Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Johan August Østbye
Vemund Eggemoen

Investigating Student Interest and Time Management While Using a Cross-Platform Online Learning Platform

Master's thesis in Databases & Search, and Interaction Design, Game & Learning Technology
Supervisor: Boban Vesin
Co-supervisor: Michail Giannakos

May 2023

NTNU
Norwegian University of
Science and Technology

Johan August Østbye
Vemund Eggemoen

Investigating Student Interest and Time Management While Using a Cross-Platform Online Learning Platform

Master’s thesis in Databases & Search, and Interaction Design, Game & Learning Technology
Supervisor: Boban Vesin
Co-supervisor: Michail Giannakos
May 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Protus is an online learning platform for learning programming languages without previous experience. The system includes tutorials and activities to increase users coding knowledge within different languages and topics. Protus-next is a recreation of the original Protus. The new system was developed as part of this master thesis with a focus on user experience and optimization to support cross-platform functionality. Allowing the introduction of devices like mobile and tablets into e-learning, which could be better suited for studying.

This thesis investigated students' interest in using an online learning platform with cross-platform functionality to improve learning outcomes. To address the students' interest in this, the application Protus-next was developed and tested on NTNU students. Quantitative metadata was collected from user observation using an X-API implementation, in combination with a questionnaire to get insight into the students' interest. Results from the quantitative analysis enlighten an interest in a cross-platform online learning platform. The interest among students without technical background was overall greater than among those with technical background.

Students were observed while using the parts of Protus-next containing learning material. The results showed that they favored desktops over mobile and tablet devices when spending their time on Protus-next. Previous studies indicated better learning outcome on desktops when reading. This suggests that the reason for the students' time management is improved learning.

Sammendrag

Protus er en digital læreplatform for å lære programmering uten tidligere erfaring. Systemet inkluderer informasjon og oppgaver innen flere språk og temaer. Protus-next er en rekreasjon av den originale Protus. Det nye systemet var utviklet som en del av denne masteroppgaven med fokus på brukeropplevelse og optimalisering for å støtte multiplatform funksjonalitet. Dette muliggjør en introduksjon av enheter som mobil og nettbrett i elektronisk læring, som kan være bedre egnet for å studere.

Denne oppgaven forsket på studenters interesse i bruk av en digital læreplatform med multiplatform funksjonalitet for å forbedre læringsutbyttet. For å adressere studentenes interesse for dette, ble Protus-next applikasjonen utviklet og testet på NTNU studenter. Kvantitative metadata var samlet inn gjennom bruker observasjon ved bruk av en X-API implementasjon, i kombinasjon med en spørreundersøkelse for å få bedre innsikt i studenters interesse. Resultatene fra den kvantitative analysen ga nytt innblikk i en interesse for en digital læreplatform med multiplatform funksjonalitet. Interessen blant studenter uten teknisk bakgrunn var generelt større enn de med teknisk bakgrunn.

Stundentene ble observert mens de brukte delene av Protus-next som inneholdt læringsmateriale. Resultatet viste at studentene foretrakk å bruke PC ovenfor mobil og nettbrett under testingen av Protus-next. Med tanke på at tidligere forskning har indikert bedre lærings utbytte ved bruk av pc når en leser. Dette kan indikere at studentenes begrunnelse for ulik tidsbruk ligger i økt læringsutbytte.

Acknowledgement

This endeavor would not have been possible without our supervisors Associate Prof. Boban Vesin and Prof. Michail Giannakos, for guiding us with their valuable and constructive suggestions, motivation, and knowledge during the entire work of this research.

Special thanks to Johan's mom, Dr. Tone-Kari Knutsdatter Østbye for her valuable knowledge of Empirical research methodologies and paper writing.

We would like to extend thanks to all the participants, for their contribution. We are grateful to the numerous libraries across Australia for providing us with places to work.

We would like to thank Shara Smith for her hospitality in the last months of writing this thesis. Last we would like to thank our parents, siblings, aunts, uncles, grandparents, and friends for their patience and moral support even when we were half a world apart.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis explores how cross-platform technology in combination with data tracking can be used in online learning platforms to capture student behavior, and how this can be used to give new insight into online learning, and potentially increase the learning outcome.

Protus is an OLP, designed to contribute with programming courses and examples for students. The system was inspired by *MILE* (Multifunctional Integrated Learning Environment), an e-learning (electronic learning) tool designed for learning, teaching, and student assessment in basic programming courses [1]. The system was originally created by Vesin et al. [2]. It has since been through iterations to add additional features [3]. This thesis presents the authors' approach to re-designing, implementing, and testing a new iteration of the Online Learning Platform Protus. A new version of the system has been developed to support this research study and will from this point be referred to as Protus-next.

Initially, the motivation behind the study is presented, introducing the Online Learning Platform (OLP) Protus. Additional details related to both the background and the purpose of the system are then explained. Furthermore, the goal of this study is presented, followed by both research objectives and questions. The next chapter provides an overview of the research methods chosen, including the different stages like strategy, data generation methods, and data analysis. The contribution section enlightens to what degree the study contributes to scientific research. Lastly, a thesis structure is included to give insight into what to expect from each chapter.

1.1 Motivation

As a continuation of the work on Protus, Vesin et al.[2] wanted to apply personalization and recommendations to e-learning. A new version of the Protus system, Protus 2.0 was implemented to increase the effect of personalization with semantic data [4]. Later on, Vesin et al. created Protus 2.1 [5] where tagging was used to give users recommendations of concepts and resources.

The Protus system was originally designed for students with zero programming experience and provided courses for learning the essentials of the programming languages offered. However, the system's technology is outdated, and prohibits any future improvements. A rework of the system with updated technology would allow the system to be improved and updated. New technology for tracking has also been introduced since the last iteration of Protus, and introducing one of these would likely improve the system with valuable data collection.

1.2 Goal and Research Question

The goal of this study is to provide valuable insight into the students interest in cross-platform OLP and how students use OLP supporting cross-platform for learning outcome. Collecting metadata such as device type when navigating the application and performing different activities has the potential to uncover learning habits. Such information could be used to alter the outline of courses if there is room for improvement, resulting in better study behavoir and possibly increased learning outcomes.

The objective of this thesis is to develop a new version of the Protus system, Protus-next, with the same essential functionalities. However, the design and internals of the system will be updated to match today's standards with respect to web development. Protus-next will include a feature for easily adding new courses, as well as an implementation for tracking students across the site. The tracking data is added in hopes of providing valuable insight in study behaviour and how it can improve online learning. The system will allow students to browse between courses, choose topics, read explanations and examples, and lastly perform different activities. Furthermore, the system will support web responsiveness for all devices. However, the activities is supplied by a third-party, and the users get redirected to the third-party's website when accessing the activities. This causes the responsiveness to be out of scope for Protus-next.

This research aims to get insight into students interest in using an OLP designed for cross-platform. The students might use different devices deepening on purpose and location, which can provide valuable insight to improve online learning. Research questions for this study follows.

1. Is an online learning cross-platform of any interest to students.
2. How do students manage their time on a online learning platform for improved learning.

1.3 Research Method

Before performing the study, it was necessary to plan research methods and strategies. The planning was conducted in accordance with Oates principles [6]. Figure 1.1 gives an overview of possible research fundamentals used in this thesis. A *literature review* was conducted to gain knowledge of relevant topics such as online learning, mobile learning, and cross-platform, and to identify research gaps. This made the foundation for forming the research questions.

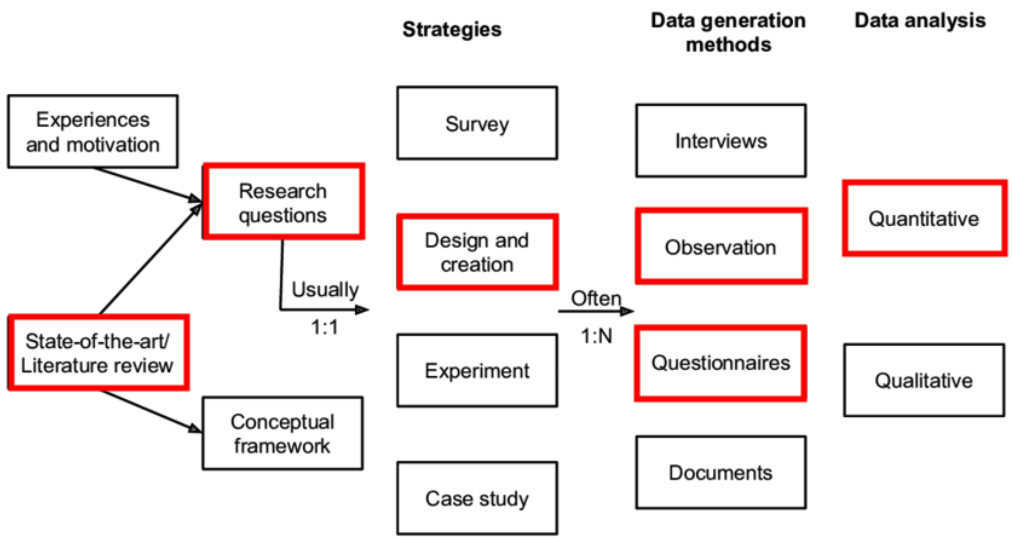

Figure 1.1: Oates's research model, (the highlighted boxes are used in this thesis).

1.3.1 Research Strategy

Design and creation also known as Design-Based Research (DBR), was the research strategy chosen from Oates's model [6, p. 108], as an IT application was to be planned, designed, and created. The design and creation strategy is a problem-solving approach, as an iterative process with five steps: awareness, suggestion, development, evaluation, and conclusion [6, p. 111]. These steps were considered throughout the process and some were followed as further discussed in Chapter 3. Importantly, the IT application provides research potential. However, the application itself is only a tool used in research [6, p. 110]. Other strategies like a survey and experiment were not considered relevant for this study.

1.3.2 Data Generation Methods

Observation was one of two data generation methods used to collect data. When users navigate the website, metadata are collected and stored. Importantly, all participants in this study were informed of data collection and had to accept the terms before being given access to the website. A collaboration with Sikt [7] ensured that all personal data was managed complying with Norwegian law [8] and the General Data Protection Regulation (GDPR).

This approach provides the opportunity to collect a lot of data with minimal effort. One of the core features of Protus-next is navigating the user to an external website where they are to perform activities. However, due to restrictions, the external website does not allow data collection on their website. Thus concerns arose as to how viable the data collected would be from the observations.

To ensure viable data, it was decided to include a *questionnaire* for participants to evaluate the artifact produced. [6, p. 116-117]. Thereby, it is possible to compare the participants' behavior from the observation with their evaluation from the questionnaire. The combination of the two data generation methods has the potential to uncover additional findings that would not be possible with one strategy or the two performed separately. Additional methods were considered but excluded as the chosen ones proved more suitable.

1.3.3 Data Analysis

As mentioned in the previous Section 1.3.2, automatically collecting data when participants are navigating the system does not require a lot of resources once the system is running. Combining the observations with a questionnaire can be used to look for patterns within the data. Thus, a quantitative analysis approach was chosen as the method aims to look for patterns and draw conclusions [6, p. 245-246]. Furthermore, the collected data are a combination of nominal and interval data. In order to easily make charts tables and graphs for visual aid, the participants were given little freedom for the input in the survey. Responses in the form of alternatives or scales can easily be visualized without a lot of resources. The quantitative data analysis is introduced in Chapter 4.3

1.4 Contribution

This study aims to address gaps in the research and contribute with new knowledge containing significant value. First, the study provides extended research to the already existing topic OLP. This thesis is among the first to consider how cross-platform can be used in online learning. Second, the experience API (see Section 2.1.5), provides the opportunity to track user metadata, including student behavior while participants are navigating the website. The theoretical lens in this study is that the experience API is able to detect student behavior on an OLP regarding cross-platform so teachers can adapt learning methods based on student study behavior and possibly increase learning outcomes. Additionally, this study can uncover new directions for research and discussions for future work.

1.5 Thesis Structure

This thesis contains seven chapters and is structured as follows:

- **Chapter 1: Introduction** - Introduces the thesis by explaining the problem, motivation, and contributions, in addition to building a foundation for the rest of the thesis.
- **Chapter 2: Background & Related Work** - Presents the research gap and findings of the literature review.
- **Chapter 3: Design an Implementation** - Explains the design phase of Protus-next, including the iterations and final product.
- **Chapter 4: Method** - Explains the research methodology and strategy utilized for this study.
- **Chapter 5: Results** - Presents the quantitative results and analysis.

- **Chapter 6: Discussion** - Discusses the findings, how it is related to the research gap, and compared to related studies.
- **Chapter 7: Future Work** Summarises the main findings and conclusion of the research and highlights approaches for future development and research.

Chapter 2

Background & Related Word

In this Chapter, a literature review has been performed to gain better knowledge on the relevant research topics such as e-learning, cross-platform, and OLP like Protus. The information from existing literature creates the foundation for this thesis.

2.1 Literature Review

A literature review was conducted to enlighten the state of the art of previous research and identify research gaps before any functionalities could be implemented into the Protus-next system. Google Scholar and ResearchGate were the main databases used for reviewing literature material. Both databases are acknowledged and contain peer-reviewed research. However, each study was reviewed independently to get an understanding of quality and trustworthiness.

Search terms: *Protus, online learning platform, X-API, cross-platform, e-learning, user experience* and, *mobile learning.*

The search terms were chosen to cover a large range of studies related to OLP. A large variety of studies provided insight into the state-of-the-art and research gap within different categories, which proved to be useful.

This section explores relevant studies in relation to the fields: *online learning platform, e-learning, and cross-platform.* It focuses on how to achieve sufficient learning outcomes, including quality education, motivation, and progression. And to optimize the system, user experience was explored to establish a strong foundation when developing Protus-next.

2.1.1 E-learning

E-learning is defined as "the acquisition of competencies, knowledge, and skills through electronic media, such as the internet or a company intranet." [9] became an important part of both learning and teaching tools for most educational institutes during the Covid pandemic. Students have different learning styles, and therefore different needs for how they learn best. Thus e-learning and learning outcomes vary among students. A study of 383 Chinese students by Wan et al. [10, p. 518] provided valuable insight into student behavior regarding learning outcomes. It was discovered that students with ICT (Information Communications Technology) habits were more pleased with e-learning and also had a higher learning outcome compared to others. Wan et al. explained that students with ICT skills and that usually search for information online, had higher virtual competence, which would help the individual during the process.

Regardless of the learning method, student behavior is a variable of the learning outcome. A study by Hussan and Sultan [11, p. 1900-1903] conducted on university students revealed confirming results. It was discovered that procrastination and lack of study habits had a negative effect on learning outcomes. Furthermore, the students tended to look for shortcuts, which do not necessarily mean low achievement on certain assignments. However, shortcuts can lead to a lack of learning outcomes, causing low achievements over time. Another study by Goda et al. [12, p. 77] supports Hussan's and Sultan's study research. Goda et al. investigated learning behavior in e-learning in relation to learning outcomes. Procrastination had a negative effect on academic learning, while proper learning habits had a very positive effect in relation to e-learning. E-learning depends on independence, meaning that students can spend different amounts of time on the same learning. Skilled students with study habits can thereby have an advantage as they can complete learning outcomes faster as they are not depending on others. To summarize, e-learning can have a positive or negative effect depending on human behavior and habits.

2.1.2 Mobile Learning

E-learning has been increasingly important during the last few years. However, it has been around for a couple of decades at a smaller scale. In a study by Ally from 2009 [13, p. 186-193], with a focus on *mobile learning* (small devices used for learning), the students were given mobile devices for e-learning when not in school. One of the main research goals was to reveal if mobile learning could be used during "lost" time (while on the bus etc.). The results were positive as the students were engaged with the device and thought it was fun. However, the study was performed in 2009 when it was less common to have a mobile device and therefore more fun since the participants were not used to it. Furthermore, the students were afraid of losing the device as it was not their own. Additionally, some issues regarding reception and connection came to light. These issues have probably affected the outcome of the study, which most likely would have be different if the study was done in 2023.

Measuring the learning outcome from mobile learning can be difficult. Kljunić and Vukovac [14] tried to estimate the learning outcome by performing a study measuring mobile habits and activities among university students. The goal was to uncover learning activities most suitable for mobile or tablet, giving the best learning outcome. Interestingly, activities with high scores are related to passive learning (viewing, browsing, downloading). This was arguable because the participants normally use their phones for similar non-learning activities.

Lin conducted a study [15] to test English reading learning on mobile devices and desktops. Interestingly the desktop group was outperformed in reading achievement and online activities. Also, the mobile group showed greater appreciation of the program. Another study [16] by Mesfer et al. measured skim reading effectiveness on mobile and desktops. The participants were divided into two groups that performed the experiment on separate devices and were tested remembering what they had read. From the results, the learning outcomes were better using desktops. However, an interesting part showed that participants used significantly less time on mobile devices compared to desktops. Even though the two studies proved different outcomes, they also performed different tasks, which might imply that some tasks are more suited for different devices. It should be noted that the former study was conducted as a ten-week experiment in an English class in a high school, whereas the other was random students asked to participate in a study. Thereby the participants in both studies have different motivations and ambitions.

A more recent study was performed by Abbasi et al. [17] aiming at capturing college students' perception of e-learning during the Covid lockdown. With a total of 382 responses, 76% of all students used mobile devices for e-learning. 72% were negative towards e-learning during the lockdown and would prefer face-to-face. This number is high, though other factors such as the length of the lockdown period would most likely have impacted their answer. Nevertheless, the study concludes that not everyone prefers e-learning, but would rather meet physically in a classroom.

Duolingo [18] is a free app used for learning languages. It is most popular on mobile devices, but available on all platforms and supports cross-platform. A study [19] from 2019 by Loewhen et al. investigated the learning effectiveness of using Duolingo on mobile devices. The results proved to have a positive motivating effect on learning effectiveness and improved knowledge. However, Duolingo uses gamification to boost motivation. Also, there were some concerns regarding more complex tasks to be performed by users, as Duolingo only contains simple translations of words and sentences. Nevertheless, the study proves how mobile learning can be used to increase learning effectiveness and knowledge.

The interest in mobile learning within e-learning has fluctuated a lot as the technology has evolved. It has gone from a high interest shown by Ally's study in 2009 [13] to a decreasing interest shown by Abbasi et al.[17] during the Covid-19 pandemic. However, a study by Barrero et al. [20] showed an increase from 15% to 5% in people working from home and an increased interest in working digitally from home. After investigating this topic, no studies related to students' interest or motivation for online learning after the pandemic were discovered.

2.1.3 User Experience & Optimization

Advanced technology has resulted in increased use of dashboards and websites in many industries including academic institutions. With new exciting possibilities, it is important not to forget the design principles when developing something new. All the design principles contribute to the overall UX (User Experience), which is the total effect including the emotional impact on a user when interacting with an interface [21, p. 5]. Several issues can lead to a negative UX which can affect the users' motivation and willingness to learn and use the product. Hence, it is important to optimize the system in order to obtain a satisfactory UX.

Responsive web design is an essential part of the UX, where the media changes based on the screen size. For instance, a desktop screen is bigger than mobile devices, which also are horizontal. Thus the media should look different because the user interacts with the device types differently. However, as it is the same website, one should develop a responsive web design that supports multiple screen sizes [22, p. 14-15]. Another principle influencing the UX is performance. Studies prove that loading time correlates to users' motivation to stay on the website. Optimizing for speed in the form of pre-loading, reducing requests, and using caching improves performance. [22, p. 17].

2.1.4 Cross-Platform

Cross-platform is defined as "able to be used with different types of desktop systems" [23]. In more detail, cross-platform systems should be able to run on different operating systems.

The technology of cross-platforms has been around for some time, but it is first in the last years that the technology has improved and solved many of the challenges. In 2013 Amatya et al. [24, p. 224-226] conducted a study where a survey was used to look at cross-platform mobile development. At the time of the study, there were many challenges, such as portability, security, user experience, and lack of graphic standards for handheld devices. However, there were also some opportunities like cross-platform mobile development that relies on web technologies for easy cross-platform development [24, p. 227], whereas developing a native application requires separate applications for each operating system [24, p. 220]. A lot has changed since 2013, but the fundamentals are the same, including the fact that a website can be accessed on any device regardless of the operating system. This is because a web browser such as Google Chrome is required to access applications on the web [25]. The biggest change over the last decade is indisputably the use of smart devices. Users now expect that applications can be used on several devices such as laptops, phones watches, and TVs. However, developing multiple applications is both expensive and time-consuming as the different devices have different operating systems. An alternative development approach is hybrid applications, which combine the advantages of both native and web development. The result is a hybrid application supporting cross-platforms[26].

2.1.5 X-API

X-API (Experience API or Tin Can API) is a technical specification for interoperability between learning technology products. The specification is meant to ease the implementation of a connection to a LRS (Learning Record Store) and ensure all implementations communicate with the same standard format of data described by the X-API specifications [27]. X-API is essentially used to track learning events wherever the users are located (digitally) and regardless of device [28].

AICC was the first API to store learning data followed by SCORM. SCORM is to this day the most used technical standard, but it is starting to get outdated compared to newer standards like X-API. SCORM was meant to record statements like completing a course with a score. Studies on how humans learn have discovered that learning comes from additional experiences other than just educational courses. Thereby X-API was invented, allowing many applications to connect to the same LMS and record learning experiences of different types [29].

Interestingly, the X-API has some limitations. Nouira et al. point out that the API has "Insufficiency of information dedicated to the track of the assessment result and lack of information dedicated to the assessment context" [30, p. 571]. This weakness is rather important as the learning behavior does not include the assessment process which is an extensive part of the learning process [30, p. 569]. An assessment process is an approach where teaching and learning create feedback which is used to increase students' performance. Additionally, the students become more involved and their confidence is improved as they can expect what and how to learn [31].

2.1.6 Time management

Time management is highly relevant in an online learning platform. A study by Uzir et al. [32] showed a relation between time management strategies and learning outcomes, where some time management strategies gave significantly increased learning outcomes. There exist many more studies similar to this enlightenment on how student time is used during a time period of a day or month, and how it relates to learning outcomes. However, the studies did not look into students time management while using the different parts of an Online Learning Platform.

