 | 01:25,2 | 01:34,3 | 01:49,5 | 01:56,2 | 02:54,0 |
| Task 3 | 01:49,9 | 02:27,0 | 02:47,6 | 04:00,5 | 04:29,6 |
| Task 4 | 02:12,3 | 02:32,4 | 03:51,8 | 06:10,6 | 17:54,9 |

Figure 6.2: Time median for every task

Our expectations were similar to the median value of each task, which is highlighted in yellow on figure 6.2. As planned, the first and second scenarios did take the shortest amount of time, and there is no surprise that the final task took the longest as it did require multiple objectives to be achieved.

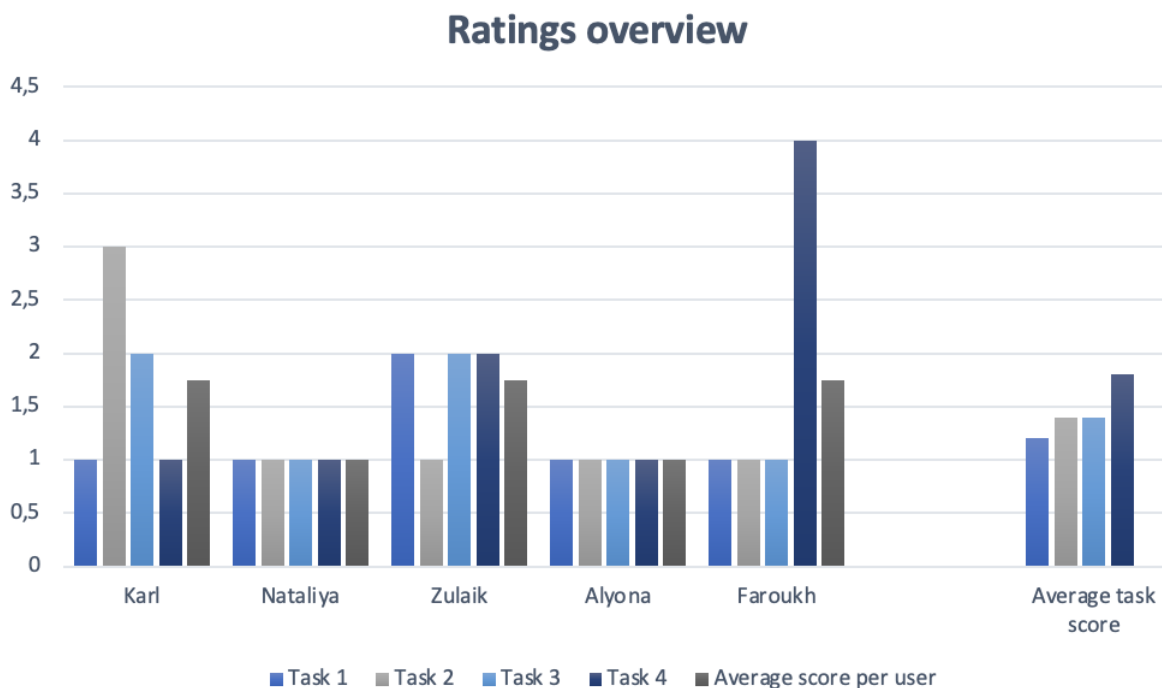


Figure 6.3: Rating overview

We have collected all task difficulty ratings given by all participants in figure 6.3. The rating method consisted of a scale from one to five, with one being easiest and five the hardest possible. The ratings were low in the majority of the cases, but none of the tasks got all ones. The most difficult scenario to do was the last one, with the average at 1.8. The highest ratings particular tasks got are four (Scenario 4 6.1.1.4) and three (Scenario 2 6.1.1.2). By looking at figure 6.3, one can see that the first task was the easiest one, as the participant gave it 1.2 on average. Once again, we can confirm that the participant with a software engineering background found the tasks easier; both rated them with straight ones. The other three testers had equal average ratings for the tasks (1.75).

As mentioned, the task which was most difficult to accomplish is the last one. The user was supposed to finish a course and take the certification test. Even when three out of five users rated it as simple, the other two had some trouble understanding the interface. The biggest issue in our interpretation is that users were not sure where they are located and how to continue with the work. We have not implemented any visible features on the page to give them that information, except the overview at the lower parts of the page.

A confusing fact is that the third scenario got an average score equal to the second task;

even its highest rating was two. This came as a surprise when one takes into consideration that the user has taken the same steps in the previous tasks, has logged in, and found the course. Henceforth, the data itself contradicts our statement and conclusion about learnability. It would seem that the users did not get more comfortable with the interface purely by looking at the data.

6.1.3 Qualitative findings

For every task completed by each participant, we asked some questions regarding that particular task. After the session was done, we conducted another questionnaire about more general thoughts from the participants, such as difficulty and design changes they would like to make themselves. Additionally, many of the qualitative data was gathered during the test itself, since every participant expressed their opinions as they were doing their tasks. In correspondence to facts about emotional design, as described in chapter 3.3, some of the reactions people had to the product can be understood better.

Do you think the registration process was missing some options for personalization?

We asked this question since we expected that the user might feel the lack of personalization. Our design never featured such a thing, but we knew it might need it in the future, so the best way to determine another requirement was through users.

One of the participants, who was a software engineer, suggested that there should be opportunities for selecting your domain of interest so that the user can get more relevant course recommendations. However, they meant this feature depended on the platform or that it could be added in later stages. Otherwise, the rest of the participants said that the registration process was very clear, simple, and enough, with one exception. At the time, we were not certain if the platform would be an internal or an external Schlumberger website, and one participant asked about this during the test. We assumed it was internal for the sake of the test, and this particular tester suggested that the registration process was unnecessary since they expected the site to log them in with their Schlumberger account by automatically detecting it from browser cookies. This response is certainly interesting since it allowed us to look into more varied perspectives on registration, despite our intentions for the process to be simple and quick.

What kind of information would you like to see on the course page?

All testers thought there needed to be more information about the course on the front page, with one exception. They thought it would be great to know the course's difficulty level, such as beginner, intermediate, or expert, to know what they are getting into. Another common remark was the lack of information about how large the course is. Many of the participants suggested that there should be an indicator for the course's duration. They mean this would be useful for people that need to plan ahead in their schedules. Aside from that, all the participants thought the page was clear and concise, with one who really liked the colors. One participant, who was also a project manager, thought that the course page could use some form for course suggestions for people who intend to complete the particular course. Also, they'd like to know if the course is relevant for them, assuming the platform records their course history for calculating relevance.

Why did you choose to find your course the way you did?

This question was asked with the front page course suggestions and search bar in mind. We were aware that the design featured two ways to look for a course in the third task, where the participants were asked to find one. Our intention was to find the most optimal path to their goal, but we initially designed it in such a way that they would have to scroll down on the front page to find the course via the "course history" area.