2.1.7 Most Important Findings

The literature review provided valuable insight in relation to access to information. Wan [10] Hussan and Sultan [11] and Goda[12] all conclude that no matter the quality of an e-learning program, human habits and ICT experiences play a big part in the learning outcome. Furthermore, learning with mobile devices can both have a positive (Ally [13]) and negative (Kljunić [17]) effect. Studies from Lin [15] and Mesfer et al. [16] tried to measure learning outcomes by comparing mobile devices and desktops. Diverging methods proved to give different results. Also, Loewhen et al. [19] enlighten Duolingo's success on mobile devices. Last, Amatya [14] proved insight into activities giving the best learning outcome using mobile learning.

The interest in mobile learning within e-learning decreased during the pandemic, but Barrero discovered that after the pandemic, there was an increase in people interested to work from home. After the pandemic, no studies were discovered that measure student interest or motivation for an OLP, identifying a research gap. Uzir discovered a relation between time management strategies and learning outcomes when using an OLP. Both this and similar studies investigated student time management over a longer period. A research gap was discovered where the time management on different parts of an OLP has not been looked into.

Literature exploring UX and optimization explain how these topics affect user motivation and willingness to use the product. Studies related to cross-platform development [24] and [26] enlighten technical issues and opportunities. This is useful when creating Protus-next. Lastly, X-API provides knowledge regarding data tracking through student behavior. However, Nouira [30] points out that the assessment process is not included in the API. This limitation can be an issue as the assessment process is key in the learning process as discussed by Xhakaj [33].

To summarise, the literature review proved very useful to get an understanding of different aspects within an OLP and how to succeed in creating a system related to e-learning, dashboard, stakeholders, and technology such as cross-platform and X-API.

Chapter 3

Design and Implementation

This Chapter gives an overview of product evolution through the design process. First, the previous Protus system created by Boban, Ivanovic, and Budimac is described in Section 3.1. Second, an outline of functional requirements related to the product is discussed in Section 3.3, followed by the development tools chosen in this thesis in Section 3.4. Furthermore, the 4 + 1 architecture model is described in Section 3.5. Last, the development process including the iterations and the final user interface, is discussed in Section 3.6.

3.1 Protus

1, *Protus* is an OLP, designed to contribute with programming courses and examples for students. Protus-next created in this thesis is independent of the previous Protus versions, though aimed to keep most of the existing features. Data related to courses and exercises were fetched from external learning APIs and presented in the frontend. However, to perform the actual activities, users will be forwarded to an external website when clicking on either an exercise or a challenge. Furthermore, metadata related to users is collected when a user navigates the website and performs activities, as explained in Section 4.2.1.

3.2 Web Application as a Simulation

Cross-platform applications are defined as applications compatible with multiple operative system (OS) or platforms. This makes these applications *platform-agnostic*, meaning they do not discriminate on OS or platform. The time frame, capacity of the authors, and customer requirements of this thesis did not suffice for developing a cross-platform application. However, developing a cross-platform application does not require separate applications developed for each device and OS. For example, Capacitor [34] is a technology that allows a web application to be ported to mobile operating systems. Android devices can even download websites as applications [35], where a browser runs the application without showing the URL or features like tabs. IOS (iPhone OS) has also implemented its own version of this feature. As a result of all motioned above, Protus-next could be developed as a web application and simulate the functionalities of a cross-platform Online

Learning Platform.

Originally the web was only accessed through desktop computers and therefore only designed for desktop screen sizes. This type of design still covers 10% of websites [36], and the quality of mobile design is still improving. As cross-platform applications are designed for multiple devices and OS', a web application simulation would have to be designed responsively to be optimized for all devices and OS'. Protus-next therefore had to be fully responsive leading to the design method mobile-first, which will be described in Section 3.6.1

3.3 Functional Requirements

Functional requirements are used to capture the intended behavior by a system. Furthermore, the requirements state user needs and expectations [37, p. 2]. *Non-functional requirements* focus on a system's attributes, such as performance and security. Non-functional requirements are crucial for system success. However, quantifying can prove difficult as they synergize in competing processes [38, p. 421-422].

Developing the application with React was a requirement by the customer, as the language is widely used, and somewhat secured as a framework within web development. Thereby, web technology will likely not be an issue for further development like before. The customer gave valuable suggestions for the rest of the technology stack, though there were no limitations as long as the chosen technologies were well-known. Second, the Experience API (X-API) was required as it provides tracking opportunities, both utilized for students and teachers. A complete prioritzed list is displayed below in 3.3.

Prioritized Functional Requirements:

1. Users must be able to create an account and login.
2. Users must be able to choose between courses.
3. Users must be able to choose a chapter within the corresponding course.
4. Users must be given an explanation to the corresponding chapter.
5. Users must be able to choose activities based on the corresponding chapter.
6. Users should be able to filter activities.
7. Admins should be able to add new content.
8. Users should be able to access their profile page.
9. User progress should be visualized.
10. Teachers should be able to see all student data.
11. Users should be able to take a test both before and after they have completed a chapter to track their progress.

Prioritized Non-Functional Requirements:

1. The application must be written with React.
2. Metadata should be tracked with The Experience API (X-API)
3. Learning material should be fetched from external APIs
4. The design should be responsive and support smartphones, tablets, and desktops.
5. The application should have security to prevent non-authorized access.
6. The application should be expandable with both new content and future development

3.4 Development Tools

This Section will provide an overview of the tools used in this project in order to design and create a quality product. The reasoning for choosing the following tools will be described further in the sections respectively. Most of the tools were also known by the authors of this thesis from previous projects.

3.4.1 Figma

Figma [39] is a design tool for teams to experiment and test out ideas. The applications are interactable, which makes the program exceptionally great for high-fidelity prototype testing. One of Figma's many features is collaboration. Team members can easily work on the same project in real-time as everything is stored in the cloud. Figma was used in the design phase and enabled the brainstorming of ideas and resulted in a prototype where valuable feedback was received.

3.4.2 Notion

Notion [40] is a tool to organize work. Furthermore, several tools are integrated into Notion to make to-do lists, notes, custom workflow, and other features. Similar to Figma, collaboration within Notion is a powerful feature that allows users to get live updates from other team members. Notion was used to store all information on one platform. This includes documents, research papers and their corresponding notes, meeting notes, to-do lists and schedule a timeline.

3.4.3 GitHub

GitHub [41] is a software development platform used for version control and storing the source code in repositories online. The platform provides features such as 'issues' and 'branches' which allow team members to collaborate on the same code while having control of what others are doing. GitHub was used in the development phase where both students collaborated on the same code. Furthermore, the repositories provides a place to store code as a backup, in case of a local failure and loss of data. The repository for Protus-next is provided in Appendix A.1.

3.4.4 Next.js

Next.js [42] is a framework for building React Applications. The framework has quickly become one of the more popular frameworks. Next.js is technically a backend framework similar to Express, but with the capability of serving a react application as well. This has made the previous applications with separate frontend and backend applications outdated. Next.js also offers server-side rendering (SSR) offering faster loading as it eliminates sending unnecessary javascript files to the browser.

3.4.5 TypeScript

TypeScript [43] is a strongly typed language built on Javascript. The language has some advantages, like inferred types, compile-errors, and improved scalability. Extra types slightly increase the amount of code written, but the amount of time saved debugging via inferred types and compile-errors heavily outweighs the con. Nevertheless, TypeScript was used as it was preferred by the authors of this thesis. Additionally, JavaScript and TypeScript are interchangeable when the types are removed from TypeScript. Thereby, future development can be written with JavaScript without causing problems, making the application easily expandable.

3.4.6 Vercel

Vercel [44] is a platform used to deploy web applications. Additionally, the platform can be integrated into GitHub pipelines, meaning that building errors can easily be discovered before deploying changes. Vercel was used to deploy the Next.js application. Deploying the website was important in order to give participants access to the website during the study.

3.4.7 X-API

As mentioned in Section 2.1.5, X-API is used for tracking learning events [28]. An X-API consists of a LRS storing statements. A statement describes the actions an agent (user) did with or to an object. Statements are stored as follows:

Actor - Verb - Object.
I - Did - This.

Additionally, references to data objects can be created and thus it is possible to access information on all of the data in a record. Storing data in this structured format is part of what makes it possible for several applications to connect to the same LRS. X-API was integrated into the application (Section 3.6.3) following the documentation provided by the ADL initiative [27]. Then X-API was used for tracking and storing data during the usability testing, which was further used for discovering patterns regarding learning behavior as further discussed in Section 5.2.

3.4.8 NextAuth

NextAuth [45] is a JavaScript package used for authentication within applications. It provides the developer with many forms of authenticating a user, like credentials (email or username with password), email link, or the increasingly popular provider login with OIDC or OAuth methods [46]. NextAuth recommends the developer to use providers like Google for authentication, as the security risk of handling passwords is handled by Google with more resources for security. X-API documentation defines an authentication endpoint handling login as a provider but also defines this as optional for testing purposes. This fits well with NextAuth as the application can use credentials for testing, and if the authentication specification of X-API is developed, it can easily be switched to X-API as a provider using OAuth 1.0.

3.4.9 MongoDB

MongoDB [47] is a NoSQL (non-relational-database) database easily hosted both locally and on a server. The MongoDB organization even offers a fully cloud-managed instance with sharing for improved backup and recovery. One of the requirements specified in Section 3.3 was to use learning material fetched from external APIs. The learning API provided in this study had an average of 12 seconds latency, which means that it would take 12 seconds to load any relevant data. This performance was simply not acceptable and an alternative was needed. MongoDB was used to store both data from the learning API and additional needed data such as user credentials. Thus, MongoDB provides data when a user sends requests. Requests to the learning API is less frequent and is only used for updating the data stored in MongoDB. To summarize, MongoDB is mainly used to mitigate the performance issue caused by the learning API, and storing data like users and X-API statements.

3.4.10 Zod

Zod [48] is a TypeScript library used for declaring schemas and validating objects and values respectively. In Protus-next, Zod was used to validate inputs on all API routes and infer types from these schemas. The use of this library proved to be of immense value as the learning API does not return a standard JSON response preferred by web frameworks.

3.4.11 TRPC

TRPC [49] is a typescript package for implementing an RPC (remote procedure call) API [50] in applications. TRPC also uses Zod as mentioned in Section 3.4.10 to validate the types of inputs. The package allows for client-side loading of data with typed responses and functionality for handling React's rerendering.

3.4.12 Prisma

Prisma [51] is an open-source ORM (Object Relational Mapping) for Node.js and TypeScript. In the project, Prisma was used to covert data going between a relational database (MongoDB) and an object-oriented programming language (TypeScript).

3.4.13 Sanity

Sanity Studio [52] is a real-time collaborative application for content creation. The tool allows users to add content without being a developer, exactly the tool needed if a teacher needs to add, remove or update learning material. Sanity was used to implement the content within the course and chapter pages and with the thought of teachers being able to use it in the future.

3.4.14 Excel

Excel [53] is a tool used for organizing, formatting, and calculating data in a spreadsheet. The tool was used to keep track of the codes that had been given to the participants for testing. More information regarding codes can be found in Section 4.1.3 under User Testing. Additionally, Excel was used to analyze the result from the questionnaire as the data could be exported in Excel format. Data from both the user testing and questionnaire were visualized using graphs and are displayed in Chapter 5.

3.4.15 Storybook

Storybook [54] is a frontend workshop to create UI pages and components. One of the advantages is that Storybook runs outside the application, which is great for developing in isolation. As TypeScript 3.4.5 with its strict 'type' system was used, developing was still possible even though TypeScript gave errors within other components. This advantage makes Storybook suitable for component-driven programming [54].

3.4.16 Sass

Sass [55] is a stylesheet programming language that extends CSS (Cascading Style Sheets) with features that simplifies DOM (The Document Object Model, a programming API for HTML and XML documents) selection and allows developers to more easily create complex styling. Sass code is not compatible with HTML as is, but is complied to CSS when rendered. Some of the features are variables, nesting, and mixins which allow developers to reuse code. Sass was used as it provides additional features that would make styling more productive.

3.5 Architecture

Architecture 4 + 1 has been used as it offers a broader overview of different aspects, which are described in further detail in Section 3.5.1. The deployment of the Protus-next is discussed in Section 3.5.2.

3.5.1 4 + 1 Architecture Model

The 4 + 1 architecture model provides visual documentation of the system during the development process. The architecture consists of several views that together fully cover all parts of the system. Each view can be inspected individually and it allows for greater detail within smaller parts of the system without being distracted by irrelevant information. Furthermore, the architecture is used to detect either missing functionality or flaws [56]. The model consists of four views: logical, process, development, and physical view. In addition, a use case view is included, which represents the system's functionality from an outsider's perspective [56]. Each view is concurrent and possesses its perspective of the systems architecture.

The *logical view*, visualized as a state diagram in Figure 3.1 displays the different components comprising the frontend and the interactions between them. A functional design method was used as it suited the project better, compared to an object-oriented design. As a result, a state diagram was used to visualize the application's logic [56]. The diagram explains the global state and how it's utilized by different components. Last, the diagram gives an overview of components that can be re-rendered when the global state is updated by other components.

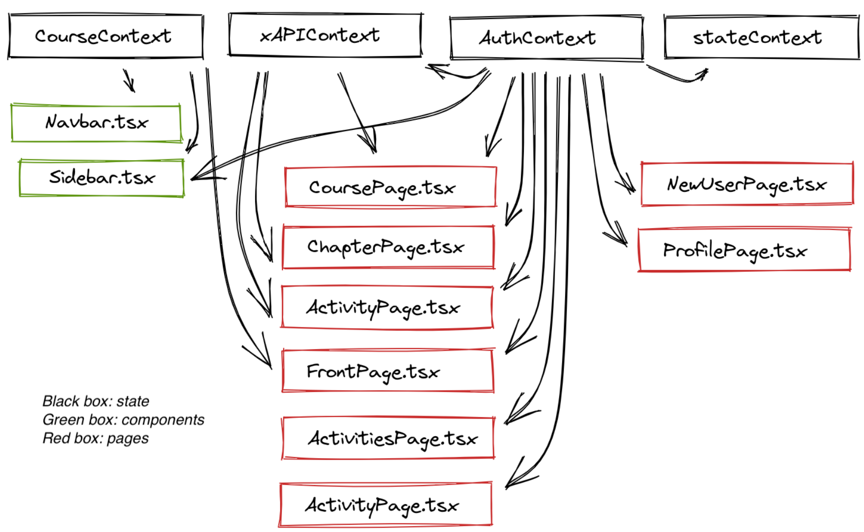

Figure 3.1: Global state flow visualizing which components in the front-end make use of the global state.

The *process view* describes the communication between the processes within the system [56]. A communication diagram can be viewed in Figure 3.2, which explains the process where queries from the frontend are returned with data from the database.

The *development view* illustrates the system from the perspective of a developer.

This view describes components including class libraries, sub-systems, and packages. Furthermore, the development view provides an overview of the different layers, as well as displays the components as building blocks, which makes the view easy to understand [56]. A component diagram is used to describe this view and contains components with their corresponding props, as seen in Figure 3.3.

The *physical view* describes the system from the perspective of a system engineer. This view both maps software artifacts onto virtual machines/hardware that hosts them and models the execution environment of the system [56]. Figure 3.4 displays a Unified Modeling Language (UML) deployment diagram, which is used to visualize the physical view of a system.

Last, the *use case view* shows the functionality of the system and provides the perspective of a user. This view covers all user goals and scenarios, which is helpful when planning and structuring the different system functionalities in the other views, hence the 4 + 1 architecture [56]. Figure 3.5 visualizes this with a UML use case diagram.

3.5.2 Deployment

Discovering how the observation would be conducted, where the X-API would be used for logging data as users navigate the website (see Chapter 4). It was decided to deploy the application as collecting quantitative data would be more efficient if participants could access the website by themselves. The application was deployed to Vercel as explained in Section 3.4.6. Furthermore, it was decided to use MongoDB to host the database in MongDB´s cloud service as discussed in Section 3.4.9. Having both the application and the database in the cloud allowed users to access the application and view the content. Also, as the participants navigated the website, their metadata was stored in the database. This Section has described the deployment of Protus-next specified in the *physical view* in Section 3.5.1.

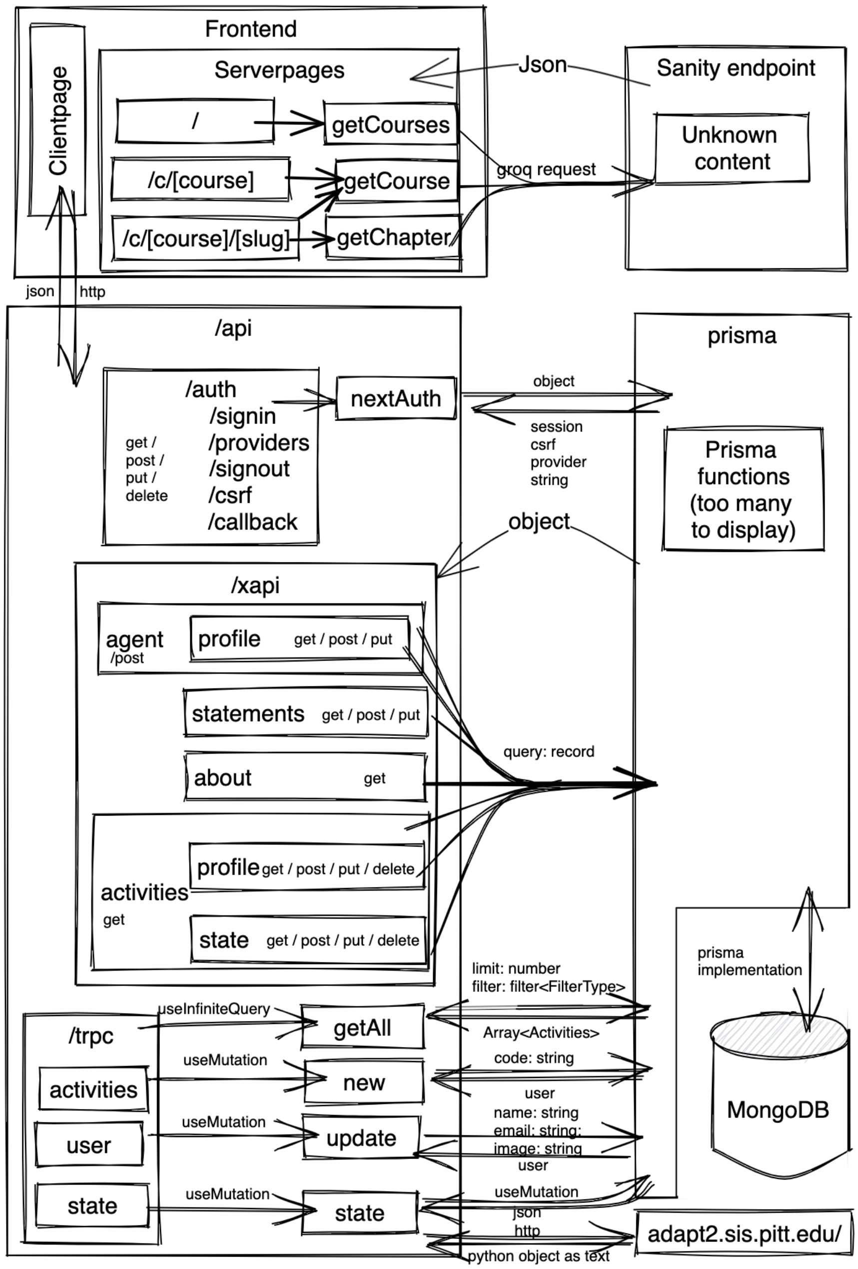

Figure 3.2: Communication diagram displaying how HTTP requests are propagated throughout the system.

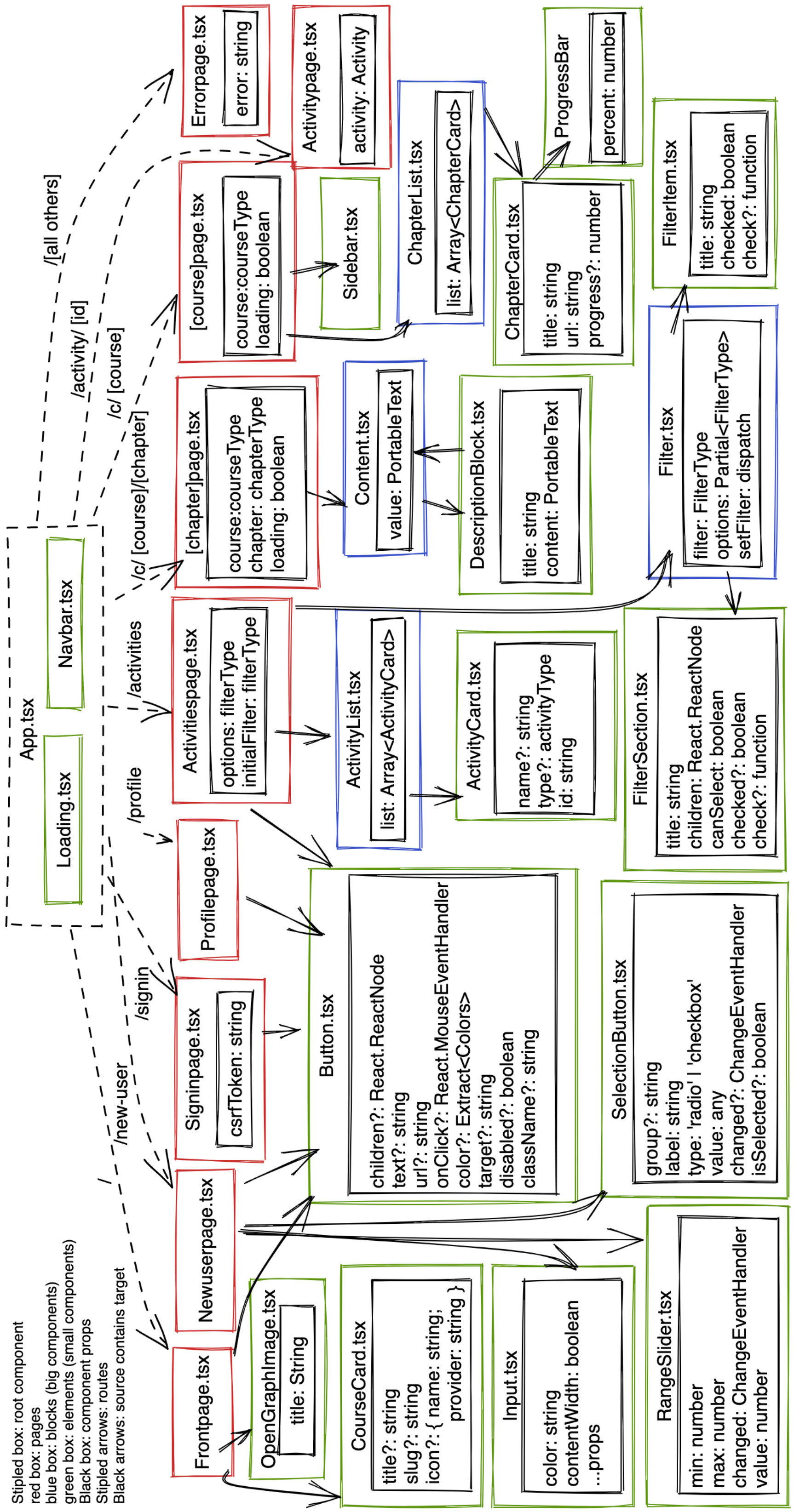

Figure 3.3: Component diagram showing the different input parameters for each component and how the components are related to each other.