The participants found our question quite unexpected since it seemed like we were asking about something too obvious. Nonetheless, the answers to this question were varied. One liked to use the search bar since it is what they usually do, and another participant just couldn't notice other options on the front page, thus being left with the search query option. These answers came from the participants with software development experience, so there is something to consider here. The others seemed to like exploring and testing the waters by scrolling or clicking various buttons they expected would lead them to the desired goal. One of these curious testers mentioned that the "course history" tab needed to show that the course history belongs to the user and suggested using the word "you" more. Lastly, one participant wanted the course history to be more apparent on the front page by replacing the large "OSDU Academy" banner with the course history tab and additional useful information.

What do you think about how the different learning mediums were represented?

This question relating to the fourth task about using a course had mixed answers. The first participant liked the way the course content was "in your face." In other words, it meant that the content was large enough and very visible. Others had similar thoughts and liked the variety of course content representation methods, namely video and text. These were common positive comments, but there was more constructive criticism. Many thought the use of arrows was good but that they were not enough for them to know how far they are into the course. Some suggested a bar on the left showing an overview of the chapters, despite that option existing by scrolling to the bottom. The first participant also thought the chapter title-bar with the next-previous arrow buttons was a title for something else further down the page. Speaking of which, not many noticed that they could scroll to the bottom to reveal the chapter overview; only one was saying it didn't belong there, so there was little to collect from this.

Since the course featured a certification test, the participants had something to say about this too, but not much. Some thought it was difficult to find the test since the arrows were not indicative of how close they were to take a test. Otherwise, the participants said the test itself was very simple and clear. One person said they were fairly confused by how the scrolling worked in this part of the prototype, but they still managed. The only constructive feedback we received about this section was that the user needed to know how much time is left on the test; otherwise, they thought the test information was quantifiable. When presented with the certificate at the end of the test, the participants had very positive feedback about the extensive features at post-certification.

Did you expect any additional features to appear in the course interface?

As mentioned earlier, some people suggested a left, collapsable side-bar with the course chapter overview. Two of which thought the bottom "overview"-title is too distracting when the attention should be on the course content. Another person expected a feature that would let the user know that a test is approaching them in the last chapter. Lastly, someone suggested that the course content could be enlarged into full-screen, especially on the video.

6.1.4 Additional participant remarks

Aside from answering questions, each participant had a lot to say during the test. The method of thinking out loud has been a great help to us in this usability test. In this last section about Prototype 1, the post-testing discussion will be covered, as well as the behavior and active thoughts from every participant.

6.1.4.1 Common behavior

The most noticeable behavior in each participant was the trouble they had with finding their course history on the front page. This could be a result of the large banner not indicating enough that the user is signed in. Another reason they may not have noticed the course history tab could be the symptoms of a paper prototype, like scrolling, in general, was not noticeable on a static image. On that note, the "sign-in"-button was mentioned, to some degree, by everyone since they were slightly confused that there was no button for specifically registering. However, everyone was able to assume the registration form could be found from the "sign-in"-button. On the topic of courses, each one had a rating under its title on the menus. The intention for these ratings was merely to act as a placeholder to indicate that a rating system will be in place, but everyone commented on them and would only often take the courses with high ratings rather than focusing on the title of each course. Lastly, the bottom part of the course interface was ignored by almost everyone.

6.1.5 Further insight

After every testing session with a participant, a few general questions were asked about their experience. Due to their vocal opinions during the test, not many of these questions could be answered directly, so some fell short.

What do you think about the design and would you use this app?

Everyone except one answered yes to using this platform. The remarks summarized by every person conclude that the design was straightforward and usable. However, one person mentioned that the platform was confusing at first but felt that they could manage.

Was there anything you expected to find that was not there?

Surprisingly everyone effectively answered no to this question, despite everyone having some experience in e-learning usage. After some discussion with the participants, we were able to hear some of their ideas for improvement again, but otherwise, there was nothing, in particular, they had to say.

What was unnecessary, if anything?

The majority of participants did not have many criticisms about unnecessary elements. For instance, the first user did not like the home page's design as it was unclean and cluttered. The lower part of the home page when the user is not logged was confusing for them. Moreover, on the same page, when the user is signed in, the first participant thought the large banner showing the name of the product was too much.

What would you change?

As this was a low-fidelity prototype, we did not implement all the features we had on our minds. A few of those have been suggested as potential changes by the users. I.e., small details on the course page, such as course duration and showing the number of reviews.

6.2 Prototype 2

For the high-fidelity prototype the feedback was collected in a similar way to the first one. We conducted a pretest survey for some of the newer participants, yet skipped that part for the previous testers of Prototype 1 since the survey did not change for this test. Just like the first prototype, we asked several questions regarding each task and some general ones at the end. Additionally, we added extra questions based on the areas the users were struggling, such as noticing the overview button at the top-right corner in the course interface.

6.2.1 Scenarios and tasks

As done in the previous prototype test, some scenarios and tasks were delivered to the participants for them to put the product into context. It is important to cover them in this chapter since it is frequently mentioned in the analysis.

6.2.1.1 Scenario I: Revisiting a course

You are a student named Johannes who already has an account on OSDU Academy. Today you decide to complete a course you enrolled in a few days ago. You remember that it was teaching a program called "Techlog Wellbore" on a beginner level. You already went through first three chapters and feel confident you will pass the certification test of the course. Sign in to your account, find the specified course, skip to the final chapter and try to pass the certification test.

Here is your account information:

- Email: jens.olsen@email.com
- Password: password

In contrast to the first task 6.1.1.1 from the previous testing session, the first scenario is not the easiest one. Yet, it should not be difficult. The main goal is to let users test the platform's basic features and navigate through the course. Furthermore, the user is supposed to pass the certification test as it lays the foundation for later scenarios.

6.2.1.2 Scenario II: Apply for a petroleum geology course

You have been assigned to a new project at your job which requires basic knowledge about petroleum geology. You think a relevant course might be present in OSDU Academy, so you want to apply for one for later use. Find a relevant course, check its relevance and apply for it if suitable.

This task targets the users' understanding of the category navigation bar on the home page, used as a filter for the different domains. Moreover, the tester gets the chance to become familiar with the information on the course front page and express their opinion about it. Additionally, it checks how the user experiences applying for the course.

6.2.1.3 Scenario III: Your certificate status

After you got certified in the Techlog Wellbore course you took earlier, you forgot to check if the certificate was registered. Now you want to check if you got a certificate and call it a day. Find a way to access your account status and then log out.

The users are expected to access their profile to find the certified courses and the log-out button. In this way, the users can get familiar with the profile page and comment on its features.

6.2.1.4 Scenario IV: Taking the petroleum geology course

You are now ready to take the petroleum geology course you applied for earlier. Find the specified course and take it.

The last scenario is testing the learnability of the entire platform. As the user has logged out as a part of the previous task, they need to log back in and find the specified course to complete it. In this task, the user can test different features from the first three tasks and get more familiar with the product.

6.2.2 Quantitative findings

Quantitative analysis is an integral part of the evaluation process. Once again, we have gathered the data and analyzed it to discover possible issues and common behavior.

We have considered the previous prototype testing and decided to use the exact measurements, namely time on a particular task, task success, and users' difficulty rating on a scale from one to five. Every tester, without exception, solved all the tasks successfully. The average time (See figure 6.4) users took to complete the task has varied between the participants.

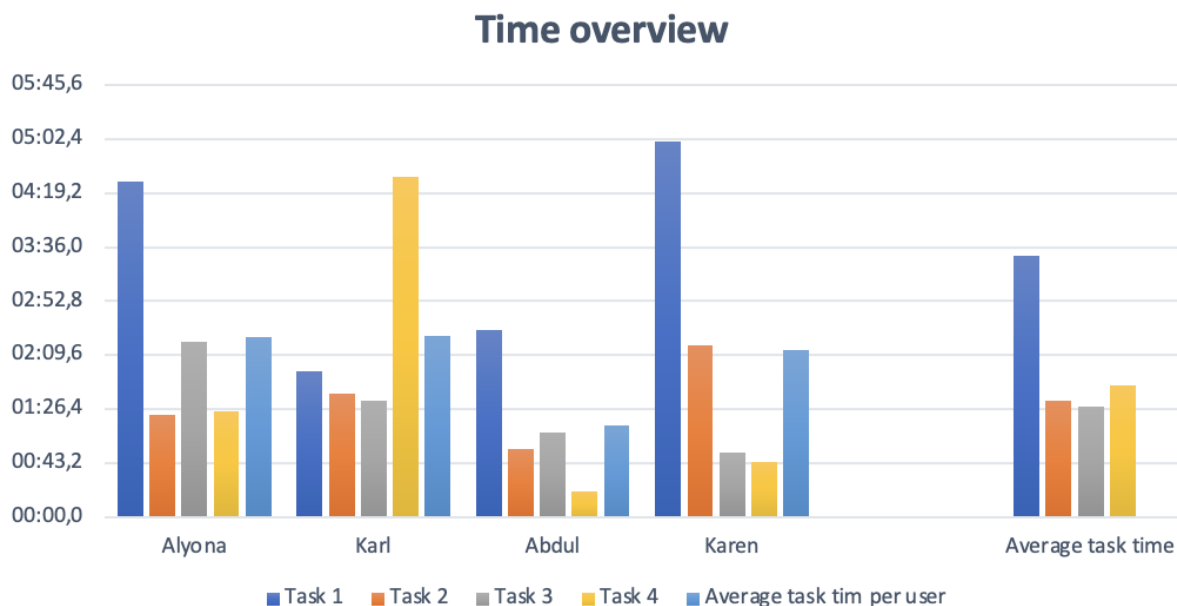


Figure 6.4: An overview of time usage for the second usability test

The first (6.2.1.1) and last scenarios (6.2.1.4) are similar to a degree, but the testers used less time to finish the fourth. The data suggests that our objective of creating a simple, easy-to-learn application has been successful, as the platform's learnability seems to be great. From the figure 6.4, we can confirm the statements as the final scenario's average time is twice as short as the first. From our observations, we could say with safety that the users have become more familiar with the software and can repeat the actions without any trouble. In comparison with the similar tasks in the first testing session, users have, on average, used less time, both the new ones and the two who have tested the first prototype. The results come as no surprise because we expected that outcome. Our goal was to make an application that would be simple enough for all the users with general computer knowledge.

| Median | | | | | |
|--------|---------|---------|---------|---------|---------|
| Task 1 | 01:56,2 | 02:30,0 | 03:29,5 | 04:28,9 | 05:00,6 |
| Task 2 | 00:54,0 | 01:21,7 | 01:30,1 | 01:38,6 | 02:17,8 |
| Task 3 | 00:51,3 | 01:07,3 | 01:20,4 | 01:33,5 | 02:20,6 |
| Task 4 | 00:20,0 | 00:44,2 | 01:04,3 | 01:24,4 | 04:32,8 |

Figure 6.5: Time median for every task in the second usability test

In contrast to the previous testing session, we did not have to use the median to represent

the time usage, as the average time used is close to the median. Yet, we have decided to do it to represent the fourth scenario more precisely (See figure 6.5). The reason is that the second participant has used over four and half minutes to complete it, and the median is just above one. Moreover, we could not help but notice that the average time per user was almost the same for every participant, except the third one, who has done the tasks much faster.

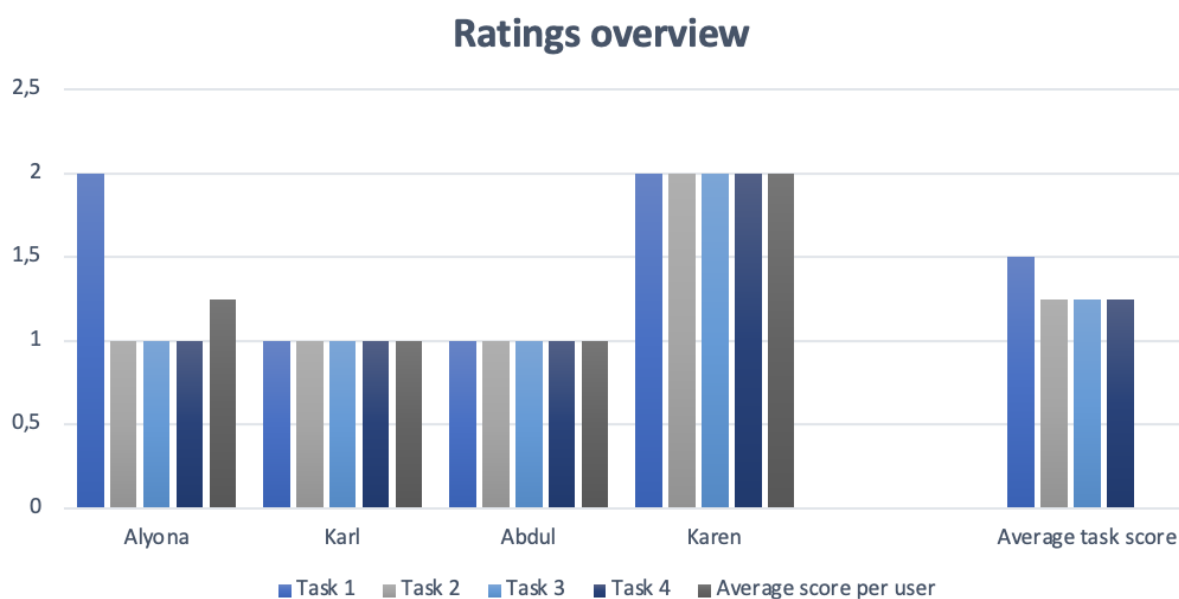


Figure 6.6: Rating overview for the second usability test

As mentioned above, the participants have rated the scenarios on a scale from one to five, where one is the easiest and five the most difficult. The ratings were low in the majority of cases and were almost straight ones. This can be seen in figure 6.6. The second and third tester rated all the tasks with one. The first participant rated all the tasks except the first one as easy. The first task was a little bit confusing for them, and they gave it a two. The last participant rated all the tasks with two. Based on their previous work and knowledge, it makes sense that the ratings were higher than usual. The participant has not used e-learning platforms as frequently as others and is not an engineer but a consultant.