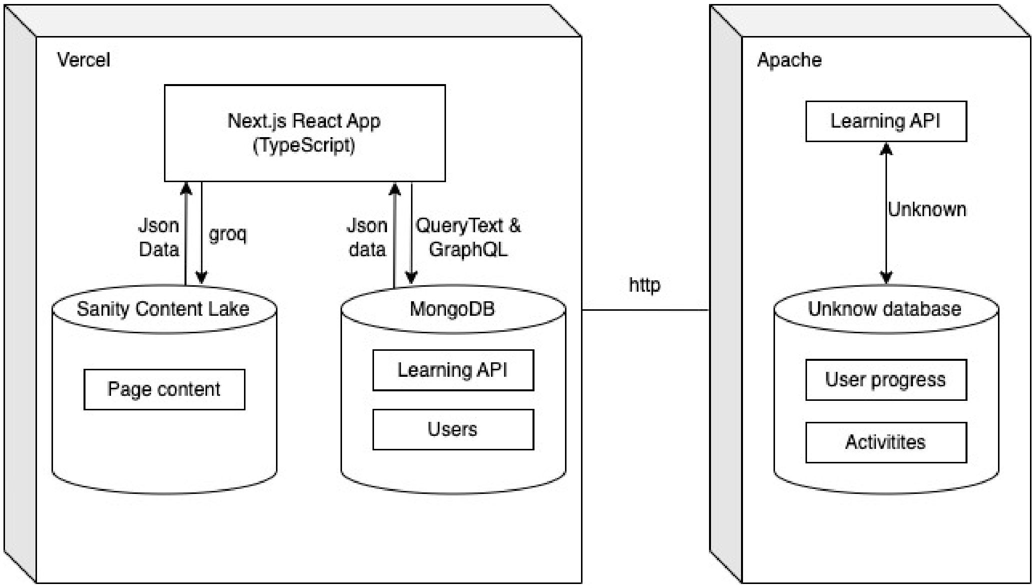

Figure 3.4: Deployment diagram showing the physical relationship between the front-end and back-end.

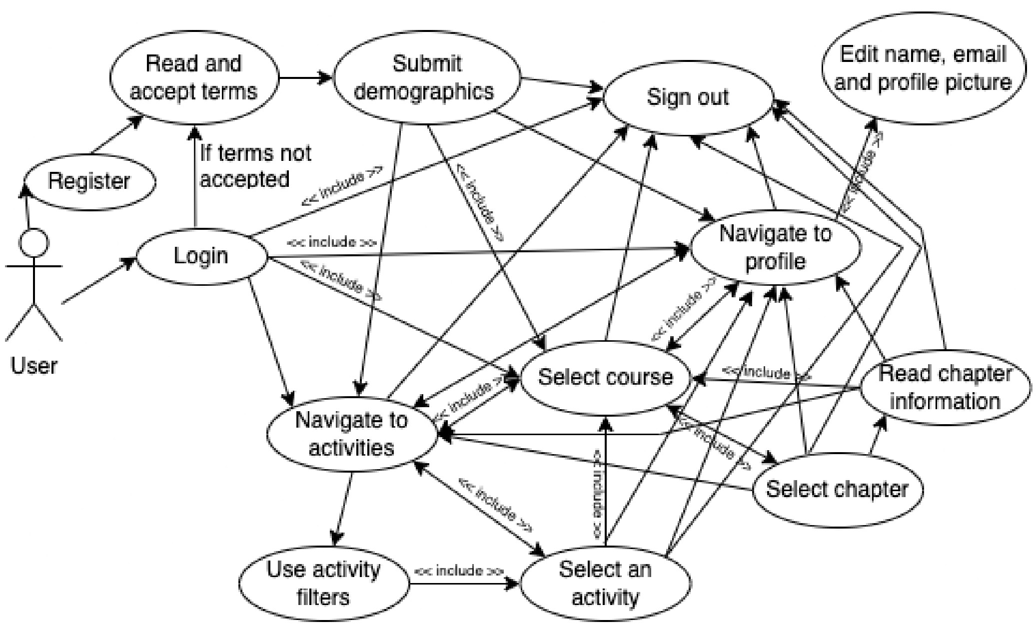

Figure 3.5: Use case diagram displaying the different use cases of the application, as well as what functionality is dependent on each other.

3.6 Development Process

After both performing a literature review and declaring any functional requirements, it was time to start the implementation. In total four iterations were performed in their respective Sections 3.6.1 through 3.6.4. Though, as this thesis aims to renew the old Protus system, only two design iterations were performed as a foundation for the design was already established. Nevertheless, some functionalities have been improved from the previous version, as these would enhance the user experience. Furthermore, testing of the system is described in Section 3.6.5. Last, the final user interface is presented in Section 3.6.6.

3.6.1 Iteration 1 - Prototyping

When creating a new system based on a previous version, it is important to get an understanding of the functionalities and purpose of the new system. This allows for better planning and overview when starting to design the new system and further into the code implementation. Altering or implementing new functions is both expensive and time-consuming in the actual implementation compared to the design process [57]. A prototype is an unfinished product and a crucial part of the design process. Creating a prototype helps to test out an idea or a proof of concept before the finished product reaches production. Depending on the company or project size, the product may undergo several prototypes before reaching production if it even comes that far [57].

In the design process, two prototypes are often used; low- and high-fidelity prototypes, where fidelity is “the level of detail that content is rendered in the interface” [58, p. 86]. The low fidelity is often sketched in the form of paper or digital drawings. These prototypes are made fast, and rapid changes can be done. However, the drawback is the lack of complexity. With a high-fidelity prototype, both complexity and advanced graphics including animations are possible, though it’s more time-consuming [59, p. 79-82]. In return, the high-fidelity looks more like a finished product as most of the design and interaction are in place. In software design, many tools can be used for high-fidelity, like Figma. This tool can be used for creating an app prototype with navigation and design. To summarize, the two prototypes differ and are used for different purposes.

Before implementing the actual code, creating a prototype was necessary to improve the already existing interface. Thereby, the first iteration was a high-fidelity prototype, and not a low-fidelity, as only a handful of changes were necessary and a low-fidelity would arguably be unnecessary. Figma was used in this thesis as the authors had experience with the program from previous projects. Additionally, Figma provided all the necessary functionalities.

Color Palette

Upon starting designing, the colors on the website became a topic. The previous versions had various color designs as seen in Figure 3.7. However, using a more standard dark-mode theme would arguably look more professional and familiar to other websites. Browsing the internet, a blue tint dark-mode color palette was found on the website ColorHunt[60]. Furthermore, inspiration was taken, and a color palette was created as the students wanted variations of light grey and slightly different blue tones. Using a color palette gives a standardization to the theme, in addition to setting the theme. Also, a palette is used for mixing the colors and creating new ones [61]. By adding a gradient filter on some buttons, new colors were created. The color palette is displayed in Figure 3.6. The following colors are:

- **Black** (#1D1D1D) was used as the main background color within the application. In addition, it was used as font color both on the gradient buttons and on input fields.
- **Dark-grey** (#292929) was used as background color both on the navigation bar and in content field blocks, such as ”Deklarasjon” in Figure 3.10.
- **Grey** (#363636) was used as background color within the activity cards, displayed in Figure 3.11.
- **Grey2** (#4B4B4B) was used when hovering an activity card.
- **Light-grey** (#676767) was used as the background color both for the icons in the activity cards and for the search input field seen in Figure 3.11.
- **Purple** (#A500C0) was used as font color in activity cards to indicate the type ”Exercise”.
- **Red** (#FF3D00) was used as font color in activity cards to indicate the type ”Challenge”.
- **Blue gradient** (#008BC7 #76FFFF) was used as background for buttons, blocks, and tags, displayed in Figures 3.10 and 3.11.
- **White** (#D9D9D9) was used as the standard font color within the application. Furthermore, it was used as the background color on login fields as displayed in Figure 3.9.

Mobile First

”Mobile first” is a method to design and develop an application with content in mind. A mobile screen can fit significantly less than a desktop, and designing the content for mobile-first forces one to evaluate what content is essential for the user and what content is unnecessary. If done the other way around, the content of the desktop will have a hard time fitting into the mobile screen. Showing content on a desktop and not mobile is not an optimal solution either. As of 2016, mobile surpassed desktop in internet usage, meaning if the focus would be on designing

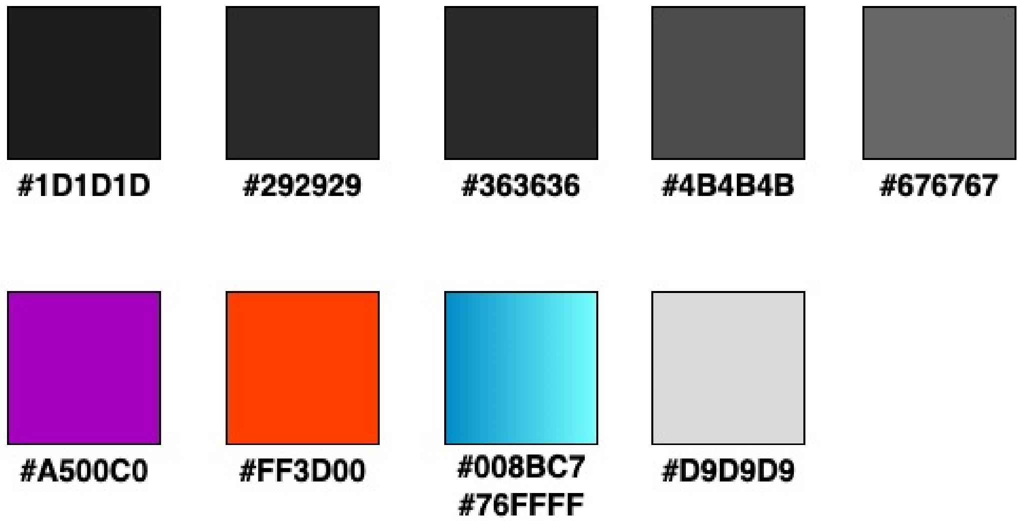

Figure 3.6: Color palette

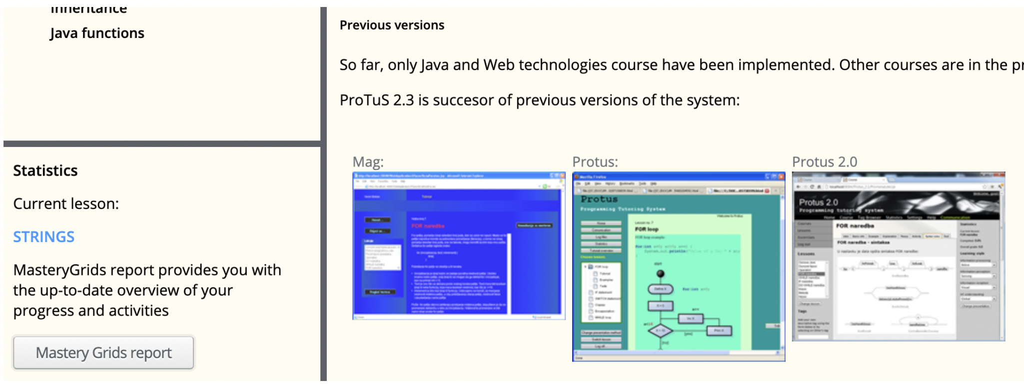

Figure 3.7: Protus old system color design

for the biggest user base first, then mobile first would also be the correct way. The use of a desktop is preferred for coding, but late trends have started bringing smaller devices into learning activities [62, p.19].

Layout Changes

The layout of the previous Protus system (displayed in Figure 3.8) had room for improvement. Table 3.1 provides an overview of all the design layout changes. The following paragraphs will discuss the changes in detail.

Id	Name	Description
1	Course layout	Changes to the course topic layout. How activities etc. are displayed.
2	Navbar	Remove buttons and reduce size.
3	Sidebar	Change the sidebar.
4	Hard lines	Remove hard lines etc. separating the left bar and main content.

Table 3.1: Design layout changes

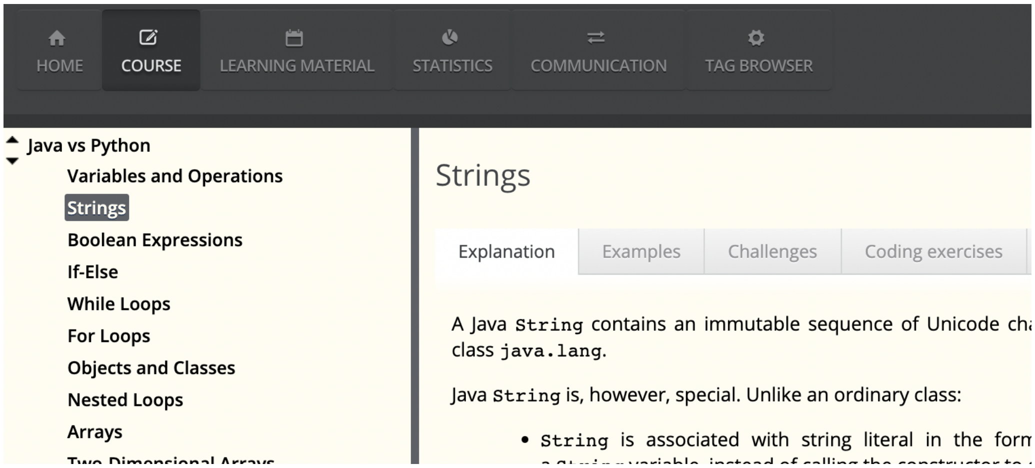

Figure 3.8: Protus old system

Course Layout

First, the old system has 4 tabs when navigated to a course chapter, which renders different content on the page as seen in Figure 3.7. Removing the tabs and including filters will allow the user to have more control over the displayed content. This means that it's possible to have content from different tabs displayed at the same time. The inspiration for this change comes from W3schools [63]. The changes can be viewed in Figure 3.10.

Navigation Bar

Second, the navigation bar is too tall with unnecessary buttons and icons. The 'Home' button, was replaced by a 'Protus' name. The only other button was the course alternative, but the text was instead displayed as a course, with a drop-down button to change course. Learning materials were moved to the corresponding course. Both the 'Statistics' and 'Communication' buttons were moved to the profile page. Removing the icons as there are no buttons, allows the height to be reduced. The changes can be viewed in Figure 3.10.

Side Bar

The main complaint about the previous sidebar is that it was not easily closable as the user had to click the profile icon to get the close option. This action lacks affordance and in order to fix the issue, a close button has been implemented at the top of the sidebar as seen in Figure 3.10. When closed, a button indicating to display of the sidebar is still visible. Furthermore, the highlighted course topic has an underline instead of changing the background color. Last, the learning material has been added to the bottom part of the sidebar, separated from the course topics by a horizontal line.

Hard Lines

The hard lines separating the different sections such as the main content, sidebar, and statistics are unpleasant to the eye. As a solution, the sidebar has a bright shadow separating itself from the main content. Additionally, the statistics are moved to the profile page as specified in the 3.6.1 Course Layout paragraph. The changes can be viewed in Figure 3.10.

Figma Prototype

Using Figma, eight screens (one for mobile and one for desktop) were created and are displayed in Figures 3.9, 3.10 and 3.11. The screens were created with the latest Protus version [3] as a template, combined with the respective color palette and ”Mobile first” design described earlier in this Section.

In the login screen in Figure 3.9, the user should have the additional option to use Google when logging in. Google login provides both easy access for the user and handles all of the security. Also, the frontpage is a new page that gives information about Protus before logging in. In earlier versions, the information was displayed after the user had logged in, which is not great for potential new users wanting information about the system.

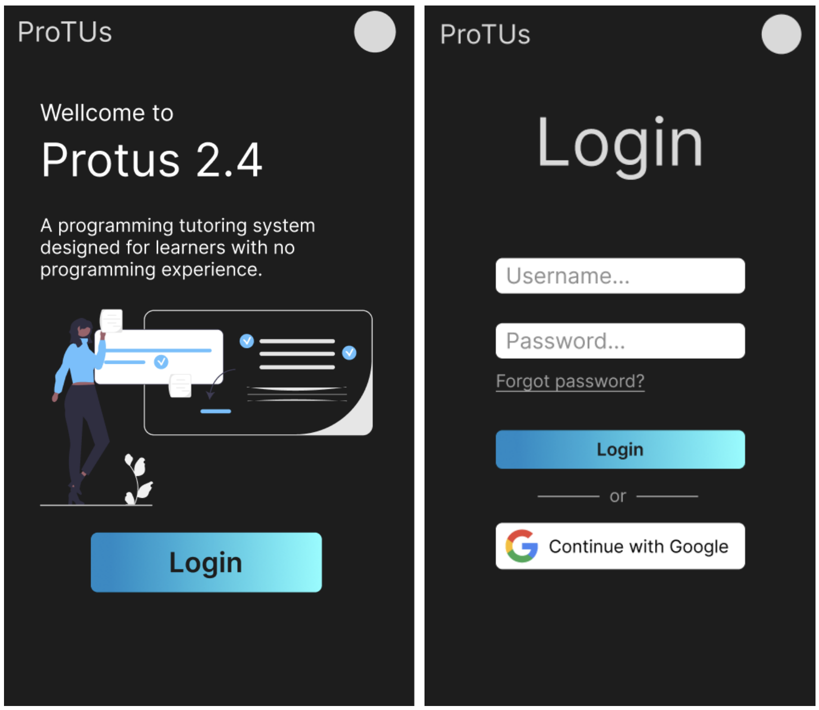

Figure 3.9: New design (mobile version) using Figma, left: frontpage, right: login screen.

Some changes have been made to the course screen in Figure 3.10. First, all previous tabs have been removed and replaced why one button ”Activities”. Clicking the button will take the user to the activity screen seen in Figure 3.11. The remaining content on the course screen is an explanation of the current topic. Furthermore, some global components have been undergoing a rework. All the buttons on the navigation bar have been removed. The buttons were either unnecessary or could be moved to another location. Next to the Protus title, the current course is displayed with a dropdown to choose another course. Furthermore, the sidebar can be opened and closed with buttons easily visible to the user. Additionally, the sidebar has a shadow which indicates it’s not part of the main screen and can be opened or closed.

The activity screen can be seen in Figure 3.11. Both ”exercises” and ”challenges” can be found on this screen. The user has the option to use the search field for a specific activity or use the filter menu. The activity card has both an icon and

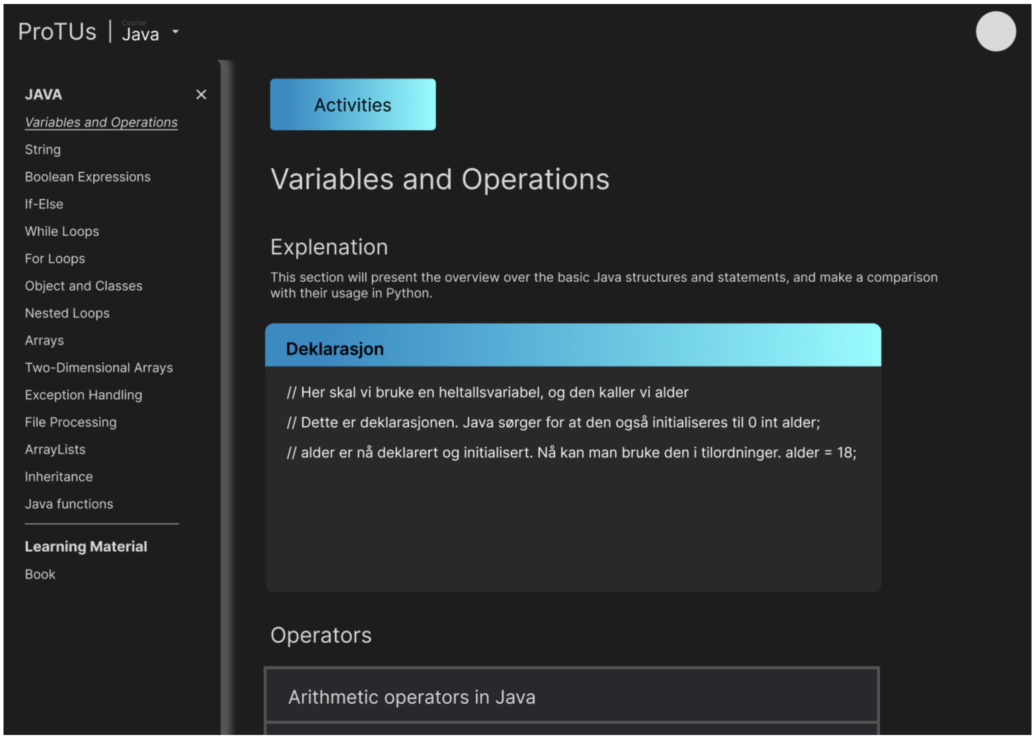

Figure 3.10: New design using Figma, course screen

text color based on the activity type.