When it comes to particular tasks, all the tasks except the first one got an average rating of 1.25. The first task got an average rating of 1.5. This is an improvement in comparison with the first usability testing. In the first usability testing, the lowest-rated task was

the first, and it seemed like the platform got more challenging to use in the further tasks. The second usability testing went just as we wanted it. The first task was most difficult, which is logical because the interface is new, but the later tasks were rated lower as we implemented the features in a similar fashion.

Furthermore, the data provides insight in that the background, previous knowledge, and job do not play a significant role in interaction with the application. The first two users have had a chance to get to know the platform at the previous prototype testing, but they did not have significantly better results than the other testers. This shows that the platform is simple enough to use for everyone in the field, no matter if they have used the platform before or not.

The overall result with the second prototype has been better than the first one. It seemed like the participants were more relaxed and less confused. This statement can be confirmed by looking at the numbers and the figures.

6.2.3 Qualitative findings

The qualitative data gathered in Prototype 2 stems from the same type of process conducted in the previous prototype. The only difference is that additional questions were asked based on a participant's confusion or struggles so that we could identify a problem with a certain feature or task. Additionally, some questions were added based on the high-fidelity traits present in this new prototype. Nevertheless, we asked predefined specific questions after each task and some general ones at the end of the session.

How easy was it to find the course and complete it?

Everyone thought performing Scenario 1, as described in paragraph 6.2.1.1, was an easy and a quick process with one exception. One of the testers from Prototype 1 thought the course navigation was confusing since they assumed that the chevrons on the sides navigated to different courses. It was not apparent to them that they were present within a course and viewing a lecture in the browser. Additionally, during the test we noticed that our first participant completely ignored the overview drop-down list in the course interface. As a result, we decided to guide them through the course interface at the end of the session to see if they could notice it. Otherwise, every participant was quick to recognize the workflows and they said they found the process simple.

As for exploring the platform and finding the course, there was little to note about the feedback. Everyone liked the simple nature of looking for a course, with only two exceptions. These two participants mentioned that a search function is something they prefer over scrolling around the home page.

What do you think about the layout of the certification test?

The first participant thought the certification test was much better than in the first prototype. This is interesting since the design has not changed much at all. We think it might be because of how the interface is presented in a browser that could have changed the look and feel since every widget was more centralized and smaller this time. The second tester liked the certification test as well, with only one thing missing: keeping the certificate somewhere. This participant liked the fact that they were presented with a "congratulations" text at the end, yet wished they could get their certificate emailed or store it in some form or another.

Besides that, the other participants had little to say about the certification test. Everyone thought it was simple and could not comment much on it since they had to think long about the negative sides.

Do you think there was enough information about the course on its front page?

Generally positive feedback here. All the participants had the common opinion that the front page for a course was very informative and that it did not lack anything. Another positive remark was from the third participant who thought the way the information was presented could attract more users to a particular course. However, the first participant had one small negative concern with how the course index was placed before the description. They thought it would be more intuitive to read a general description first and have the index below that in case opening the varying sections would eventually take up too much space. This is why we decided to create a new question for future participants, which will be covered next.

Custom Question: What do you think about the order of the index being presented before the course description?

As mentioned earlier, this particular question was created for the sake of receiving extra

feedback for a feature we think others might not comment on, yet might become a crucial future issue. While some agreed that the placement of the description and index was not intuitive for large courses, the common thought did not strike this as a large problem and the participants only noticed this problem as soon as we mentioned it. Nevertheless, they agreed that it could be problematic, yet that the features themselves were quite useful at its core.

Would you describe the profile page as simple enough or too basic?

Several testers expected a drop-down list when hovering over their profile icon to reveal some buttons for logging out or accessing other preferences. Despite this expectation, they all managed to navigate to the correct section and complete their task. As for the question itself, some participants argued that the page could be too simple depending on the information that is needed to be displayed. Otherwise, they think it was enough and everyone described it as simple. The second participant thought there should be an option to view their certificates in more detail since the option was not there. Additionally, this participant decided to flip this question back to us and ask if *we* thought the interface was too simple. We answered with a decisive yes, yet we thought it is enough based on our current minimum viable product standards. Nevertheless, this particular participant thought that there is a beauty of the simplicity in this design. One last remark that should be covered is from the third participant. They wished that there was a feature to change how the certificates are displayed. In other words, they would like a feature that can display the courses they've taken and completed from a range of options like lists or grids.

What do you think about how the different learning mediums were represented?

The answers to this question was a mixed bag. While there was mostly positive feedback, some deal-breakers were mentioned by each participant.

The first participant did not like the fact that the course content could be scrolled using an internal scroll-bar and would prefer there to be less scrolling options in a website generally. Additionally, they expected that there should be some hints somewhere on the page that indicates their progress on the course.

The YouTube videos were frequently criticised. The second participant mentioned that they did not notice it at all, and our fourth participant preferred that videos, in general, should not be presented from YouTube because it was easy to accidentally click the link that leads to the original video.

The third participant thought the learning mediums were presented very well, yet that the presentation is not unique to any other e-learning platforms. They also mention that scrolling down the course content and the overview tab were not very apparent at first glance.

Aside from these unique criticisms, the testers found the course interface and learning mediums to be presented fairly clearly. Additionally, our second participant loved the simplicity and even mentioned that building on this base design could bring a good future for the platform.

Was there anything that surprised you in this task?

In relation to the fourth scenario described in paragraph 6.2.1.4, everyone answered this question with a no. Only our second participant mentioned that they were positively surprised by the simplicity of the layout. It seems that the simplicity and straight-forward nature of OSDU Academy's design elicits familiarity among the testers.

What do you think about the responsiveness of the website?

This is a post-testing session question related to the high-fidelity aspects of Prototype 2. Besides everyone saying that the website was responsive and did the job well, there were small remarks, both positive and negative. The third participant thought the lack of animations helped the application be more responsive. This comment is in no relation to technical performance, which came to a surprise to us since we were slightly concerned with that area for a long time. On the other hand, the fourth participant noticed that the courses on the front page took a little too much time to load in, yet that it was not a big issue and that some more apparent indication that loading occurs could make the website more comfortable to use.

6.2.4 Changes and improvements

Based on the feedback we collected, we were able to distinguish some areas that contained flaws. There was little to improve based on the fact that Prototype 2 is a minimum viable product, since some of the negative criticism stemmed from expectations the participants had from using other *complete* products in the industry. Nevertheless, their comments were taken into account, especially in the last chapter of this thesis.

Improving Prototype 2 alone involves changing the profile page's features and the overview at the course interface. The latter mentioned could be more visible and perhaps vertically aligned with the course content. When it comes to the profile page, we should consider the drop-down menu at the navigation bar to choose to go to the profile page or log out. Additionally, the complete and active courses on the profile page can be featured differently. The participants have pointed out that it could be challenging to navigate and find the specific course there if they have many courses. An alternative for the horizontal menu with chevrons is a list with all the courses or a grid.

7 Conclusion and future work

The development of an application comes with its challenges and obstacles. The designers and developers have to take into consideration various requirements. The number of functionalities and features can often be pretty high and can impact the product's quality. One of the main objectives of this thesis was to define a minimum viable product and provide users with a fully functional user-centered prototype that can be used in the industry and built further on. To achieve this goal, we have researched various design principles and followed a user-centered iterative design process in the hope of finding suitable solutions for the platform.

This chapter summarizes the result from the testing session and includes possible future work with the platform. Moreover, we have expressed our opinion about the process and prototype here.

7.1 Conclusion

Based on the work presented in this thesis, there are several positive and negative results to conclude and discuss further. The end result of our work shows a great amount of things we wished to implement, yet had to compromise due to time and resource management. Nevertheless, there is much to learn from this experience and it is all for the better. We learned many ways to improve the quality of both the product and development process.

7.1.1 Design

The extensive amount of knowledge about user-centered design principles gave a good understanding of the basic knowledge about UX and UCD. Nevertheless, knowledge does not compare to practical experience. The tests alone gave us a large understanding for one concept: simplicity. Keeping the designs, discussed in chapter 5, as simple as possible was the reason the majority of positive feedback existed in our usability tests. Complicated work is tedious and unattractive for anyone and the user experience may become daunting if the design is too much to handle. Steering clear from any form of complexity also gave us developers an advantage in the process. There was much less to implement per iteration and grasping the core design was much easier in the long term.

Considering simplicity again, it is much easier for users to become familiar with a product when first introduced with it. Many graphical user interfaces in the industry feature similar layouts to each other, and our product is no exception. Based on our tests, it was much easier for the users to become familiar with our product, both due to simplicity and similar traits to other products in the industry.

7.1.2 MVP

This thesis gave us great insight on how development with prototypes work in the industry. It is an iterative process featuring several mock-ups of a product meeting minimal standards ready for testing. The prototypes, or minimum viable products, are meant to be tested with people from the corresponding target audience and evaluated accordingly. The tests are greatly important because the developers alone cannot evaluate the product due to several biases. It is human nature to stay biased to one's own work and this becomes a roadblock for improving a product.

There were several features we wanted to implement, such as a rating system, search engine and recommendation system, that could simply not make it into the prototypes due to the overwhelming amount of work and testing required. It is important to decouple features and implement them into small products so the tests can succeed faster and gather accurate data about specific needs rather than implementing for the sake of it.

7.1.3 Final conclusion

Designing and implementing a user-centered application is a long process. In this thesis, we have shown that with comprehensive research and by conducting usability testing with the relevant participants, one can learn a lot about the design and development and the approach in the software projects. Our work and understanding of various design principles resulted in an easy-to-use and uncluttered application presented to the testers in the oil and gas industry. In addition to suggestions for improvement, both positive and negative criticism has helped us realize what can be done better in the future and how to make an application that satisfies clients.

7.2 Future work

The future for OSDU Academy is bright considering how well we were able to perform tests and create a sufficient product that met our standards. Our tests were relevant and so were the participants. While there is much more potential in an e-learning platform than a few extra features, this chapter will cover only the ones that we expect to implement in a potential Prototype 3 or more others.

7.2.1 Search functionality

The platform did not have a significant amount of courses in the beginning, and, as a result, the search bar was not a necessity for the moment. If the platform grows and has over a hundred courses, it would become significantly more difficult for users to navigate and find a specific course with the current options. The solution for that issue would be a search function that narrows down the number of courses to more relevant user desires using keywords and filters. We have made a foundation stone for this feature on the front-end, and it is ready to be continued in the subsequent phases of OSDU Academy.

7.2.2 Supporting course instructors

We were fairly disappointed knowing we did not have enough time to create user-centered support for instructors. While our current solution works, it is a heavy task for someone to learn how to use the course templates and send it to a supposed administration in OSDU Academy. This process is time-consuming and tedious and we would prefer to have a user interface like we created for the students. In the future, we would like to create an interface for instructors to edit and create courses with creative freedom.

7.2.3 Federated login

The initial idea was to create a system to support login via external providers, such as Google or Microsoft. Doing so would improve login security on our end, since login credential security would already be handled by the external parts and OSDU Academy would not need to concern with it. Nevertheless, this was time consuming and did not really fulfill the minimum viable product standards we had set for the testing sessions. We

would like to implement this and explore more around the topic of federated authentication in the future versions of OSDU Academy.

References

- [1] User centered design. Interaction Design Foundation. [Online]. Available: <https://www.interaction-design.org/literature/topics/user-centered-design>
- [2] D. Nessler. (2016) A guide to paper prototyping testing for web interfaces. [Online]. Available: <https://medium.com/digital-experience-design/a-guide-to-paper-prototyping-testing-for-web-interfaces-49e542ba765f>
- [3] S. Bradley. 3 design layouts: Gutenberg diagram, z-pattern, and f-pattern. Vaneso Design. [Online]. Available: <https://vanesodesign.com/web-design/3-design-layouts/>
- [4] Human-computer interaction (hci). Interaction Design Foundation. [Online]. Available: <https://www.interaction-design.org/literature/topics/human-computer-interaction>
- [5] W. Community. Schlumberger. [Online]. Available: <https://en.wikipedia.org/wiki/Schlumberger>
- [6] S. Organization. Who we are. [Online]. Available: <https://www.slb.com/who-we-are>
- [7] The open group launches the open subsurface data universe™ forum. [Online]. Available: <https://www.opengroup.org/open-group-launches-open-subsurface-data-universe-forum#:~:text=The%20Open%20Subsurface%20Data%20Universe%20Forum%20is%20an%20international%20forum,data%20platform%20that%20will%20bring>
- [8] The benefits and drawbacks of online learning. Explore Talent Ims. [Online]. Available: <https://www.talentlms.com/elearning/benefits-and-drawbacks-of-online-learning>
- [9] About moodle. Moodle HQ. [Online]. Available: https://docs.moodle.org/310/en/About_Moodle
- [10] Gnu general public license. The Free Software Foundation. [Online]. Available: <https://www.gnu.org/licenses/gpl-3.0.en.html>
- [11] Rider. [Online]. Available: <https://www.jetbrains.com/rider/>
- [12] Lexico. Definiton of design. [Online]. Available: <https://www.lexico.com/en/definition/design>
- [13] E. Novoseltseva. (2020) User-centered design: An introduction. [Online]. Available: <https://usabilitygeek.com/user-centered-design-introduction/>
- [14] J. Parker. (2012) Business, user, and system requirements. [Online]. Available: <https://enfocussolutions.com/business-user-and-system-requirements/#respond>
- [15] Cambridge dictionary: Prototype. Cambridge. [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/prototype>
- [16] H. WayKen. (2018) Guiding tips of prototype development process. [Online]. Available: <https://www.waykenrm.com/prototype-development-process.html>
- [17] R. Hartson and P. Pyla, *The UX Book*. Morgan Kaufmann, 2018, vol. 916.
- [18] S. Bradley. Counterpart and counterpoint in typographic hierarchy.