Usability testing was considered, though when comparing workload to value, the authors of this thesis did not feel it was necessary to conduct testing as previous versions of the system existed. Nevertheless, in total six fellow informatics students were asked for their opinion, in case of potential improvements. Overall the feedback was positive where where the next iteration could focus on other aspects, though a few minor improvements were suggested.

3.6.2 Iteration 2

After creating the digital prototype with Figma, a short iteration was performed to improve the prototype. First, the "course" text above the current course in the navigation bar (displayed in Figure 3.10, was difficult to read. Second, the shadow on the sidebar managed to separate itself from the main content. However, as it has the same color as the navigation bar, there was no indicator of where the different components ended. Furthermore, it was pointed out that a registration form was missing, if the user did not use Google to log in.

To improve the prototype, the "course" text was made more visible, and additional shadows on the sidebar were added to separate itself from other components. Last, a register screen was implemented to support new users. Even though a complete usability testing was not conducted, the findings prove the importance of usability testing. By monitoring the user, one can identify what they are doing, discover their intentions, figuring out what works and what does not. Essentially, the focus shifts from the system to the user need which is crucial when creating a system [64, p. 9-10].

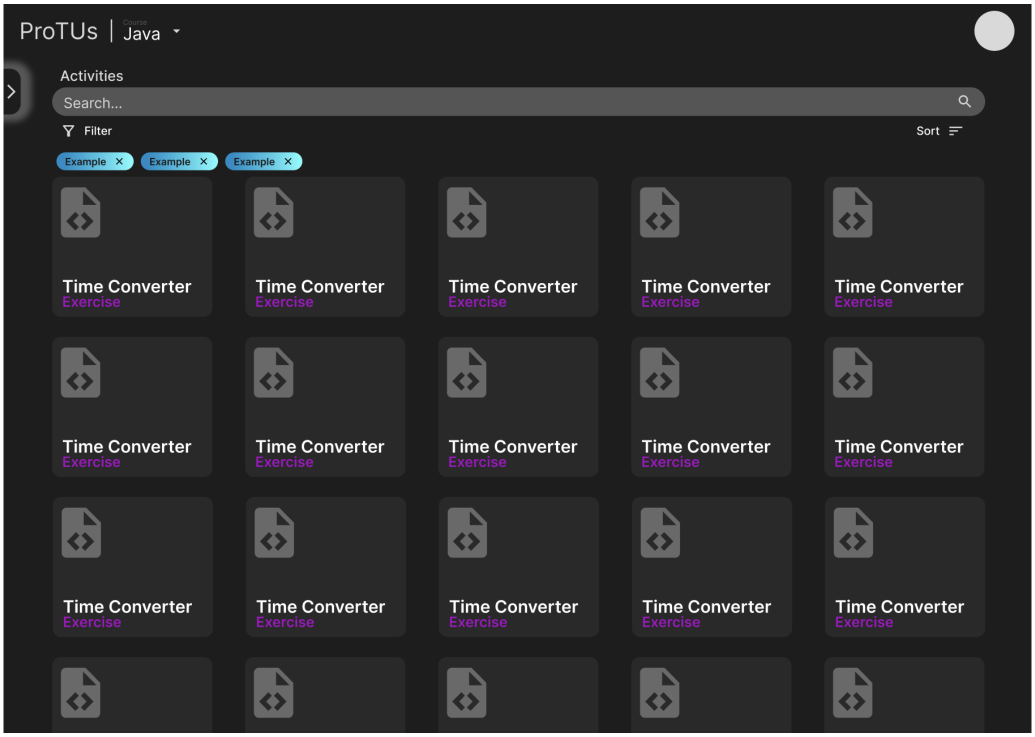

Figure 3.11: New design using Figma, activities screen

3.6.3 Iteration 3

After two successful design iterations, it was time to incorporate the digital prototype into software. Next.js [42] was used for development, which is elaborated in detail in Section 3.4.4. Additionally, Storybook in Section 3.4.15 and Sass in Section 3.4.16 were used for creating and styling both components and pages without relying on the entire application. It was also in this iteration the X-API (see Section 3.4.7) integration was implemented, though tracking of data had to wait as it was depending on the learning API which was not yet implemented. Finally, the website was deployed to Vercel as discussed in Section3.5.2, which was required for the research strategy in order to gather data.

Changes

As discussed in the Figma prototype paragraph, Google login (see Section 3.6.1) was preferred as all security would be handled by Google. However, due to X-API's limitation as discussed in Section 3.4.8 Google authentication was no longer an alternative and the idea had to be discarded. Second, both the sidebar design and accessibility had minor changes. While using a large screen it was no longer possible to close the sidebar, but it would stay open. Additionally, the shadow properties were replaced by a line separating the content from the sidebar as this proved a better solution. A sign-out button was implemented and can be found in the top right of the navigation bar. A completed list of all changes is displayed in Table 3.2.

Id	Name	Description
1	Google	Dropped login with Google
2	Sidebar	Changed Sidebar interactivity and design
3	Sign Out	Sign out functionality

Table 3.2: Changes from iteration 2

Challenges

Furthermore, it was in this iteration challenges related to the learning API were discovered. The API did not return data with a standard JSON format response, which is preferred by modern web frameworks. Instead, the data were a Java object with inconsistent use of tags and unrecognized functions by JSON. This proved it difficult to parse the data to JSON format. Additionally, it was discovered that the API had a bad performance with an average of 12 seconds response time. The solution was to intermediate storage of the data with MongoDB as discussed in Section 3.4.9.

3.6.4 Iteration 4

Contemplating the upcoming data generation, a page for new users was required to obtain the unique code given to participants, that would be used for data tracking. Also, the new page provided the opportunity to gather relevant demographic information about the participants. To increase visibility, it was decided to include the different chapters on the ”course” page and not only present them in the sidebar. Furthermore, charts that could be used for both visualizing user data and learning progression were created using Nivo. Nivo was initially implemented as a tool to generate graphs in the system for users. However, these graphs were later discarded as it was deemed unnecessary for the observation, and data visualized from the observation was instead visualized through Excel. Some of the intended Nivo graphs were user-specific graphs visualizing activities completed, and a teacher specific with the average completed in the class. Next, a profile page was implemented where user can change their account information. Furthermore, Sanity (see Section 3.4.13) was used to create the needed learning content that was not provided by the learning API, such as the explanation of chapters. Finally, state management was implemented and X-API tracking was working accordingly. It was decided to implement iframes, to display the activity from the external learning API page to Protus-next. This decision was to reduce user confusion when they are redirected to another website.

Changes

The ”activities” button within a specific chapter page (displayed in Figure 3.10) was moved to the navigation bar as it seemed more fitting. The change can be seen in Figure 3.13.

Challenges

The learning API provides activities of different types. However, the types use different protocols for accessing information. The exercise activities use *https* while the challenge activities use *http*. This proved difficult as iframes do not support non-encrypted protocols (*http*). The solution was to redirect the user to the external API without an iframe if the activity was a challenge.

3.6.5 Testing

Testing was implemented differently for each implementation of the system. The frontend and its components were tested after development with image snapshots (automatic screenshot testing) to ensure further development didn't affect what was already developed. Conversely, X-API was developed with test-driven development (TDD) based on the X-API specification (see Section 3.4.7).

3.6.6 Final User Interface

After several successful iterations through the design phase, the application was ready for testing. The final interface consists of 12 screens displayed in Figures 3.7 through 3.18. Detailed images of all screens including both mobile and desktop can be found in appendix B. The next paragraphs will discuss each screen in further detail.

Frontpage

This page has two variants depending if the user is authenticated. Upon navigating to the frontpage, the user will get information regarding what Protus is and have to press the ”Login” button for navigation to the sign-in page. If the user is already authenticated, the frontpage will consist of cards will available courses which will navigate them to the Course page. A course could be put as standard, however, it would affect user decision-making. Thereby, no course is set to default in the navigation bar. Next to the course button is the ”activities” button that upon clicking will navigate the user to the activities page. At the top right, the user can press the profile icon to either navigate to the profile page or sign out. All icons in the navigation bar are first visible after the user has been authenticated. The frontpage variants can be viewed in Figures 3.12 and 3.13.

Sign-in Page

The user needs to use an email and password to log in to the system. Password recovery is implemented and can be accessed by pressing the ”Forgot password?” link. Last, the user can also register a new account by pressing the ”Register” button. If the user already exists, pressing the login button will navigate to the frontpage if the corresponding credentials are provided. If a new user registers, logging in will navigate the user to the user-new page. The sign-in pages can be seen in Figures 3.14 and 3.15.

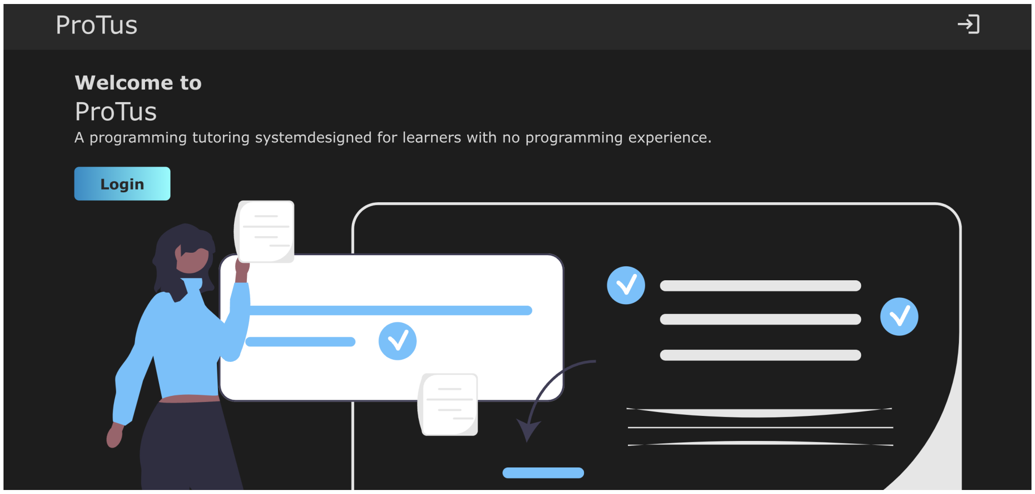

Figure 3.12: Frontpage not authenticated

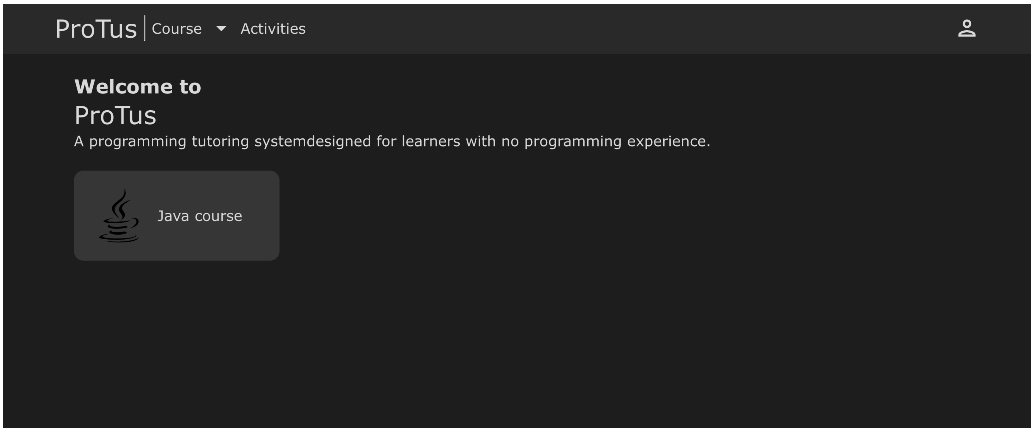

Figure 3.13: Frontpage while authenticated

User-new page

A user will only reach this page once if never logged in before. General information regarding the study and data tracking are presented and the user has to press the ”Accept” button to agree to the terms. Following the information, the user has to enter a code given to them beforehand that will be used for tracking. Additionally, some relevant information has to be inputted that will be used for analyzing the data. At any time there is the option to go back to the general information using the ”Back button”. The ”Send in” button will send the data to the database and navigate the user to the frontpage. The variants of this page are displayed in Figures 3.16 and 3.16.

Course Page

The user gets presented with chapters corresponding to a course in the form of cards. Clicking a chapter card will navigate the user to the chapter page. Additionally, the cards display both available and completed activities within each chapter. This gives the user a better overview of learning progression. The course page can be viewed in Figure 3.18.

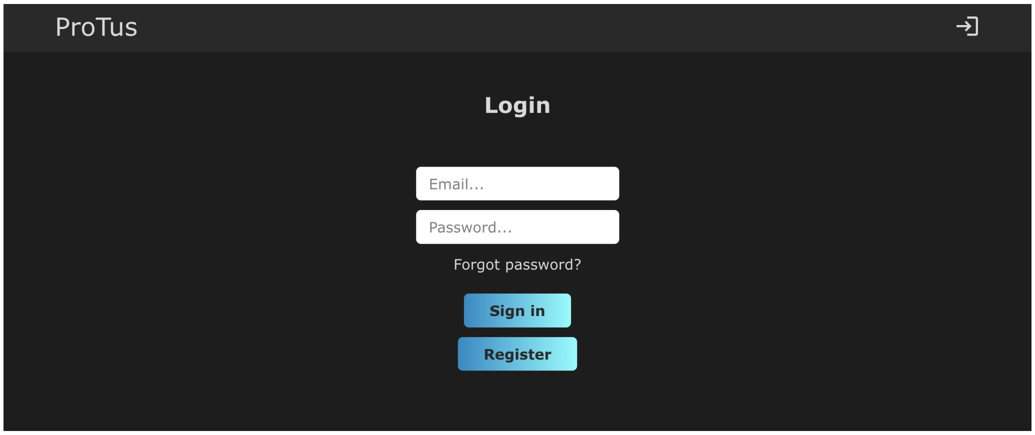

Figure 3.14: sign in page

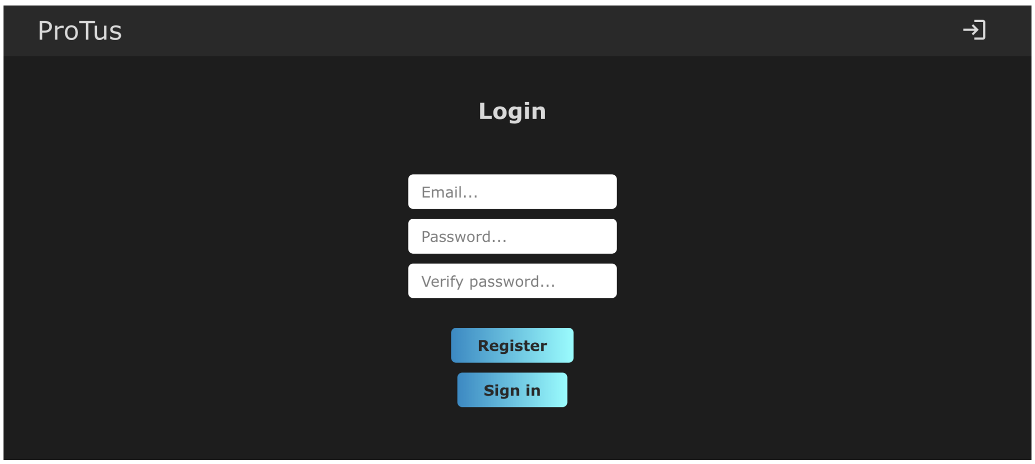

Figure 3.15: Register page

Chapter Page

Upon navigating to the chapter page, an explanation of the current chapter is presented, followed by examples in frames. A course can be changed by pressing the arrow-down button next to the course in the navigation bar. The chapter can be changed by clicking on the presented topics in the sidebar. The sidebar will behave depending on screen size. On mobile phones (small screens), an arrow on the top left indicating a hamburger menu can be pressed to open the sidebar. The sidebar will be displayed over the current content and cover the entire screen. It can be closed by pressing the X icon on the top right. While using tablets (medium screens), the sidebar can be opened and closed similar to using a small screen. However, it will only overlay a small part of the screen. If the application is used with a desktop (large screen), the sidebar will be indefinitely open. The chapter page can be viewed in Figure 3.19.

Activities Page

All available activities corresponding to a chapter will be presented as cards. Each card has a unique name together with an icon and an activity-type text

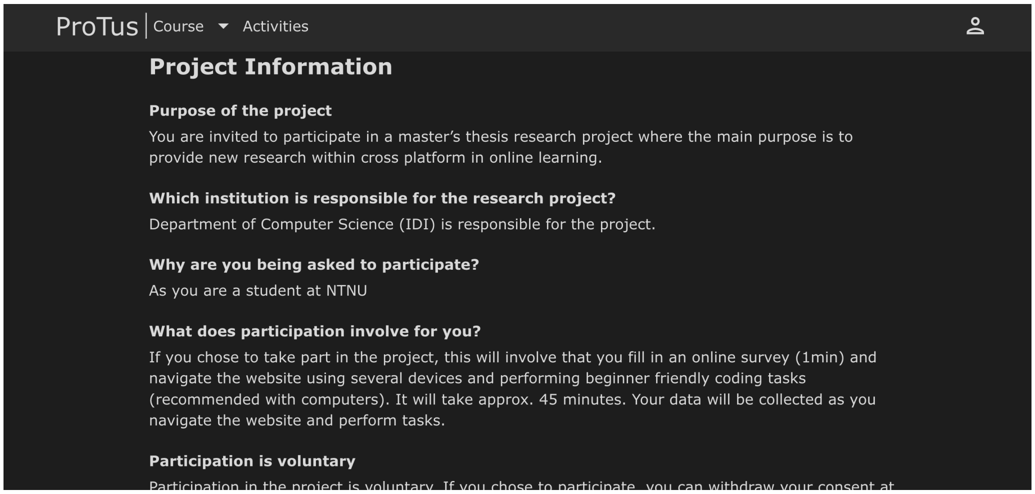

Figure 3.16: New-user page consent form

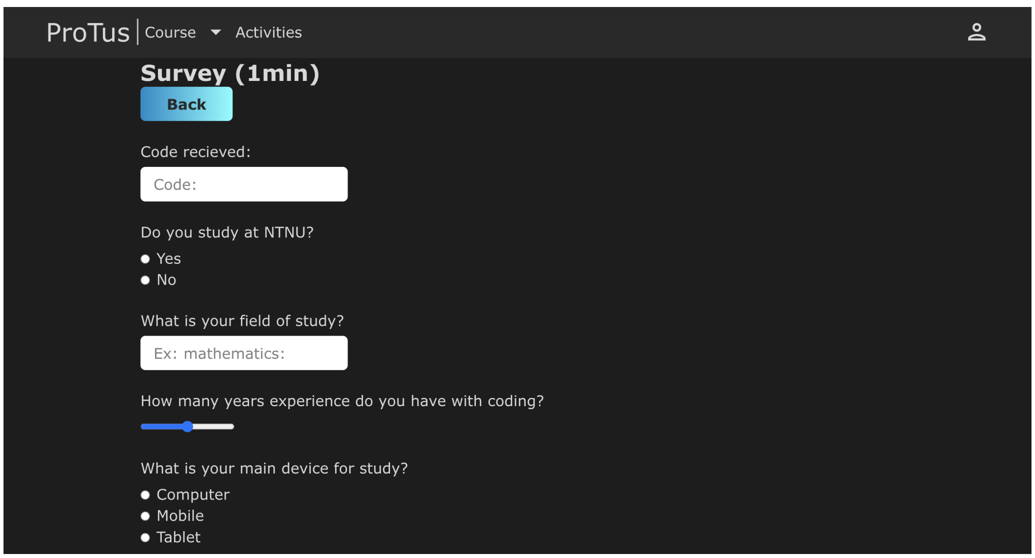

Figure 3.17: New-user page input code and demographics

that identifies what activity it is. To easily visualize the different activity types, they are assigned a different color. Users can filter activities by either using the search field or accessing the filter menu by clicking the "Filter" button. The user can filter courses, chapters, and activity types individually or combined. Clicking an activity will navigate the user to the activity page to complete the task. 3.20.