- Vaneso Design. [Online]. Available: <http://vansodesign.com/web-design/counterpart-and-counterpoint-in-typographic-hierarchy/>
- [19] What is emotional design? Interaction Design Foundation. [Online]. Available: <https://www.interaction-design.org/literature/topics/emotional-design>
- [20] D. A. Norman, *Emotional Design*. Basics Books, 2003, vol. 272.
- [21] D. Norman, *The design of Everyday Things: Revised and Expanded edition*. MIT Press, 2013, vol. 368.
- [22] S. H. Rogers, Yvonne and J. Preece, *Interaction design: Beyond human-computer interaction*. Wiley, 2011, vol. 569.
- [23] R. F. Dam. (2020) Personas-a simple introduction. [Online]. Available: <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them?fbclid=IwAR0Bqae0nI1xAXJ89ko7Ff3Ik38B1fQvciV5zvpiPrlv83V4fe63-ktDHDk>
- [24] User story template. Agile Alliance. [Online]. Available: <https://www.agilealliance.org/glossary/user-story-template/#:~:text=A%20user%20story%20template%20is,information%20about%20that%20user%20story.>
- [25] K. Wieggers and J. Beatty, *Software Requirements, Third Edition*. Microsoft Press, 2013, vol. 619.
- [26] (2018) Functional and non-functional requirements: Specification and types. Altexsoft. [Online]. Available: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>
- [27] Software engineering | architectural design. Geeks for geeks. [Online]. Available: <https://www.geeksforgeeks.org/software-engineering-architectural-design/>
- [28] Design patterns for modern web apis. [Online]. Available: <https://blog.feathersjs.com/design-patterns-for-modern-web-apis-1f046635215>
- [29] What is a rest api? [Online]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- [30] TechSmith, “Usability testing basics.”
- [31] Microsoft, *Microsoft Power Point*.
- [32] J. Nielsen. (2012) Thinking aloud: The 1 usability tool. [Online]. Available: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>
- [33] Schlumberger’s official website. [Online]. Available: <https://www.slb.com>
- [34] I. Wigmore. What is learnability? [Online]. Available: <https://whatis.techtarget.com/definition/learnability#:~:text=Learnability%20is%20a%20quality%20of,all%20their%20features%20and%20capabilities.&text=A%20very%20learnable%20interface%20or,to%20interact%20with%20the%20system.>

Appendix

A1 Software Requirements Specification

A1.1 Introduction

With the growing demand for e-learning and easier access to learning material in all industries, the oil and gas industry is no exception. OSDU Academy will provide the learning needs of said actors in a robust, interactive and reliable environment. The consumers of the product will be categorized into two different roles which have common and separate privileges: teachers and students. The teachers should be the platform's facilitator of learning content and certification authority, while the students are the consumers of the available content.

A1.2 Functional requirements

A1.2.1 Home page

| ID | Priority | Key Drivers |
|----|----------|--|
| 1 | P1 | A central main page the platform identifies with |

Requirement description:

The user needs a comprehensible central page where the interface contains navigation towards the important components of the platform such as login and registration. The page should look eye-catching and identify the platform as it is the first thing the user will encounter when interacting with it, no matter if the user is new or has used the platform before. Additional navigational features include:

1. Featured courses
2. Introductory information
3. Footer with contact information

A1.2.2 Log-in

| ID | Priority | Key Drivers |
|----|----------|---------------------------------|
| 3 | P1 | server identification, security |

Requirement description:

1. Identify the end user.
2. Confirm that the user has permission to connect.
3. Connect the user to the application without compromising users' data.

A1.2.3 User profile

| ID | Priority | Key Drivers |
|----|----------|-------------------------------------|
| 4 | P1 | Save work, progress and preferences |

Requirement description:

Every user should have their own profile where all the work and progress is stored. All of the information included in the profile must follow a security standard, so that no data can be misused. User profile should also keep information on which course recommendations are based.

A1.2.4 Certificating

| ID | Priority | Key Drivers |
|----|----------|--------------------------------------|
| 7 | P1 | Authentic proof of learning a course |

Requirement description:

After completing a course the learner must prove their knowledge in the material. A multiple choice test should be available to the user when the course is completed. Passing the test will grant the user a certificate which can be publicly shown on the user's profile.

A1.2.5 Use of different media

| ID | Priority | Key Drivers |
|----|----------|---|
| 9 | P2 | Multimedia, productivity, knowledge sharing |

Requirement description:

The courses can have multimedia approach, meaning that the content should not be textual only. There are various media one can use, for instance videos or interactive images could be a method to engage student even more.

A1.2.6 Registration

| ID | Priority | Key Drivers |
|----|----------|--|
| 2 | P2 | A method to get users into the system safely |

Requirement description:

Users must be identified uniquely and authenticated. Authentication is important for ensuring the user is human, has no malicious intent with the software and to secure the user's progress and personal data. The method of registration will allow the system to know that the real end user has intended to use the platform.

A1.2.7 Course management

| ID | Priority | Key Drivers |
|----|----------|--|
| 6 | P3 | The opportunity to create and change courses |

Requirement description:

Teachers who want to provide content on the platform need a way to create and manage the courses accordingly. There should be a user interface for the tools that allow a teacher to accomplish desired outcomes with the course in question. The teacher should be allowed to edit or create the following for their courses:

1. Multimedia learning material
2. Lecture ordering
3. Interactive exercises
4. Certification method

A1.2.8 Feedback and reports

| ID | Priority | Key Drivers |
|----|----------|---------------------------------|
| 10 | P3 | Feedback, analysis, improvement |

Requirement description:

Every course should have a grading system, so that the students can share their opinions and give a rating on a scale from 1 to 5 stars. In this manner, one can make filter courses by their rating and the teachers can improve their courses accordingly or keep up with a good work if the feedback is satisfying.

A1.2.9 User Roles

| ID | Priority | Key Drivers |
|----|----------|--|
| 5 | P3 | Separating user privileges based on platform contribution method |

Requirement description:

The platform is an open e-learning application where the users can both provide or consume the available learning content. When registering, the user should be granted the option to be represented as a teacher or student, thereby having different privileges on the platform.

Teachers will have the following privileges:

- Provide learning content
- Facilitate learning certification

Students will have the following privileges:

- View and apply for courses
- Provide feedback for applied courses

A1.2.10 Forum

| ID | Priority | Key Drivers |
|----|----------|-----------------------------------|
| 8 | P3 | Access for communication and help |

Requirement description:

Every user should have a medium to get help and communicate with other users which have taken or are currently taking the same course. In this way, not only the teacher of the course can answer, but also every other student who has the access to the forum. Furthermore, students can share their experiences and recommendations.

A1.3 Non-functional requirements

A1.3.1 Scalability

| ID | Priority | Key Drivers |
|----|----------|---------------------------------------|
| 11 | P1 | Capacity for updates and data storage |

Requirement description:

Since the platform is used by many people, it should hold the capacity to be extended and updated. The architectural design will take into account managing the course and user data along with opening the opportunity for more features which is important for the maintenance phase in the SDLC.