Activity Page

The page will try to render the activity from the learning API within an iframe. However, as explained in Section 3.6.4 under Challenges, activity types have different protocols. Exercises using *https* work as intended and can be viewed in Figure 3.21. Challenges using *http* is not supported by iframes and the user has to manually click the link to get redirected to the external API as seen in Figure 3.22. Nevertheless, metadata are collected for statistics that could be used for learning improvements. For more information regarding the collected data see

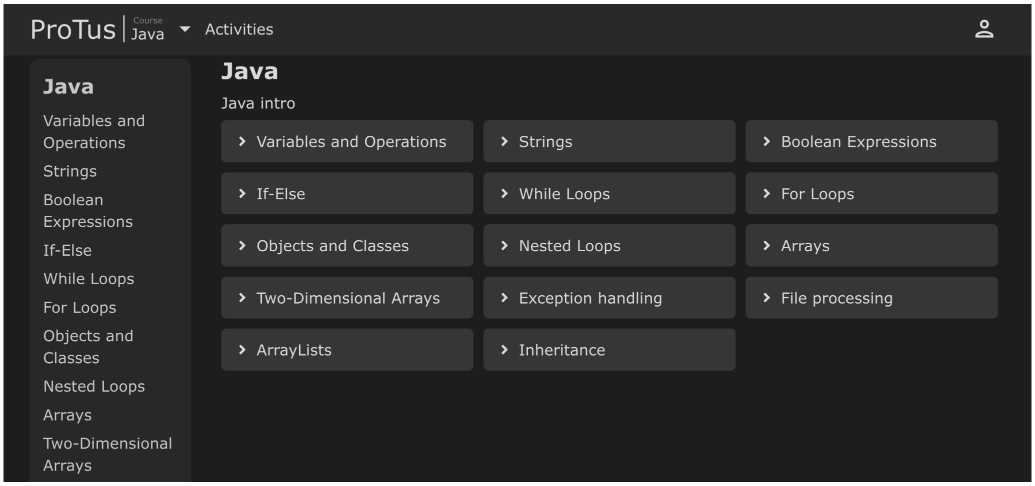

Figure 3.18: Course page

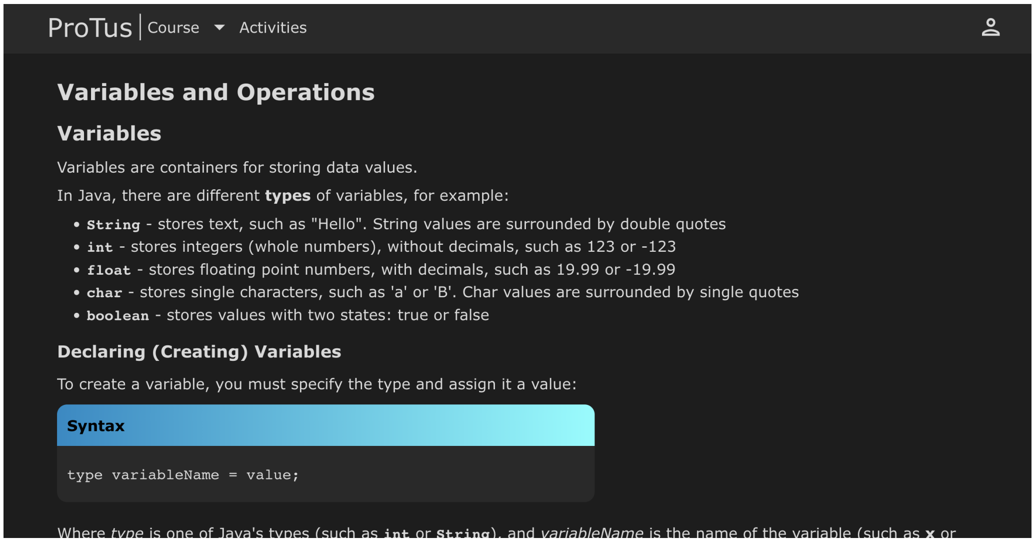

Figure 3.19: Chapter page

Section 4.2.1.

Profile Page

The user gets presented with information including their name, email, image, and, their corresponding user type. Everything can be changed except the latter, by pressing the fields and entering new information, followed by pressing save.

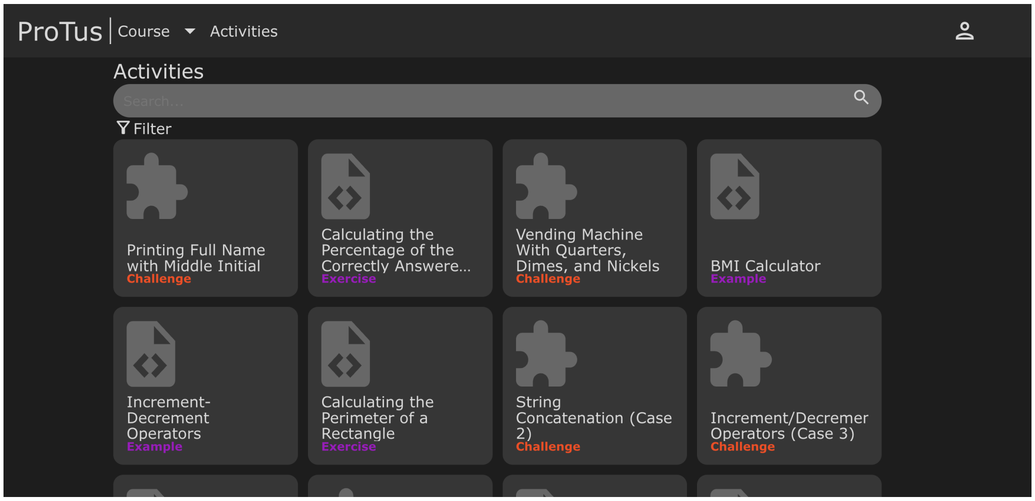

Figure 3.20: Activities page

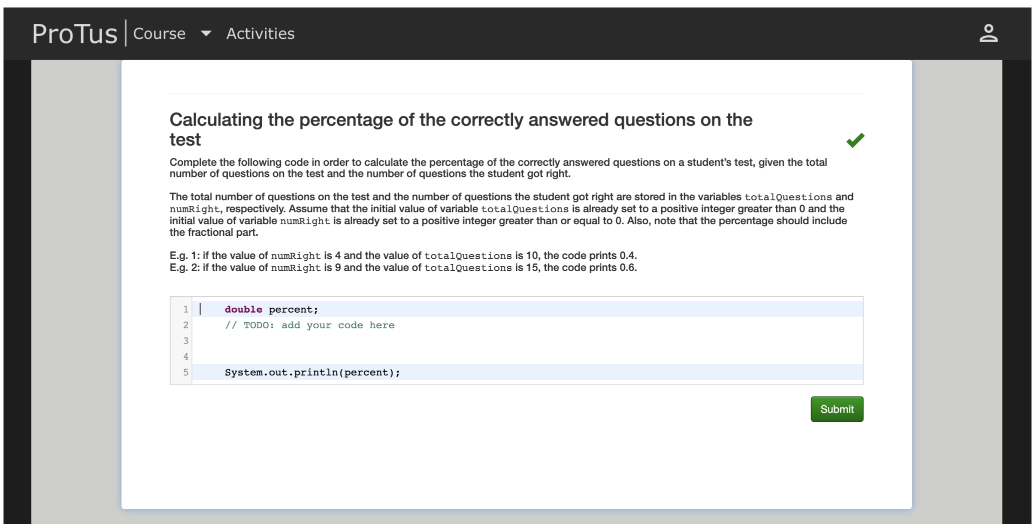

Figure 3.21: Activity page using *https* protocol

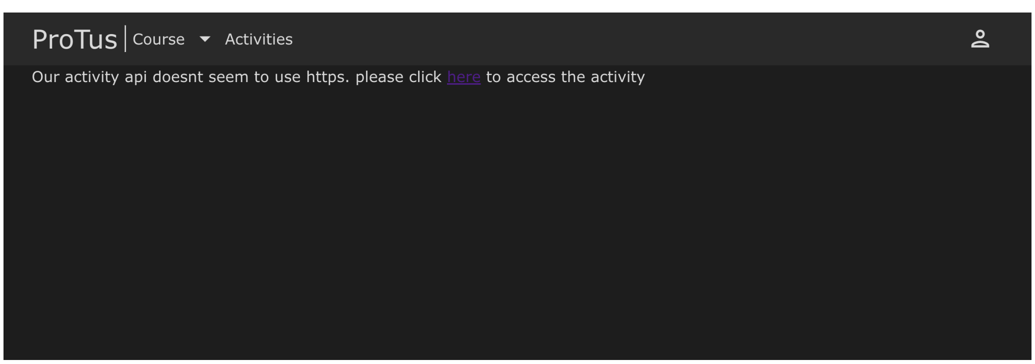

Figure 3.22: Activity page using *http* protocol

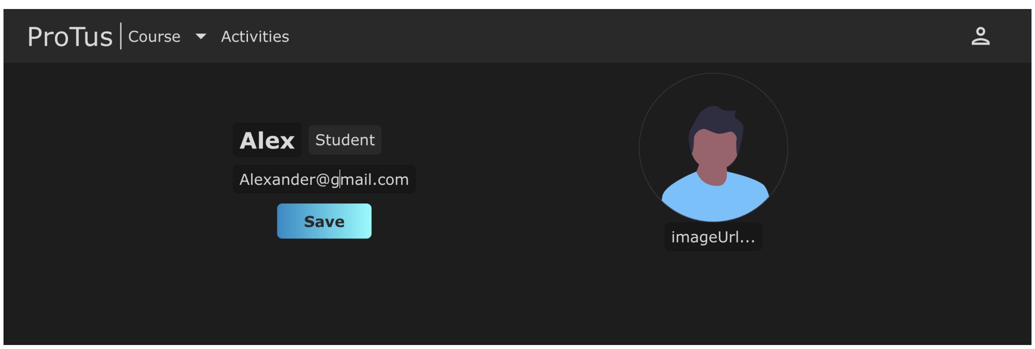

Figure 3.23: Profile page

Chapter 4

Method

This Chapter gives an overview of how the study was conducted. It was conducted in accordance with the description in Section 1.3. First, the study design is introduced in Section 4.1 containing participants partaking in a digital observation. Additionally, the observation procedure is listed. Next, the data collection methods including observation and questionnaire are discussed in Section 4.2. Last, Section 4.3 includes how quantitative analysis was used to analyze the results in this study.

4.1 Study Design

This Section will introduce the participants who took part in this study, followed by the test environment and how the observation was performed. These study design methods were used in similar studies mentioned in Chapter 2. For instance, Mesfer et al. [16] developed a software prototype used in an experiment and observed the participants. Similarly, Protus-next was created and the difference is that this study contains observation digitally. However, as the X-API contains limitations related to the assessment process in learning as described in Section 2.1.5, it was decided to include a questionnaire similar to the study by Kljunić and Vukovac [14].

4.1.1 Participants

A total of 51 participants partook in the observation. As Protus-next is designed for students without any programming knowledge, there were no requirements to participate in the study, except that the participants had to be students. Additionally, it was decided to limit the target group to students at Norwegian University of Science and Technology (NTNU), since it would arguably provide a narrower educational level as NTNU have high educational standards. Also, it would allow exploring patterns regarding the field of study at NTNU. Furthermore, this would provide the opportunity for conducting similar studies at different universities, which can be comparable.

Ultimately, the sampling technique would ideally be *probability random sampling* [6, p. 97] to capture random participants that are representable as NTNU students. The sample size would be gathered by walking on campus and asking students at random. However, as the authors of this thesis were not located in Norway while the study was conducted, a different approach was needed. A *non-probability* sampling technique were used by mixing *self-selection* and *convenience sampling* [6, p. 98]. Reaching out to student friends and posting in different student media allowed for gathering participants to conduct the study. Protus-next is an OLP for programming but does not require any previous experience. Nevertheless, some people might be skeptical to participate in the study as they do not have any previous knowledge. Thus, it was important to give a clear message stating that previous knowledge is not required as it is not included in the research field. Additionally, the observers tried to entice potential participants by informing them that a random participant will receive a gift card for participating in the study. They were all contacted using the messages found in Appendix A.2. Additional information regarding the participants is discussed in Section 5.1

4.1.2 Digital Observation

As previously mentioned, the authors of this thesis were not located in Norway while the study was conducted, therefore a physical observation with participants from NTNU was not possible. Fortunately, the observation did not require the authors to be available as the participants will only navigate the website, as explained in Section 1.3.2. Thus making the observation low maintenance as each observation session did not require additional resources [6, p. 136]. In order to obtain trustworthy results, the participants should perform the test in a controlled environment to avoid distractions. However, it was not possible to ensure this as the students were not present. To combat this, the participants were informed in their brief to perform the test alone and without distractions.

4.1.3 Procedure

Each observation session lasted between 30-45 minutes. As previously mentioned, the authors of this thesis were not present during the observation, thus a precise duration was arguably difficult to obtain as factors like reading time affected the individuals' efficiency.

The observation sessions were divided into three parts to ensure structure and similar experiences. The parts were **Briefing, User Observation** and **Debriefing**. Following this structure ensured good planning and progress for the user, rather than keeping track of user progression during the observation.

Briefing
Briefing the participants with clear instructions was essential as they would perform the observation session by themselves. Additionally, a short description introducing the system and research purpose was introduced. The participants were informed by a private message through different social communication platforms, which can be viewed in Appendix A.3

User Observation
Following the instructions from the brief, the participants were asked to create an account when navigating to Protus-next. The participants had to fill out a consent form before collecting their data. To ensure that they would give their consent before navigating the website, a form was implemented into the system and required acceptance before the participants were able to proceed. The consent form is displayed in Appendix A.4.

Furthermore, the participants were asked to enter information (referred to as account information) including demographic, study habits, and a provided code (to enable data tracking). Next, they were given access to the website and were able to explore freely. Also, they were asked to use different types of devices, like a mobile and a tablet (if available). Last, the participants were encouraged to try the activities, though not required. It was decided to create an observation with close to no guidelines when they were navigating the website, as collecting data from user interest was key for the research. Thus, a strict guideline would limit the user and not provide the data needed. The observations are elaborated in more detail in Section 4.2.1.

Debriefing
The participants were asked to proceed to a questionnaire (Appendix A.5) after navigating Protus-next for a minimum of 20 minutes, which is further elaborated upon in Section 4.2.2. As the system uses English as the standard language, settling upon the same standard within the questionnaire kept terms from changing. Also, changing language could cause confusion. It was decided to include a questionnaire for two reasons. First, it was important to get either confirmation or negation that the website was optimized for the devices used by the participants during the observation sessions. Second, uncovering the motivation for why or why not use additional devices could give more insight when creating an online learning cross-platform. Combining both an observation and a questionnaire provided different data types that combined can be used to verify user thoughts with user behavior.

4.2 Data Collection

This Section discusses the process of data collection. It elaborates on the respected methods observation, and questionnaire, along with why they were chosen in this study.

4.2.1 User Observation

This study uses observation to collect data in human-computer interaction to observe the participants during the sessions [6, p. 212-213]. As the observers were not present since it was not required based on the data generation method where user data was logged. Systematic observations are defined by Oates as ”where you decide in advance the particular type of events you want to observe, and use a predesigned schedule to note their frequency or duration.”[6, p. 214]. This type of observation was chosen as it fitted logging user data. Additionally, Oates describes systematic observation with the following advantages and disadvantages.

Advantages

- It discovers what people really do, rather than what they say they do.
- It is a means of collecting substantial amounts of quantitative data relatively quickly, and the data is pre-coded, so ready for analysis.
- It can generate data about things that most participants are normally unaware of or would regard as mundane, such as the time wasted deleting spam from email inboxes or logging onto a network.

Disadvantages

- It is restricted to studying overt behavior, and cannot explain intentions, meanings, or reasons.
- It is restricted to studying overt behavior, and cannot explain intentions, meanings, or reasons.
- It is often difficult to provide feedback to the people who have been observed, raising questions about the ethics of using people for the researcher's own ends with no benefit to the others involved.

Oates points to an interesting advantage where data generated could be perceived as mundane. It is exactly this type of data that was collected during the observation sessions. The amount of time spent on part of a website may not be compelling for participants, however combining this data with additional behavior such as the device used to navigate the website, could give answers to the research questions in this study. Furthermore, the advantages are arguably greater than the disadvantages.

4.2.2 Questionnaire

As previously mentioned, the questionnaire was included in the debrief of the procedure. The participants were asked about their experience and option in relation to navigating Protus-next. Furthermore, the questionnaire was created using Nettskjema.no [65] as it is recommended by NTNU [66] to avoid storing personal data such as IP addresses etc. The questionnaire was also submitted to Sikt [7] as described in Section 1.3.2.

Additionally, the questionnaire solved some of the disadvantages mentioned in the previous Section 4.2.1, as intentions and reasons can be explained using this method. Thus, the results are improved by combining the two chosen data generation methods.

The questionnaire was created using Openheimer's principles [67]. First, the questionnaire contained closed-ended questions to get general feedback in relation to the user experience during the observation. This approach was chosen as it is easy to process the data, combined with easy comparison, and useful for answering the research questions. Nevertheless, closed-ended questions have some drawbacks as the alternatives are biased and restrict spontaneous responses [67, p. 114-115]. Furthermore, including follow-up questions deepening on specific responses, provided the opportunity for additional feedback. Also, categorizing questions can be used to approach a topic through different angles, which might uncover otherwise lost information [67, p. 74]. The questions used in this questionnaire study are categorized in Table 4.1. Scales with multiple values were used to capture attitude scaling among participants [67, p. 187-189].

Category	Constructs
Evaluation of system optimization	Usability evaluation, emotions
Application accessibility	Motivation and usefulness
Cross Platform	Motivation, learning, and usefulness

Table 4.1: Categories within the questionnaire

The first category in Table 4.1 contained questions relating to optimization. These questions give insight relating to the participants' evaluation of the system optimization. The second category explores the system's accessibility. This category helps understand how accessibility affects motivation for using a system. The last category is comprised of questions relating to the usefulness of cross-platform applications and motivating factors supporting or rejecting such applications.

4.3 Data Analysis

This Section explains how the data were analyzed using quantitative analysis. Additionally, it describes to what extent the analysis provides insight regarding the research questions.

4.3.1 Quantitative Analysis

As both the account information (described in Section 4.1.3 under User Observation) and questionnaire contained mostly nominal and interval data, the data were nearly ready to be analyzed. However, the study program answers had to be cleaned and grouped as similar study programs were close to identical, compared to other fields. For instance, a nurse and a doctor are both working in the health sector, making it unnecessary to give them individual groups. Another reason was that the participants were typing their answer, resulting in some answers being misspelled.

Both account information and user observation data were collected through scripts connected directly to the database. The data was then analyzed and visualized using Excel. The data from the questionnaire were exported into Excel to calculate mean values and give a visual representation with graphs as mentioned in Section 1.3.3.

The quantitative result from both data generation methods provided insight into the participants' study behavior and their experiences while navigating the system. Furthermore, their attitude and motivation were uncovered by comparing and analyzing the results. These discoveries answer the research questions regarding interest for using for OLP with cross-platform functionality.

Chapter 5

Results

This Chapter provides the results of the study. Section 5.1 contains the demographics collected before the observations started, followed by the results from the user observation in Section 5.2. Last, Section 5.3 provides data from the questionnaire. Excel was used to calculate and visualize the data from the observation, demographics, and the questionnaire.

5.1 Participants and Demographics

This Section introduces the response group, and presents the demographics results.

Study field	**Number**	**Percentage**
Economy	4	7.8%
Engineering	1	2%
Health/ medicine	5	9.8%
Informatics/ computer science	25	49%
Musician	1	2%
Psychology	1	2%
Teaching	2	3.9%
Other technology	11	21.6%
Total	*51*	*100%*

Table 5.1: Demographic responses, study program grouped by field

The observation results in a total of 51 responses from participants. As mentioned in Section 4.1.1, the only requirement for participates was that they were NTNU students, resulting in all the responses being valid. Not surprisingly, the participants' coding experiences were highly related to their field of study. Table 5.1 displays an overview of all the different studies. Combining informatics and other technology fields resulted in 70.6% of the respondents having a technical background. Years of coding experience can be explained by the participant's class year, as no specific restrictions related to class year were implemented into the questionare 5.1. It should be noted that years of coding experience might have affected the observation as participants with low experience might not be as familiar with programming or learning online.

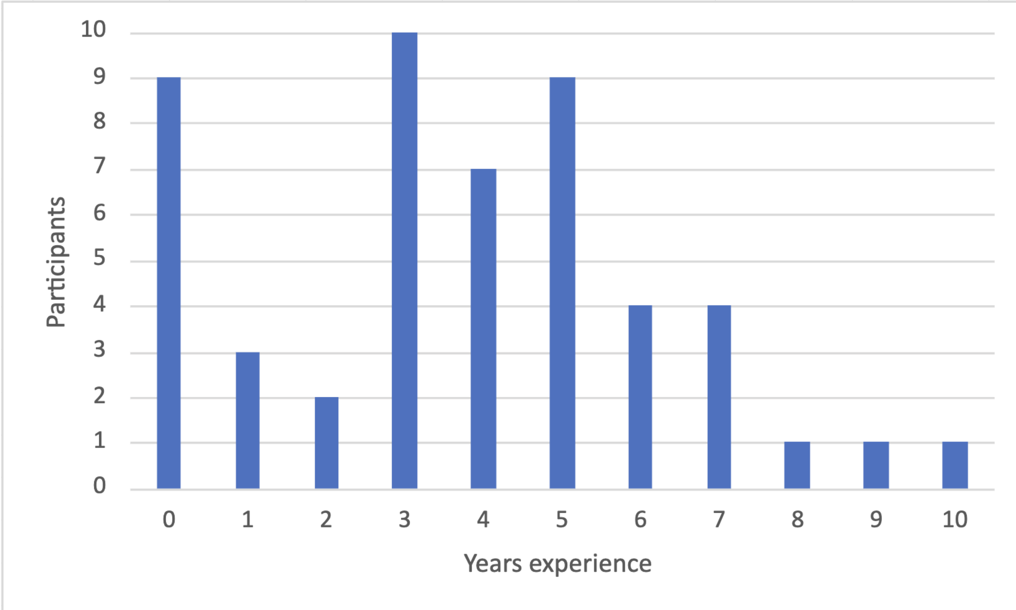

Figure 5.1: Participants' years of coding experience

Furthermore, the demographics provided insightful data where all participants use computers as their main study device. Also, 37.3% use additional study devices, where 25.5% use mobile phones and 7.8% use tablets as seen in Figure 5.2. Only 3.9% use both mobile and tablet as additional devices. Interestingly, there were mixed responses for using tablets/mobile for studying when computers are impractical. Figure 5.3 displays the results where 33.4% showed low interest (values 1 and 2) and 51% indicated high interest (values 4 and 5) in using mobile/tablet to study if the location was impractical for using computers. This resulted in an average interest of 3.2.

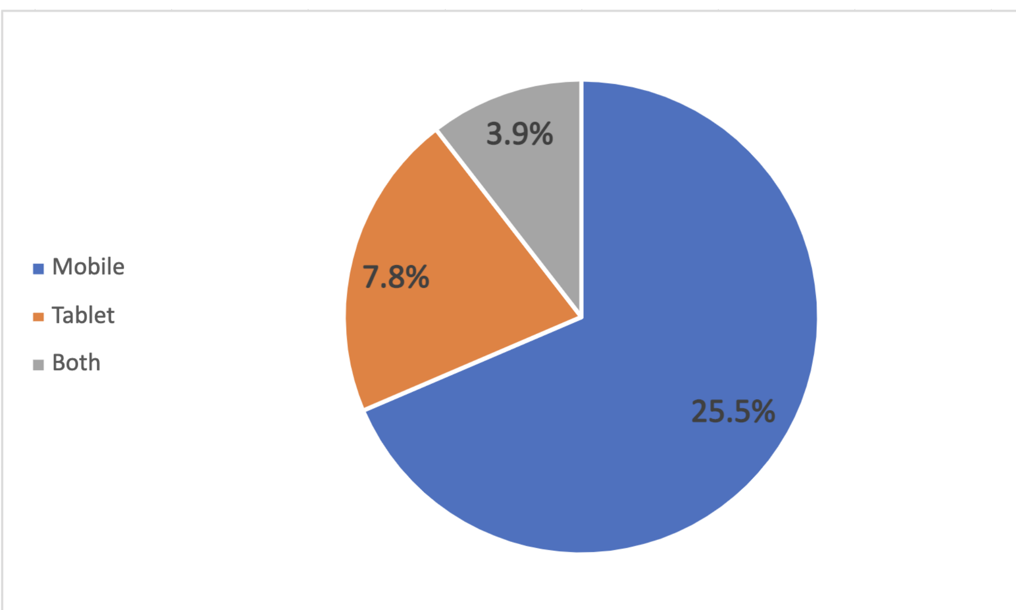

Figure 5.2: Additional study devices used participants

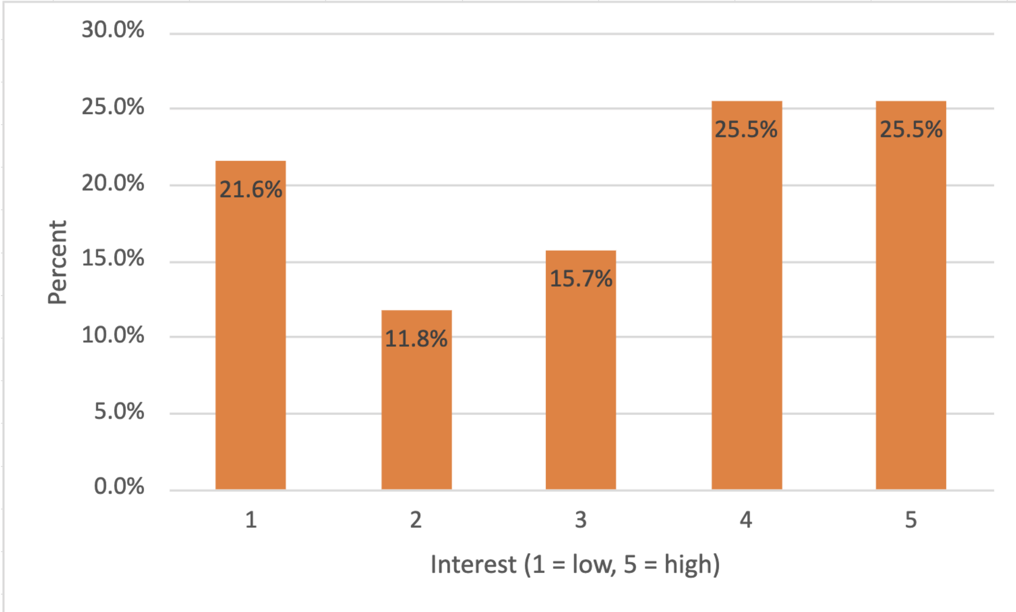

Figure 5.3: Participants interest for using mobile/tablet to study if the location is impractical for using computers.

5.2 User Observation

This Section presents the data collected from the user observation. The outcome of the observation showed that no tablets were used by the participants. Thus, the data is comprised of mobile and desktop usage.

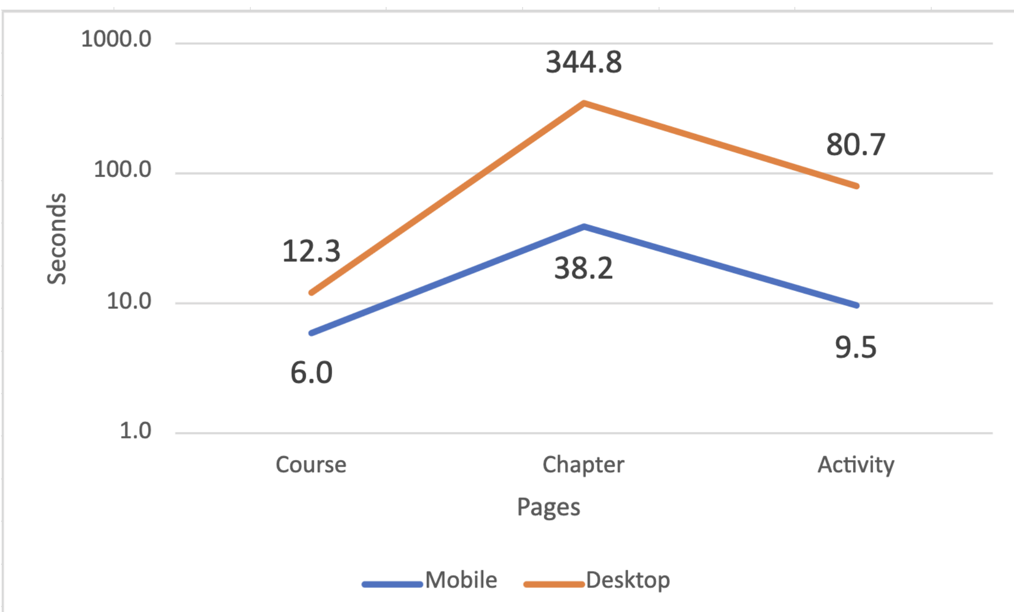

Figure 5.4: Average time spent on each page by device type.

First Figure 5.4 contains an overview of the average time the students spent on each page using different device types. On average, the desktop devices were used more than mobile on all pages. The average time spent on chapter pages was five minutes and 44 seconds using desktop devices, whereas 38 seconds was used on mobile. Looking at time spent on activities, desktop usage had an average of 80 seconds in comparison to 10 seconds on mobile. The time the students used on navigating the course pages was on average 12 seconds with desktop, and only half the time, 6 seconds on mobile. Interestingly, there is a clear pattern where participants spent most time on the chapter pages, followed by performing activities, and last on navigating the course pages. The graphs in Figure 5.5 confirms both the pattern and that desktop was most used. Further, comparing the results signifies that desktop devices had an increase of 902% time spent on chapter pages and 848% on activities compared to mobile devices.

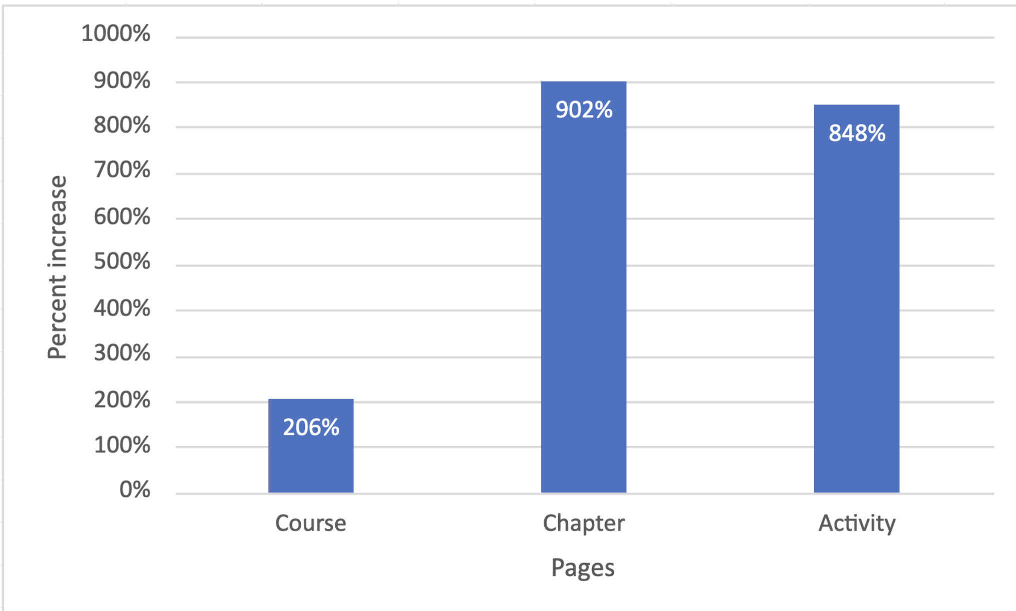

Figure 5.5: Percent increase in time spent on desktop devices vs mobile devices.

5.3 Questionnaire

This Section introduces the data collected from the questionnaire. These results were gathered after the participants' had tested Protus-next.

Questions	**Yes**	**No**
Q1: Was the system optimised for the devices you used in the application (except when preforming coding activities)?	88.2%	11.8%
Q2: Do you see yourself using phone/tablet in addition to a computer for a learning platform if it is optimised for all devices?	64.7%	35.3%

Table 5.2: The optimization related questions from the questionnaire with percentage results.

Table 5.2 contains the questions given in the questionnaire with the alternatives *yes* and *no*. 11.8% responded with the system not being optimized (excluding the coding activities), with six participants pointing to missing mobile optimization, and one pointing to missing tablet optimization. The second question resulted in

64.7% interested in using mobile/tablet in addition to desktops if the system is optimized. Those who were not interested pointed to smaller screens being the number one reason for not using additional study devices. Another highly voted issue was that the participants did not have a need for other study devices. An overview of all the results can be viewed in Figure 5.6. It should be noted that the participants could choose multiple reasons.

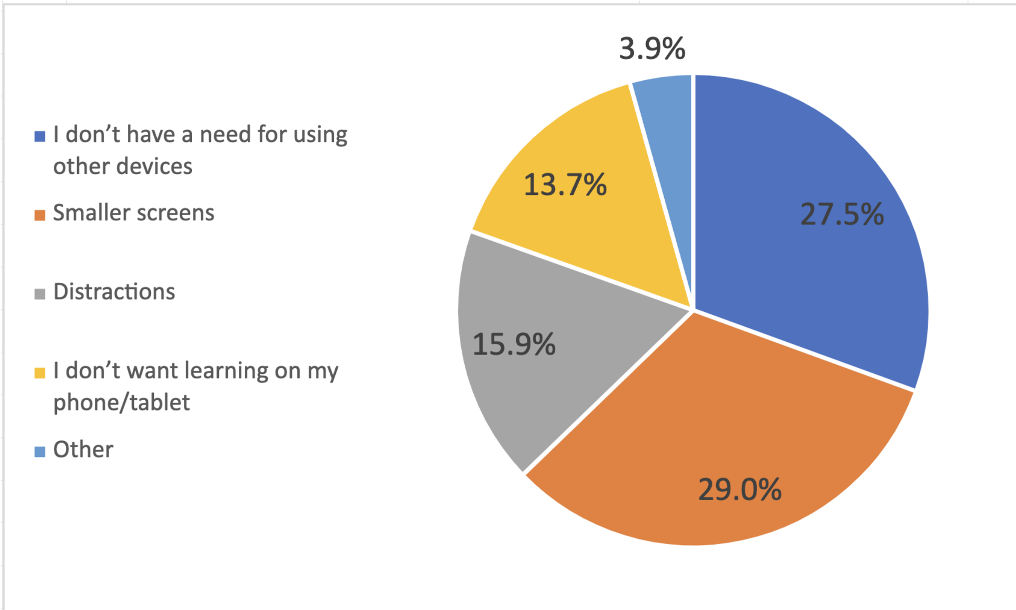

Figure 5.6: Participants' arguments for not using mobile/tablet as additional study devices.

Last, a question regarding ease of access is displayed in Figure 5.7 where the goal was to identify if accessibility is related to interest. The average interest was 3.2. captivatingly, the interest with value four spiked with 23 responses, which was more than double the amount for the next highest response.

Figure 5.8 divides participants into two groups technology and non-technology students, where the former contains students studying informatics/computer science and other technology from Table 5.1. Non-technology students had an average interest of four when asked if they would use mobile/tablet where desktops would be impractical. When asked about their interest if the website could be downloaded as an app, non-technology students had medium to high interest of 3.47. Technology students have medium average interest regarding both questions with no significant results.

Continuing the group division, Table 5.3 provides an overview of the response to the use of mobile/tablet as additional study devices if the platform is optimized for all devices. Non-technology students had a notable positive response where 80% said *yes*. Technology students had a more average result, nonetheless, 58.33% were positive to use mobile/tablet as additional devices.

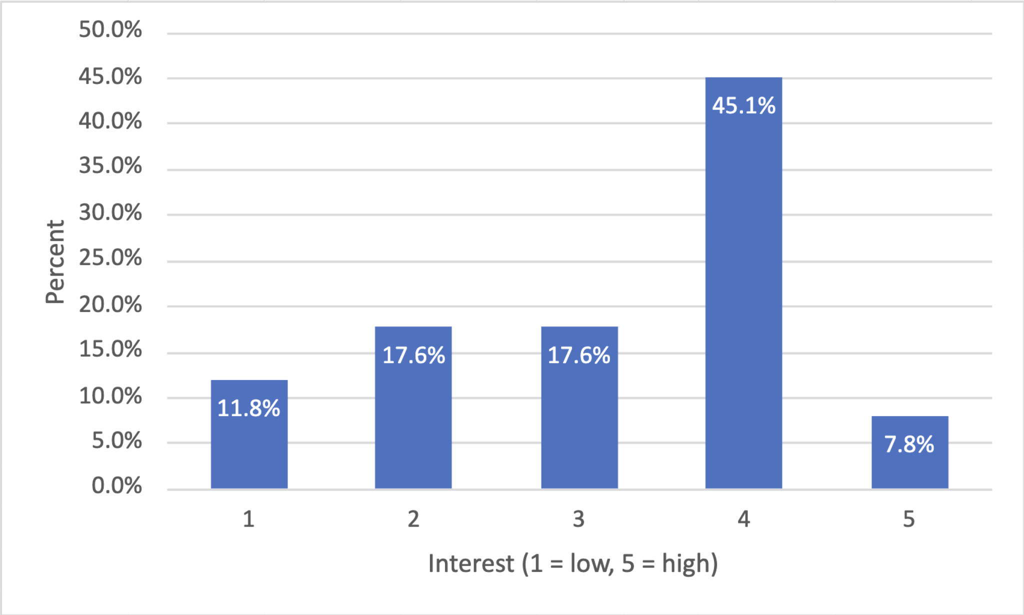

Figure 5.7: Participants' interest to use the website if it would had been a downloadable app.

.	**Tecknology students**	**Non-technology students**
Yes	58.33%	80%
No	41.67%	20%
Total	*100%*	*100%*

Table 5.3: Participants' response to using mobile/tablet in addition to a computer for learning if the platform is optimized for all devices.

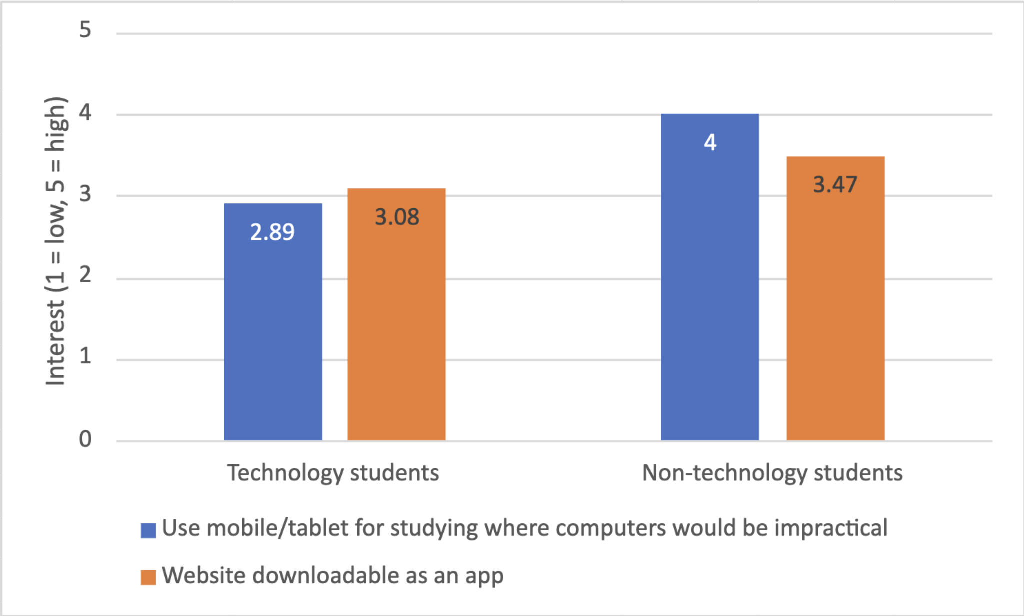

Figure 5.8: Average interest divided by study field.

Chapter 6

Discussion

6.1 Is an Online Learning Cross-Platform of any Interest to Students

6.1.1 Optimization as an Effect on Interest

As seen in Figure 5.2, 88% thought the system was optimized, in relation to user experience. However, the coding exercises were not optimized when using mobile (see Section 5.2). Hence the question explicitly excluded activities on mobile devices. Nevertheless, some of the participants might have misunderstood the question, resulting in stating that the application was not optimized. Furthermore, the questionnaire could have been improved by asking participants what was not optimized. Some participants were motivated to give feedback by private messages wanting the opportunity to state what was not optimized in the questionnaire. In retrospect, this option should have been included. Satisfaction rate of 88% is a great result and should be considered close to sufficient. Applications lacking optimization reduce the UX. These systems are also frustrating to work with and have the potential for decreasing the user motivation as discussed in Section 2.1.3. Furthermore, lack of motivation can arguably lead to procrastination which has a negative effect on academic learning as mentioned by Goda in Section 2.1.1.

Overall, 64.7% were motivated to use additional study devices if the systems are optimized. This result proved that two third of the participants were interested in using study devices like mobile or tablets. Interestingly, only 37.3% of the participants use an additional study devices. That implies a third of the participants would like to use these devices, but are currently not. There may be several reasons for this. For instance, tablets are an extra expense and are maybe not prioritized for the students. Most people have a mobile phone, however, the device might not meet their personal criteria for being used as a study device. Nevertheless, the results indicate that there is interest in using cross-platform OLP.

6.1.2 Interest Related to Location and Familiarity

The results from Section 5.1 showed that every participant's main device was a desktop, and it can therefore be assumed to be used in most locations. Though every participant used a desktop, 37% also used at least one additional study device. Among these 37%, 29.4% of the participants had a mobile, 11.7% a laptop, and 3.9% had both. There are two scenarios for using additional study devices. The additional device can either be used in combination with the main device or it can be used standalone. This present study does not give insight related to the latter scenario. However, a study by Ally [13, p. 186-193] showed students were engaged and interested in the use of mobile learning to mitigate "lost" time in scenarios like the former. The study concluded that there is an interest in devices with smaller screens (mobile and tablet) that could be used for studying. Similar results were discovered in Section 5.3 where 52.9% of the participants showed high interest in using the system as a downloadable app that could be used for smaller devices.

Mobile used as an additional device was preferred by 29.4% of the participants, whereas tablets by only 11.7%. This indicate that the interest for cross-platform is not equally spread across all platforms. As previously mentioned, technology students prefer a desktop for studying and are generally less interested in using any additional devices. This can be explained by the sunk cost fallacy when deciding to adopt a new device. The sunk cost fallacy takes place when deciding whether or not to adopt something new, where the perceived value invested in the current choice would be lost [68]. This can explain every student's preference for a computer, as a computer continues to be introduced as a learning tool in the Norwegian school system at an increasingly earlier stage [69]. An average Norwegian student will therefore be substantially more familiar with using a computer when studing, compared to a tablet or mobile. Hence the interest in tablets and mobile would be affected by this. Today, almost everyone owns a mobile, making it more attractive as they have already spent time getting familiar with the device. This eliminates some of the sunk cost effect, but mobile learning also requires some learning time, and therefore poses an additional sunk cost effect. Conversely, tablets is far less common compared to mobile, and thus has a bigger sunk cost effect when adopting this device over the preferred device.

6.1.3 Techincal Background

The systems may lack accessibility, in relation to device type and how fast the user can access the application. Figure 5.7 shows that the participants had middle to high interest to use the website if it would have been a downloadable app. Looking more in detail, the students with technical background showed neither high nor low interest in using mobile/tablet for studying where computers would be impractical, and if the website could be downloaded as an app. Technology students will generally use a desktop more than an average student, and their time spent learning to study with a desktop creates a big sunk cost effect explained in Section 6.1.2. However, if platforms are optimized, they are more willing to use them on additional devices. This might also be a cause of the sunk cost fallacy, as better UX can reduce time spent learning the new system.

As discussed by Wan in Section 2.1.1, students with ICT capabilities have a higher virtual competence and are more independent. Therefore, the accessibility where the website can be downloaded as an app arguably does not matter to technical students as they are capable of accessing the information regardlessly. As expected, this is more important to non-technical students which is supported by the results in Figure 5.8. The students with no technical background showed a high interest in using additional devices where computers would be impractical. This is discussed further in Section 6.1.2. There were also participants not interested in using additional study devices. Though only 35.3% of the students showed low interest, reasonable arguments were given in Figure 5.6. The results might be explained by the participants being satisfied with their current study habits. For instance, distractions (from mobile devices) leads to procrastination which results in negative learning outcome as discussed by Goda in 2.1.1.

To summarize, there is some interest in using mobile/tablets as additional study devices, especially for students without a technical background. Optimized systems increasingly motivate and interest the students regardless of technical background.

6.1.4 Summary

After discussing the analyzed findings in combination with related studies, the first research question can be concluded. The findings proved that students' had an interest in online learning cross-platform. However, the interest in using an OLP was affected by the optimization standard. The present study found that students with and without technical background had different opinions regarding device type, UX, and optimization. Overall, the latter group showed higher interest in all categories. More specifically, among the non-technology students there was only an interest in using mobile or tablets for studying when computers would be impractical. These results can be affected by several factors, such as the sunk cost fallacy.