A1.3.2 Security standards

| ID | Priority | Key Drivers |
|----|----------|--|
| 12 | P1 | Confidentiality, integrity, and availability |

Requirement description:

The platform must follow a certain security standard in order to achieve the required security goals. The ISO9000 standard will be taken into account at all security areas of the development.

A1.3.3 Responsiveness (Performance)

| ID | Priority | Key Drivers |
|----|----------|-------------------------------|
| 13 | P2 | Usability and user-experience |

Requirement description:

The interaction of the application must be as seamless as possible. This means that

navigation using buttons and widgets is fast, while other interactions such as loading a different page can be compromised in a small amount when considering system resources (depends on the platform which it is deployed to). Nevertheless, transitions will take as little time as possible.

A1.3.4 Down-time (Serviceability)

| ID | Priority | Key Drivers |
|----|----------|--|
| 14 | P2 | Balancing maintenance and availability |

Requirement description:

The hardware utilized by the application requires maintenance from time to time, which can result in down-time. This means that the system resources must be utilized accordingly such that down-time is minimized, however it can be inevitable. The users should know about an upcoming maintenance and that the system is undergoing some form for down-time.

A1.3.5 Appealing design

| ID | Priority | Key Drivers |
|----|----------|---------------|
| 15 | P2 | Visual design |

Requirement description:

The design should be eye-catching and easy to be familiar with. There are many aspects one has to think about when it comes to designing. One has to think about the interaction between the user and the application and how easy it is to navigate around. In addition, the designer should use an appropriate color-scheme.

A1.3.6 Support access to external resources

| ID | Priority | Key Drivers |
|----|----------|--|
| 16 | P3 | System resource conservation and easy content management |

Requirement description:

The application should support access to external resources, such as YouTube. For instance, it should be possible to put links or embedded videos from other sources as

a part of the course content. This will make it easier for the content providers to add content to their courses, and saves space on the servers which host the multimedia.

A1.3.7 Alternative log-in methods

| ID | Priority | Key Drivers |
|----|----------|------------------------------------|
| 17 | P3 | Registration usability improvement |

Requirement description:

There should be more than one login method. In addition to registering using the platform's base registration method, the user should be able to use alternative ways. Some examples are using a Google account and a Facebook profile.