6.2 How do students manage their time on an online learning platform for improved learning

It is in the students own interest to be as effective as possible when learning, to either learn more or to increase their spare time. Being effective when learning means increasing learning outcomes, thus student interest lies in increasing learning outcomes. How this is achieved is dependent on the student, and this Section will discuss how students used an Online Learning Platform for an increased learning outcome.

6.2.1 Lack of Tablet

The demographics showed that 7.8% of the participants used tablets in addition to desktops as study devices and 3.9% uses both mobile and tablet. However, none of the participants used tablets during the observation. They were asked to use several devices and to use the one they liked more (Appendix A.3). It is a possibility that some participants did not have their tablets available when conducting the observation. As most young adults have their mobile device nearby, it would not be surprising if they thought of using mobile before tablet. After testing Protus-next with desktop and mobile, maybe the participants with a tablet available thought testing with two devices was sufficient. In retrospect, the participants should have been asked to use the same devices as they previously stated they use for studying, in order to retrieve tablet data. On the other hand, standard tablet sizes are not very different from desktop screens, implying that the lack of data is not critical, which would also make it more interesting to test on desktops and mobile compared to desktops and tablets.

6.2.2 Time Management Related to Tasks

As discussed in Section 6.1.3, there is a possibility that certain study tasks are better suited for smaller screen sizes. This is supported by Loewhen et al. (Section 2.1.2), who concluded that one of the reasons Duolingos success on mobile devices was the task simplicity. Gamification in combination with task simplicity improved users' motivation which improved their learning behavior. Furthermore, Loewhen et al. expressed their concern that implementing increasingly complicated tasks on smaller devices would affect the degree of success. Mesfer et al. [16] and Lin [15] both conducted studies of students performing different tasks, and parts of the students' learning outcomes were measured. Mesfer et al. discovered an increased learning outcome on a desktop compared to mobile when gathering information through reading. Lin, however, discovered an increased learning outcome on mobile compared to desktop when performing learning activities. All activities performed in the mentioned studies are activities that can be found within a OLP. The tasks in Protus-next are gathering information from the chapter pages and performing activities fetched from the learning API. Figure 5.4 indicates that the participants on average spent more time on the chapter pages than the activity pages. This implies that the main task was reading and thus gathering information. However, it can also be affected by other factors like the website not being fully correct when tracking activities as the participants were redirected to a third-party service. Other factors affecting time management can be that the participants already have tested the page on another device, which will be discussed in the upcoming Section.

6.2.3 Time Management Related to Device Type

When looking at the average time spent on the website, the participants used more desktops than mobile on all pages, as shown on Figure 5.4. There is a significant difference in the time used by participants on the various devices, as presented in Figure 5.5. All participants used desktops as their main study device, which may indicate that most participants would complete the test on their desktop first. When testing a system for the first time, everything is new, and emotions like curiosity, excitement, and interest affect the user. Also, the user does not know the existing navigation, which may correlate to the time spent on the system. However, when testing the same system for a second time, these emotions may not be as strong or not present at all. Furthermore, Section 2.1.3 explains how the lack of UX affects users' motivation and willingness to use the system. Thus, one always uses more time when testing a system with no prior knowledge, which might be one of the reasons the time spent on desktop devices exceeded mobile devices. To avoid that all participants would have seen the website first on desktop and then mobile, they could have been divided into two groups. The first half could have tested with their mobile first followed by their desktop, while the remaining half could have conducted the observation in the opposite order.

The results from time spent on the different devices, is supported by the study of Mesfar et al. [16], who found that mobile devices were used less than desktops. In addition, using mobile devices could be more efficient as the participants spend less time on the tasks. However, this is not parallel with the findings in the study by Mesfar et al. Nevertheless, the tasks performed in the study of Mesfar et al. differed a little from the tasks in the present study. The tasks Protus provides are as previously mentioned: gathering information from the chapter pages and performing activities fetched from the learning API. The increased time spent on desktop devices can imply that students prefer reading for learning. This also aligns with the results in Table 5.2 indicating that 35% of the participants were not motivated to use additional study devices (tablet or mobile), as students prefer using an OLP for gathering information through reading.

6.2.4 Summary

Previous research indicates that students show increased learning outcomes when using mobile in combination with simple interactive tasks, and desktop in combination with reading. This states the importance of pairing device types and tasks resulting in the optimal learning outcome. In the presented study, desktops proved to be significantly more used than mobile devices on all pages during the observation. These results were affected by different factors, such as mobile being tested after desktop. However, the differences were too significant to be explained by any of the factors. Considering that students are interested in being as effective as possible when learning, and desktops being the most effective for reading information, strongly indicates that students mostly use an OLP for reading.

6.3 Limitation

The results seem promising, but it does not come without limitations. This Section discusses issues and limitations in relation to the conducted study.

6.3.1 Sample Selection

As discussed in Chapter 4.1.1, the participants were found by using self-selection and convince sampling. This may have affected the participants' answers when performing the questionnaire as they know the authors conducting this study.

6.3.2 Learning API

The learning API had a bad performance with an average response time of 12 seconds. This resulted in the necessity to copy the data into a separate database. This used unnecessary resources, especially since the API did not return a response recommended for website frameworks. Furthermore, the learning API did not have any documentation, which resulted in extra time used to understand the API. The learning API had only one endpoint, where best the practice is to have multiple based on research collections [70].

6.3.3 X-API Integration

When performing an activity, the integration with X-API could have been improved to log more data. At the current integration, the X-API has limited logging functionality of the learning API.

6.3.4 Mapping Study Behaviour by Using X-API

Logging scenarios in the form of activities does not cover the student behavior with the current integration. Section 2.1.1, presented several aspects that affect student behavior and in Section 2.1.5, Nouira [30] points out that the X-API does not cover the assessment process which is a major part of study behavior and learning outcome. The study in this thesis has investigated cross-platform functionality within an Online Learning Platform (OLP) to improve learning outcomes. To fully cover different parts within the chosen topic, two research questions were created for evaluating student interest and track the students' time management on an OLP. In order to answer these questions, an OLP Protus-next was designed, developed, and tested on 51 NTNU students.

Chapter 6 discussed the result of this study, which showed the participants' interest in using additional study devices either in combination with a desktop or standalone. The results were affected by the systems UX and optimization. Dividing the participants into groups with technical and non-technical background, showed a divided interest in cross-platform OLP. Non-technical students tended to appreciate the accessibility of a system and had a higher interest in using cross-platform OLPs. Students with technical background were less interested in using additional study devices, due to reasons like the sunk cost fallacy. Nevertheless, it was discovered that a third of all participants showed interest in using additional study devices, but are currently not using them. Further results showed that the time used with desktops exceeded mobile devices while navigating the pages within Protus-next. And for the different pages, the students preferred the reading material over interactive tasks for learning. Previous studies have shown that using the correct device type for a task improves learning outcome, including both mobile and desktop devices. This may indicate that the students mostly use Online Learning Platforms for reading.

Future Work

This study has discovered new insights related to online learning and cross-platform. However, as discussed in Section 6.3, there are limitations related to the study method and system design. The following list contains directional opportunities for future research and development.

- Conducting a similar study with a larger and more diverse target group. This would strengthen the data and ensure that aspects like technical background do not affect the results.
- This study was conducted on university students. Similar studies performed on another target group, or over longer periods could give different results. For instance, conducting a study over a longer period, would allow for a more accurate testing of learning outcome, such as the study of students in a classroom by Lan [15]. This could then verify the learning outcome from Lan and Mesfar et al. and give a better comparison of the two.
- A few participants stated that Protus-next was not optimized for mobile devices, which could have affected time spent on different pages. Correcting the lack of optimization can give better UX and has the potential for different results if the same study was conducted again.
- For further research, it would be interesting to add technical background as metadata to X-API statements and investigate how it affects the time spent on different pages using different devices. Combing this implementation with finishing tests could reveal the learning outcome and any relation to their lower average interest for cross-platform OLP.
- X-API has a lot of potentials that is not utilized in Protus-next. As discussed in Section 6.3.3, the learning API sets limitations for the X-API, which could provide new useful insight.
- The learning API has a bad performance of an average 12 seconds response time. Improving the current API with new standards would minimize latency and remove the need for a database to store data from the API. This would ultimately improve the optimization of the website and increase the UX, especially on mobile.

Acronyms

ICT Information Communications Technology. 7, 10, 52

LRS Learning Record Store. 9, 15

NTNU Norwegian University of Science and Technology. 38, 39, 41, 44, 55

OLP Online Learning Platform. 1, 2, 4, 6, 10–12, 39, 43, 50, 52–56

ORM Object Relational Mapping. 17

OS operative system. 12, 13

TDD test-driven development. 31

UML Unified Modeling Language. 19

UX User Experience. 8–10, 50–52, 54–56

Bibliography

[1] Vesin Boban et al. 'Mile - Multifunctional Integrated Learning Environment.' In: (Jan. 2008), pp. 104–108.

[2] Aleksandra Klašnja-Milićević et al. 'E-Learning personalization based on hybrid recommendation strategy and learning style identification'. In: *Computers & Education* 56.3 (2011), pp. 885–899. ISSN: 0360-1315.

[3] Boban Vesin. *ProTuS*. http://www.sudskavestacenja.com/protus/. (Accessed on 04/07/2023). 2023.

[4] Aleksandra Klašnja-Milićević, Boban Vesin and Mirjana Ivanović. 'Social tagging strategy for enhancing e-learning experience'. In: *Computers & Education* 118 (2018), pp. 166–181. ISSN: 0360-1315.

[5] Boban Vesin, Aleksandra Klašnja-Milićević and Mirjana Ivanović. 'Protus 2.1: Applying Collaborative Tagging for Providing Recommendation in Programming Tutoring System'. In: (2016), pp. 236–245.

[6] Briony June Oates. *Researching Information Systems and Computing*. Sage, 2005.

[7] Sikt. *Sikt - Norwegian Agency for Shared Services in Education and Research*. https://sikt.no/en/home. (Accessed on 04/17/2023).

[8] LOVDATA. *Lov om behandling av personopplysninger (personopplysningsloven)*. https://lovdata.no/dokument/NL/lov/2018-06-15-38/KAPITTEL_1#KAPITTEL_1. (Accessed on 04/04/2023).

[9] Oxford Reference. *E-learning*. https://www.oxfordreference.com/view/10.1093/oi/authority.20110803095745661. (Accessed on 13/03/2023).

[10] Zeying Wan, Yinglei Wang and Nicole Haggerty. 'Why people benefit from e-learning differently: The effects of psychological processes on e-learning outcomes'. In: *Information & Management* 45.8 (2008), pp. 513–521. ISSN: 0378-7206.

[11] Irshad Hussain and Sarwat Sultan. 'Analysis of procrastination among university students'. In: *Procedia - Social and Behavioral Sciences* 5 (2010). WCPCG 2010, pp. 1897–1904. ISSN: 1877-0428.

[12] Yoshiko Goda et al. 'Procrastination and other learning behavioral types in e-learning and their relationship with learning outcomes'. In: *Learning and Individual Differences* 37 (2015), pp. 72–80. ISSN: 1041-6080.

[13] M. Ally. *Mobile Learning: Transforming the Delivery of Education and Training*. Canadian electronic library: Books collection. Athabasca University Press, 2009, pp. 183–393.

[14] Jasmina Kljunić and Dijana Plantak Vukovac. *A Survey on Usage of Mobile Devices for Learning among Tertiary Students in Croatia*. https://www.researchgate.net/profile/Dijana-Plantak-Vukovac/publication/282857297_A_Survey_on_Usage_of_Mobile_Devices_for_Learning_among_Tertiary_Students_in_Croatia/links/561f7df308ae93a5c923fefe/A-Survey-on-Usage-of-Mobile-Devices-for-Learning-among-Tertiary-Students-in-Croatia.pdf. (Accessed on 14/03/2023). 2015.

[15] Chih C. Lin. 'Learning English reading in a mobile assisted extensive reading program'. In: *Computers & Education* 78 (2014), pp. 48–59. ISSN: 0360-1315.

[16] Alrizq Mesfer et al. 'Analysis of Skim Reading on Desktop versus Mobile Screen'. In: *Applied Sciences* 11.16 (2021). ISSN: 2076-3417. URL: https://www.mdpi.com/2076-3417/11/16/7398.

[17] Sahar Abbasi et al. *Perceptions of students regarding E-learning during Covid-19 at a private medical college - PMC*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7306963/. (Accessed on 14/03/2023). 2020.

[18] Duolingo. *Duolingo*. https://www.duolingo.com/. (Accessed on 05/13/2023).

[19] Shawn Loewen et al. 'Mobile-assisted language learning: A Duolingo case study'. In: *ReCALL* 31.3 (2019), pp. 293–311.

[20] Barrero Jose Maria, Bloom Nicholas and Davis Steven J. *Why Working from Home Will Stick*. Working Paper 28731. National Bureau of Economic Research, Oct. 2021. DOI: 10.3386/w28731.

[21] Hartson Rex and Pyla S. Pardha. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. 1st. Morgan Kaufmann Publishers Inc., 2012, pp. 3–7. ISBN: 0123852412.

[22] Gardner S. Brett. 'Responsive web design: Enriching the user experience'. In: *Sigma Journal: Inside the Digital Ecosystem* 11.1 (2011), pp. 13–19.

[23] Cambridge Dictionary. *CROSS-PLATFORM*. https://dictionary.cambridge.org/dictionary/english/cross-platform. (Accessed on 11/26/2022).

[24] Suyesh Amatya and Arianit Kurti. *Cross-Platform Mobile Development: Challenges and Opportunities*. https://link.springer.com/chapter/10.1007/978-3-319-01466-1_21. (Accessed on 11/26/2022). 2013.

[25] Google. *Google Chrome: Fast & Secure*. https://play.google.com/store/apps/details?id=com.android.chrome&hl=en_AU&gl=US. (Accessed on 11/28/2022). 2022.

[26] Paulo R. M. de Andrade and Adriano B. Albuquerque. *Cross platform app: a comparative study*. https://arxiv.org/abs/1503.03511. (Accessed on 11/28/2022). 2015.

[27] adlnet. *GitHub - adlnet/xAPI-Spec*. https://github.com/adlnet/xAPI-Spec. (Accessed on 05/15/2023).

[28] Rustici Software. *xAPI*. https://xapi.com/overview/. (Accessed on 04/05/2023).

[29] Bakhouyi Abdellah et al. 'Evolution of standardization and interoperability on E-learning systems: An overview'. In: (2017), pp. 1–8.

[30] Azer Nouira, Lilia Cheniti-Belcadhi and Rafik Braham. 'An Enhanced xAPI Data Model Supporting Assessment Analytics'. In: *Procedia Computer Science* 126 (2018), pp. 566–575. ISSN: 1877-0509.

[31] Cambridge International. *Getting started with Assessment for Learning*. https://cambridge-community.org.uk/professional-development/gswafl/index.html. (Accessed on 04/05/2023).

[32] Ahmad Uzir Nora'ayu et al. 'Analytics of time management strategies in a flipped classroom'. In: *Journal of Computer Assisted Learning* 36.1 (2020), pp. 70–88.

[33] Françeska Xhakaj, Vincent Aleven and Bruce M. McLaren. 'Effects of a Dashboard for an Intelligent Tutoring System on Teacher Knowledge, Lesson Plans and Class Sessions'. In: (2017), pp. 582–585.

[34] Capacitor. *Using Capacitor with React - Capacitor*. https://capacitorjs.com/solution/react. (Accessed on 05/22/2023).

[35] Google. *Add & open Chrome apps - Chrome Web Store Help*. https://support.google.com/chrome_webstore/answer/3060053?hl=en. (Accessed on 05/22/2023).

[36] Yaqub M. (CBC). *Responsive Web Design Stats: State of Responsive Design 2022*. https://www.businessdit.com/responsive-web-design-statistics/. (Accessed on 05/22/2023). Apr. 2023.

[37] Ruth Malan and Dana Bredemeyer. *Functional Requirements and Use Cases*. https://www.bredemeyer.com/pdf_files/functreq.pdf. (Accessed on 07/03/2023). 2001.

[38] Siva Balan and M Punithavalli. 'Software Architecture, Scenario and Patterns'. In: *International Journal of Computer Science Issues* 7 (Sept. 2010). (Accessed on 07/03/2023).

[39] Figma. *Figma: the collaborative interface design tool*. https://www.figma.com/. (Accessed on 23/02/2023).

[40] Notion. *Notion – Your wiki, docs & projects. Together*. https://www.notion.so/. (Accessed on 23/02/2023).

[41] GitHub. *GitHub: Let's build from here*. https://github.com/. (Accessed on 23/02/2023).

[42] Next.js. *Next.js by Vercel - The React Framework*. https://nextjs.org/. (Accessed on 23/02/2023).

[43] TypeScript. *TypeScript: JavaScript With Syntax For Types*. https://www.typescriptlang.org/. (Accessed on 23/02/2023).

[44] Vercel. *Vercel: Develop. Preview. Ship. For the best frontend teams*. https://vercel.com/. (Accessed on 04/06/2023).

[45] NextAuth. *NextAuth.js*. https://next-auth.js.org/. (Accessed on 04/06/2023).

[46] Okta. *What's the Difference Between OAuth OpenID Connect and SAML*. https://www.okta.com/au/identity-101/whats-the-difference-between-oauth-openid-connect-and-saml/#:~:text=The%20main%20differentiator%20between%20these,industry%20standards%20for%20federated%20authentication.. (Accessed on 05/15/2023).

[47] MongoDB. *MongoDB*. https://www.mongodb.com/. (Accessed on 04/05/2023).

[48] Zod. *Zod — Documentation*. https://zod.dev/. (Accessed on 04/06/2023).

[49] tRPC. *tRPC*. https://trpc.io/. (Accessed on 05/15/2023).

[50] IBM. *Remote Procedure Call - IBM Documentation*. https://www.ibm.com/docs/en/aix/7.1?topic=concepts-remote-procedure-call. (Accessed on 05/15/2023).

[51] Prisma. *Prisma*. https://www.prisma.io/. (Accessed on 04/05/2023).

[52] Sanity. *The Composable Content Cloud - Sanity.io*. https://www.sanity.io/. (Accessed on 04/05/2023).

[53] Microsoft. *Microsoft Excel Spreadsheet Software*. https://www.microsoft.com/en-us/microsoft-365/excel. (Accessed on 04/21/2023).

[54] Storybook. *Storybook: Frontend workshop for UI development*. https://storybook.js.org/. (Accessed on 23/02/2023).

[55] Sass. *Sass: Syntactically Awesome Style Sheets*. https://sass-lang.com/. (Accessed on 23/02/2023).

[56] P.B. Kruchten. 'The 4+1 View Model of architecture'. In: *IEEE Software* 12.6 (1995). (Accessed on 06/03/2023), pp. 42–50.

[57] Kimbell L. *Rethinking Design Thinking: Part I*. https://www.tandfonline.com/doi/pdf/10.2752/175470811X13071166525216?needAccess=true. (Accessed on 11/01/2022). 2011.

[58] Jonathan Arnowitz and Nevin Berger Michael Arent. *Effective Prototyping for Software Maker*. https://vdoc.pub/documents/effective-prototyping-for-software-makers-7t9mqqgbuf70. (Accessed on 11/02/2022). 2007.

[59] Jim Rudd, Ken Stern and Scott Isensee. 'Low vs. High-Fidelity Prototyping Debate'. In: *Interactions* 3.1 (Jan. 1996), pp. 76–85. ISSN: 1072-5520.

[60] Color Hunt. *Color Palette*. https://colorhunt.co/palette/1b262c0f4c753282b8bbe1fa. (Accessed on 11/30/2022).

[61] Linda Holtzschue. *Understanding color: an introduction for designers*. John Wiley & Sons, 2012.

[62] Brett S Gardner. 'Responsive web design: Enriching the user experience'. In: *Sigma Journal: Inside the Digital Ecosystem* 11.1 (2011), pp. 13–19.

[63] W3Schools. *W3Schools Online Web Tutorials*. https://www.w3schools.com/. (Accessed on 12/01/2022).

[64] Carol M. Barnum. 'Usability Testing Essentials: Ready, Set ...Test!' In: Second Edition. Morgan Kaufmann, 2021, pp. 9–33.

[65] Nettskjema. *Nettskjema*. https://nettskjema.no/user. (Accessed on 04/17/2023).

[66] NTNU. *Nettskjema - Kunnskapsbasen - NTNU*. https://i.ntnu.no/wiki/-/wiki/English/Nettskjema. (Accessed on 04/17/2023).

[67] A.N. Oppenheim. *Questionnaire Design, Interviewing and Attitude Measurement*. Pinter Publishers, 1992.

[68] Hal R Arkes and Catherine Blumer. 'The psychology of sunk cost'. In: *Organizational Behavior and Human Decision Processes* 35.1 (1985), pp. 124–140. ISSN: 0749-5978.

[69] udir. *Den digitale tilstanden i Skole-Norge*. https://www.udir.no/tall-og-forskning/publikasjoner/utdanningsspeilet/utdanningsspeilet-2021/digital-tilstand/. (Accessed on 05/18/2023).

[70] Martin Ekuan et al. *Web API design best practices - Azure Architecture Center*. https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design. (Accessed on 05/21/2023). 2023.

Appendix

Appendix A

Study

A.1 Code repositYory

Github code repository: https://github.com/JohanAOstbye/protus

A.2 Contacting Participants

Enkeltpersoner

Hei [navn], håper alt står bra til
Jeg skriver for tiden master med en kompis og lurte på om du kunne teste systemet vi har laget. Jeg kan friste med at en av deltagerne vinner 500kr og det kan bli deg! Masteren går ut på å finne ut hvordan en læreplatform på nett fungerer på ulike enheter. Eksperimentet går ut på å navigere rundt på en nettside med bruk av ulike enheter som mobil og pc. Etterpå må du svare på noen få spørsmål i et fler-alternativ skjema. Så hva sier du?