A2 Persona - Sofia

A2.1 User profile

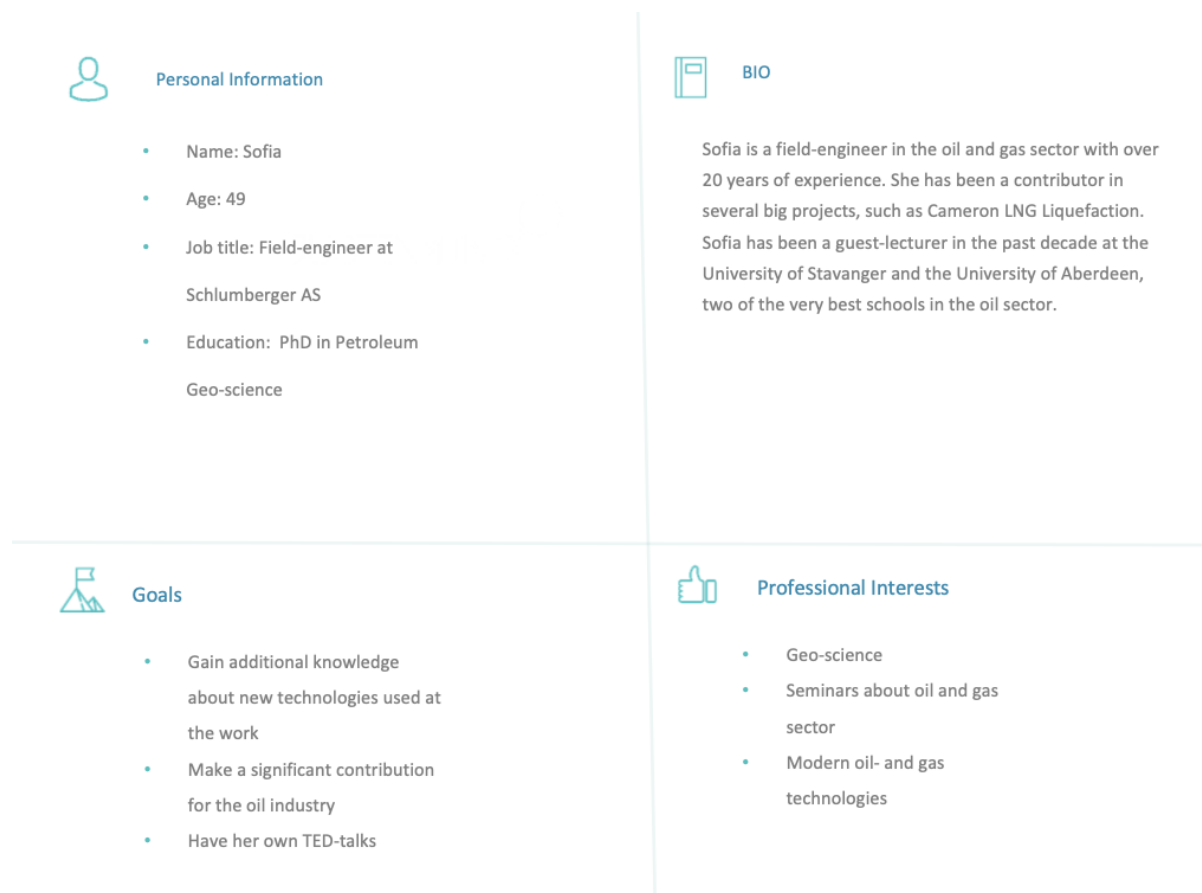


Figure A2.1: User named Sofia

A2.2 User story Sofia

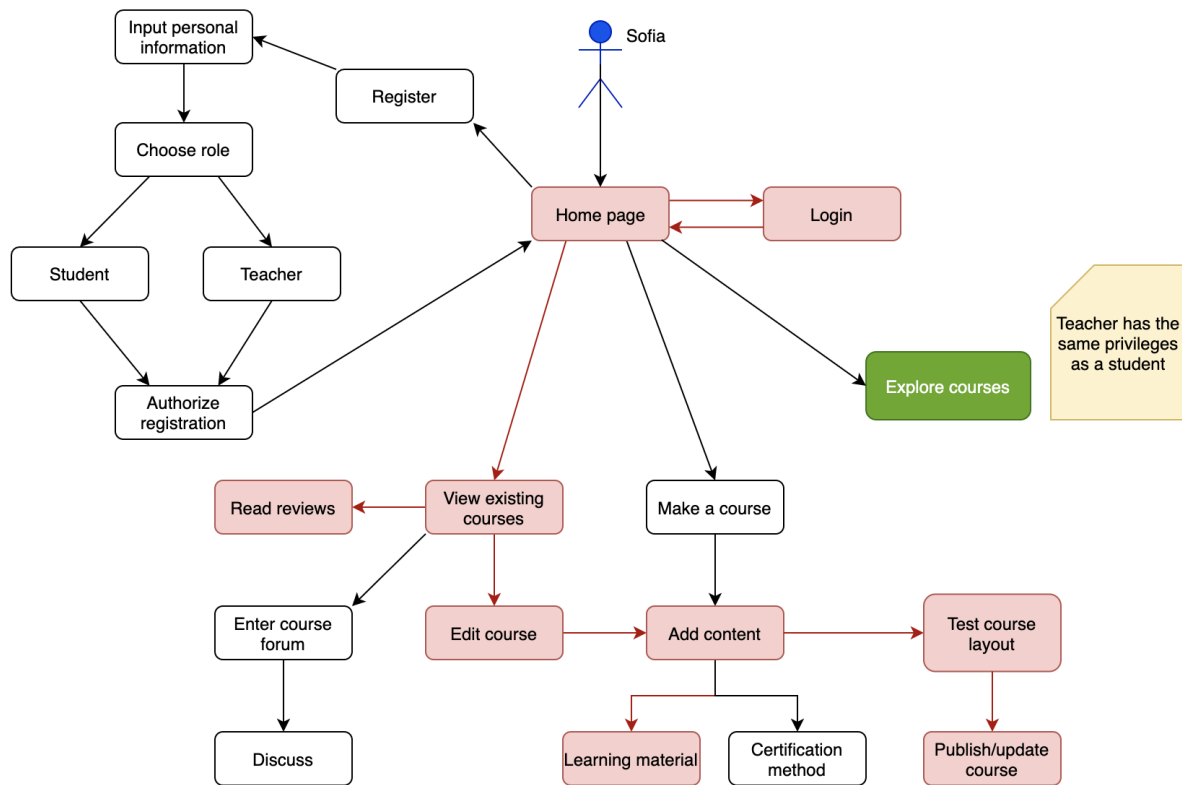


Figure A2.2: User story - Sofia

A3 Prototype 1 - testing session

A3.1 Scenarios and Tasks

The following section contains the scenarios and tasks given to the testing participants in the usability test for Prototype 1.

A3.1.1 Scenario/task I: Registration

You are a student named Jens who finds our platform intriguing and decide to be a part of it. Find a way to register an account.

A3.1.2 Scenario/task II: Finding a relevant course

You get an email from your boss about the next assignment involving multiphase flow meters. You are not very knowledgeable about the topic and want to learn more. Explore the platform until you find a course that suits you.

A3.1.3 Scenario/task III: Login and re-using the platform

You have used the platform before, but have not come back for a month. You need to catch up on learning how to use Techlog Wellbore, a wellbore analysis software. Log in and try to find the course.

A3.1.4 Scenario/task IV: Continue with previous work

You have only one chapter left of your course named "Techlog Wellbore - Beginner Tutorial". You wish to read up on the last few pages and then take the test, Since you are already logged in on the website, try to continue with the course, find the test and take it.

A3.2 Questionnaires and surveys

A3.2.1 Pre-testing survey

The pre-testing survey is available to see at the following link at docs.google.com in addition to the collected data.

A3.2.2 Post-task questions

Scenario/task I: Registration

- What kind of information would you like to see on the course page?

Scenario/task II: Finding a relevant course

- What kind of information would you like to see on the course page?

Scenario/task III: Login and re-using the platform

- Why did you choose to find your course the way you did?

Scenario/task IV: Continue previous work

- What do you think about how the different learning mediums were represented?
- Did you expect any additional features to appear in the course interface?

A3.2.3 Post-testing questions

- What do you think about the design and would you use this app?
- Was there anything you expected to find that was not there?
- What was unnecessary, if anything?
- What would you change?
- How would you rate the difficulty of tasks from 1-5?

A4 Prototype 2 - testing session

A4.1 Scenarios and Tasks

The following section contains the scenarios and tasks given to the testing participants in the usability test for Prototype 2.

A4.1.1 Scenario I: Revisiting a course

You are a student named Johannes who already has an account on OSDU Academy. Today you decide to complete a course you enrolled in a few days ago. You remember that it

was teaching a program called "Techlog Wellbore" on a beginner level. You already went through first three chapters and feel confident you will pass the certification test of the course.

Task: Sign in to your account, find the specified course, skip to the final chapter and try to pass the certification test.

Here is your account information:

- Email: j.gutenberg@email.com
- Password: password

A4.1.2 Scenario II: Apply for a petroleum geology course

You have been assigned to a new project at your job which requires basic knowledge about petroleum geology. You think a relevant course might be present in OSDU Academy, so you want to apply for one for later use.

Task: Find a relevant course, check its relevance and apply for it if suitable.

A4.1.3 Scenario III: Your certificate status

After you got certified in the Techlog Wellbore course you took earlier, you forgot to check if the certificate was registered. Now you want to check if you got a certificate and call it a day.

Task: Find a way to access your account status and then log out.

A4.1.4 Scenario IV: Taking the petroleum geology course

You are now ready to take the petroleum geology course you applied for earlier.

Task: Find the specified course and take it.

A4.2 Questionnaires and surveys

A4.2.1 Pre-testing survey

The pre-testing survey is available to see at the **following link** in addition to the collected data. It is the same as in the first prototype.

A4.2.2 Post-task questions

Scenario/task I: Revisiting a course

- How easy was it to find the course and complete it?
- What do you think about the layout of the certification test?

Scenario/task II: Apply for a petroleum geology course

- Do you think there was enough information about the course on its front page?

Scenario/task III: Your certificate status

- Would you describe the profile page as simple enough or too basic?

Scenario/task IV: Taking the petroleum geology course

- What do you think about how the different learning mediums were represented?
- Was there anything that surprised you in this task?

A4.2.3 Post-testing questions

- What do you think about the design and would you use this app?
- What do you think about the responsiveness of the website?
- What was unnecessary, if anything?
- What would you change?
- How would you rate the difficulty of tasks from 1-5?

A5 Prototype Testing Session Videos

Each testing session was recorded via Microsoft Teams, with two exceptions due to technical difficulties. This following section has a list of unlisted YouTube links to the video for each session. The videos are only accessible through these links and are not public to anyone else.

Prototype 1

1. <https://youtu.be/CnQuXrDSY6g>

2. <https://youtu.be/azl7SwZahJs>
3. <https://youtu.be/l3s0hgucc5Y>
4. <https://youtu.be/TRSIBgrGbqk>
5. <https://youtu.be/CrVfMBXTtEg>

Prototype 2

1. <https://youtu.be/isMHzbdty-M> (Divided into two parts!)
2. <https://youtu.be/3mw47c99Drc>

A6 Source code

You can find the complete source code at the provided GitHub. The project is registered under the MIT License.

Source code link: <https://github.com/AlanRostem/OSDUAcademy>

A7 Demo Video

A demo video about Prototype 2 is present on YouTube with this hyperlink: <https://youtu.be/sAtERX43kfo>