Grupper og Andre Platformer

Hei,
Vi er to masterstudenter som trenger deltagere til å teste et system vi har laget. Vi kan friste med at en av deltagerne vinner 500kr og det kan bli deg! Masteren går ut på å finne ut hvordan en læreplatform på nett fungerer på ulike enheter. Eksperimentet går ut på å navigere rundt på en nettside med bruk av ulike enheter som mobil og pc. Etterpå må du svare på noen få spørsmål i et fler-alternativ skjema. Gi bedskjed om du ønsker å delta

A.3 Brief

Takk for at du kan delta. Hele eksperimentet kommer til å ta mellom 30-45 minutter. Systemet er en platform der man kan lære å programmere uten noen forkunnskaper. Det kreves IKKE at du kan programmere! Når du navigerer på nettsiden legger du igjen informasjon som (enhet-type, hvor lang tid brukt på ulike deler av siden osv.). Den informasjonen kommer vi til å sammenligne med andre deltagere. Det betyr også at du kan gjennomføre testen uten meg tilstedet. Gjennomfør testen uten forstyrrelser eller andre tilstedet. Slik fungerer det:

1. Bruk Chrome, Safari eller Firefox nettleser.
2. Gå inn på https://protus.vercel.app/ og trykk "login" knappen.
3. Trykk på "Register" og fyll inn email og passord.
4. Når du logger inn får du generel informasjon om formålet med eksperimentet, info om hva du skal gjøre om du vil trekke deg osv.
5. Trykk "Accept" og fyll ut litt informasjon om deg selv (1 min). Under CODE skal du fylle inn dette: CODE: [...]
6. Nå skal du navigere rundt på nettsiden (den kan være litt treig når den skal laste inn data), gjerne prøv på noen "Activitites" men ikke noe krav å få det til, vi tester som sagt ikke kunnskap. Test nettsiden på flere enheter og gjerne bruk den du liker best. Vi har ingen satt tid, men helst bruk minst 20 minutter for å legge igjen gjennomsnittlig data.
7. Gå til https://forms.gle/HHHDRjpxycSekWxy9 og svar på et par spørsmål (1 min).
8. Send en melding til meg å si at du er ferdig.

A.4 Consent Form

Project Information

Purpose of the project

You are invited to participate in a master's thesis research project where the main purpose is to provide new research within cross platform in online learning.

Which institution is responsible for the research project?

You are invited to participate in a master's thesis research project where the main purpose is to provide new research within cross platform in online learning.

Which institution is responsible for the research project?

Department of Computer Science (IDI) is responsible for the project.

Why are you being asked to participate?

As you are a student at NTNU

What does participation involve for you?

If you chose to take part in the project, this will involve that you fill in an online survey (1min) and navigate the website using several devices and performing beginner friendly coding tasks (recommended with computers). It will take approx. 45 minutes. Your data will be collected as you navigate the website and perform tasks.

Participation is voluntary

Participation in the project is voluntary. If you chose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. There will be no negative consequences for you if you chose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data

We will only use your personal data for the purpose(s) specified here and we will process your personal data in accordance with data protection legislation (the GDPR). We will only use your personal data for the purpose(s) specified here and we will process your personal data in accordance with data protection legislation (the GDPR). The data collected are stored in a secure database that no unauthorized persons are able to access the personal data.

What will happen to your personal data at the end of the research project?

The planned end date of the project is in July and all personal data will be deleted.

Your rights

So long as you can be identified in the collected data, you have the right to:

- access the personal data that is being processed about you.
- request that your personal data is deleted.
- request that incorrect personal data about you is corrected/rectified.
- receive a copy of your personal data (data portability).
- send a complaint to the Norwegian Data Protection Authority regarding the processing of your personal data.

What gives us the right to process your personal data?

We will process your personal data based on your consent. Based on an agreement with Department of Computer Science, The Data Protection Services of Sikt – Norwegian Agency for Shared Services in Education and Research has assessed that the processing of personal data in this project meets requirements in data protection legislation.

Where can I find out more?

If you have questions about the project, or want to exercise your rights, contact:

- Department of Computer Science via Boban Vesin
  boba.vesin@ntnu.no 48217455
- Student Vemund Eggemoen
  vemundeg@stud.ntnu.no 46808784
- Student Johan August Østbye.
  johanaos@stud.ntnu.no 91301594

If you have questions about how data protection has been assessed in this project by Sikt, contact:

- email: (personverntjenester@sikt.no) or by telephone: +47 73 98 40 40.

By clicking the following button, I have received and understood information about the project Online Learning Platform - Cross Platform and have been given the opportunity to ask questions. I give consent for my personal data to be processed until the end of the project.

Accept

A.5 Nettskjema Questionnaire

Protus Anonymous Survey

Mandatory fields are marked with a star *

Email address *

Was the system optimised for the devices you used in the application (except when performing the coding activities)? *

- ◯ Yes
- ◯ No

If no, please elaborate

This element is only shown when the option "No" is selected in the question "Was the system optimised for the devices you used in the application (except when performing the coding activities)?"

- ☐ Computer
- ☐ Mobile
- ☐ Tablet

If the website could be downloaded as an app, would it be more interesting to use? *

Lowest — Highest

1 2 3 4 5

Verdi

Do you see yourself using phone/tablet in addition to a computer for a learning platform if it is optimised for all devices? *

- ◯ Yes
- ◯ No

If no, please elaborate

- ☐ I don’t have a need for using other devices
- ☐ Smaller screens
- ☐ Distractions
- ☐ I don’t want learning on my phone/tablet
- ☐ Other

Appendix B

Final User Interface - Protus-next

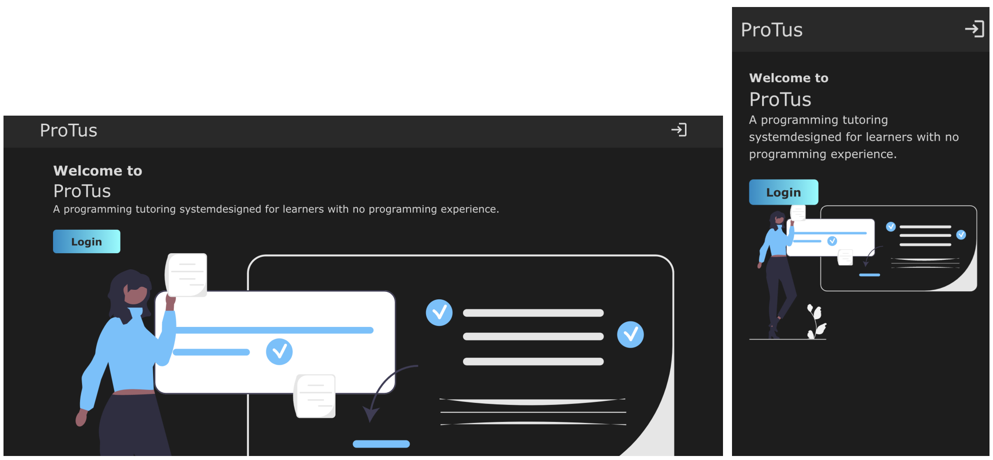

Figure B.1: Frontpage not authenticated

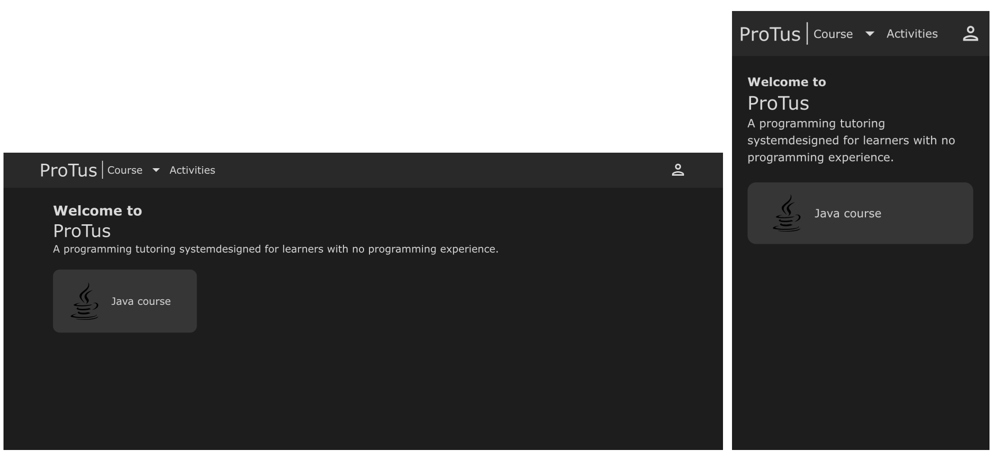

Figure B.2: Frontpage while authenticated

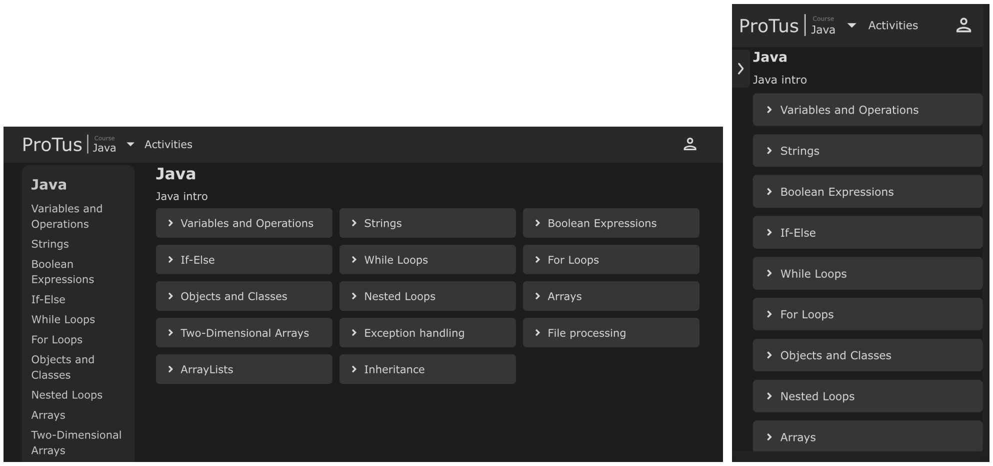

Figure B.3: Course page

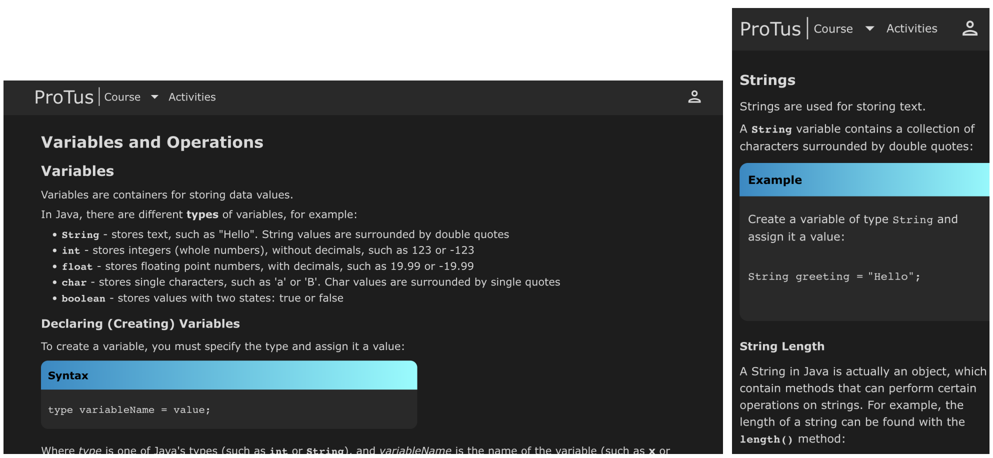

Figure B.4: chatper page

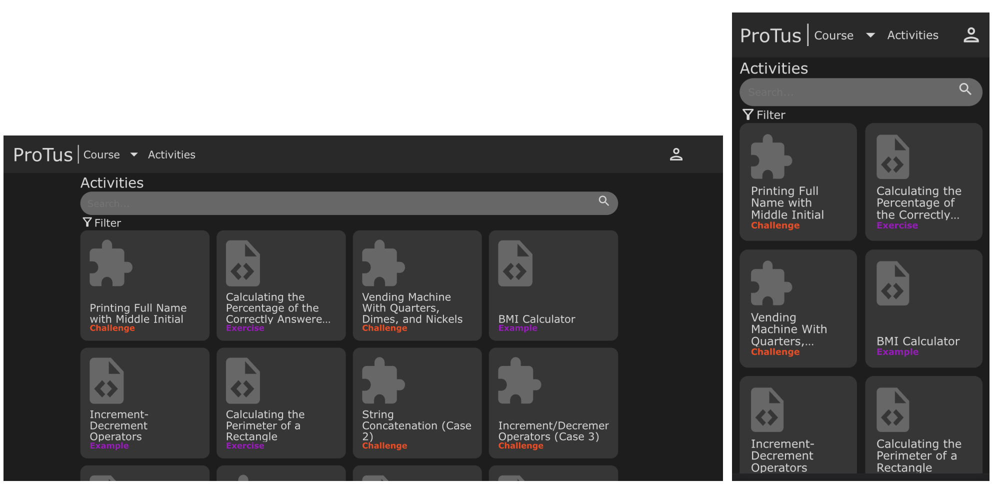

Figure B.5: activitites page

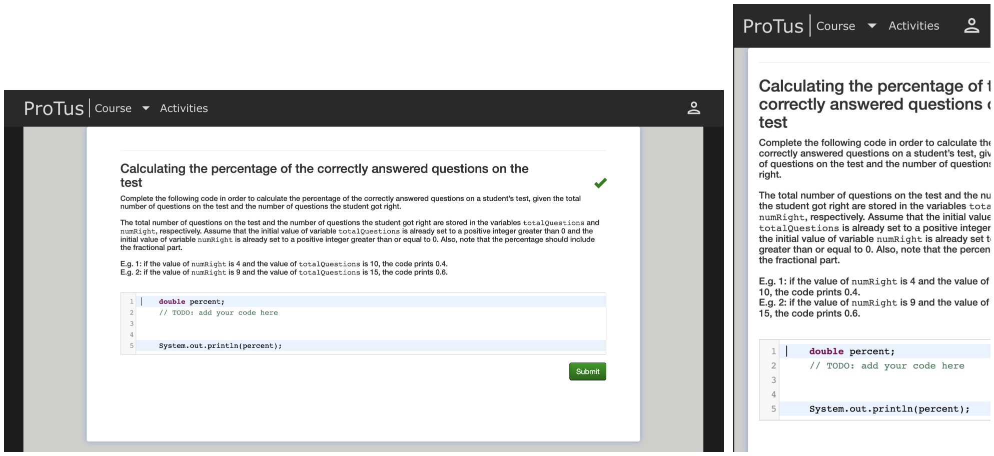

Figure B.6: Activity page using https

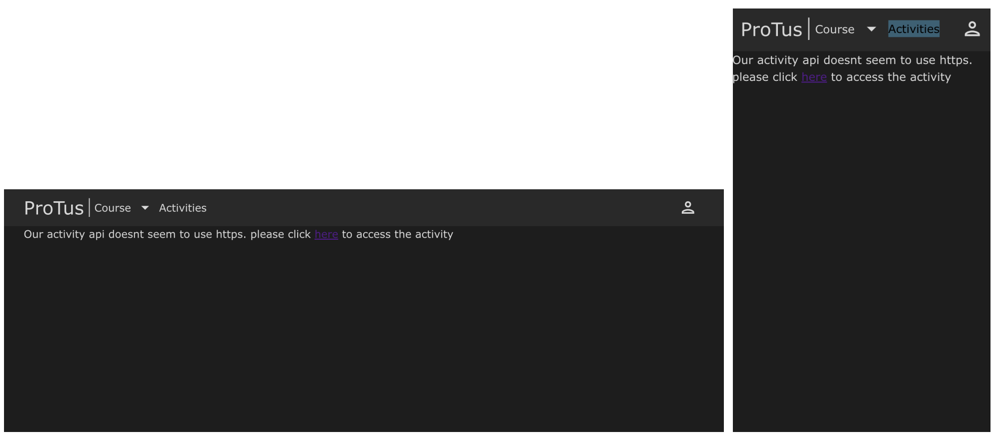

Figure B.7: Activity page using http

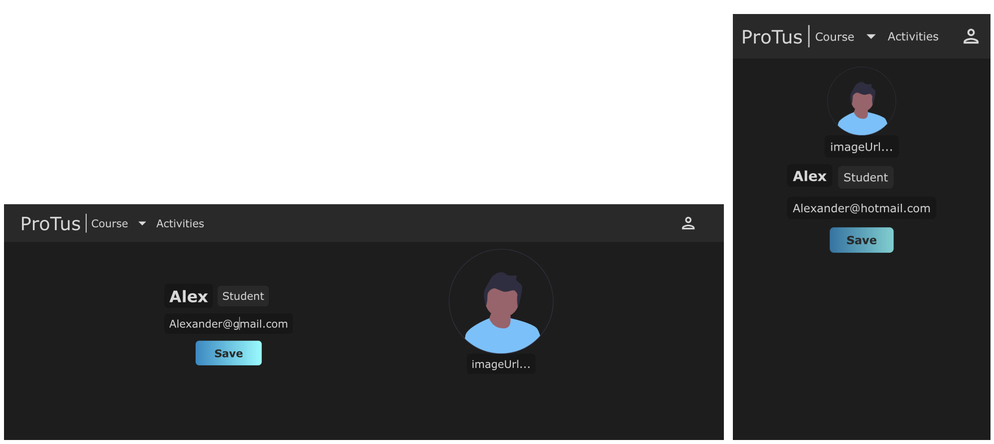

Figure B.8: Profile page

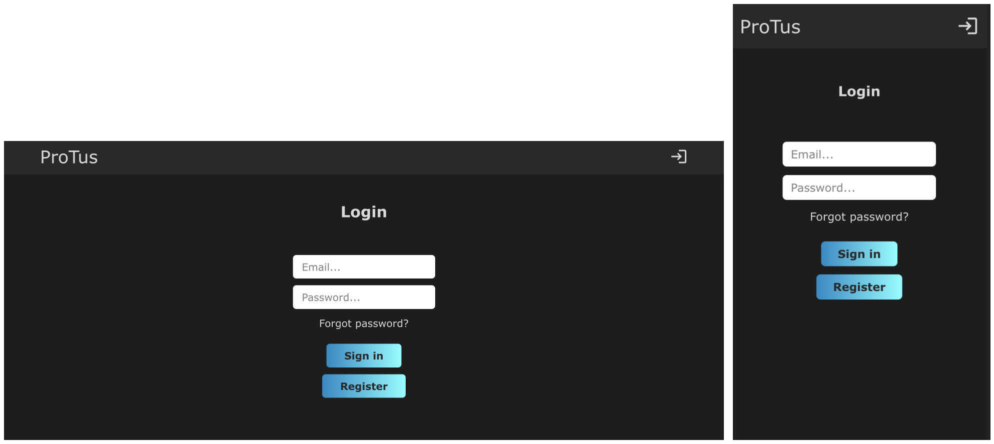

Figure B.9: Sign in page

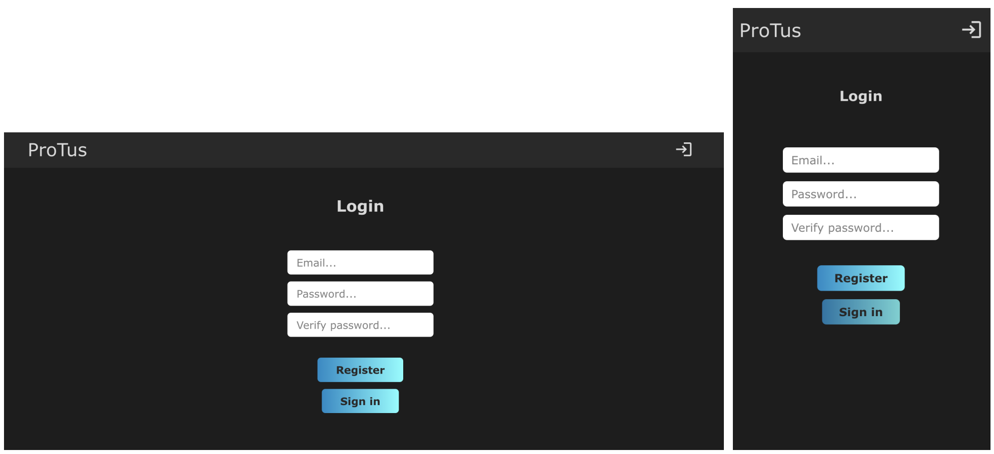

Figure B.10: Register page

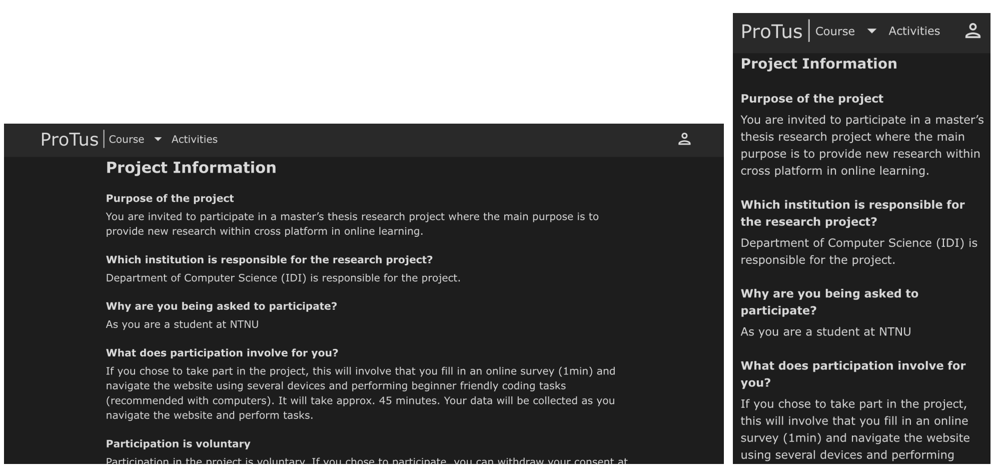

Figure B.11: New-user page terms

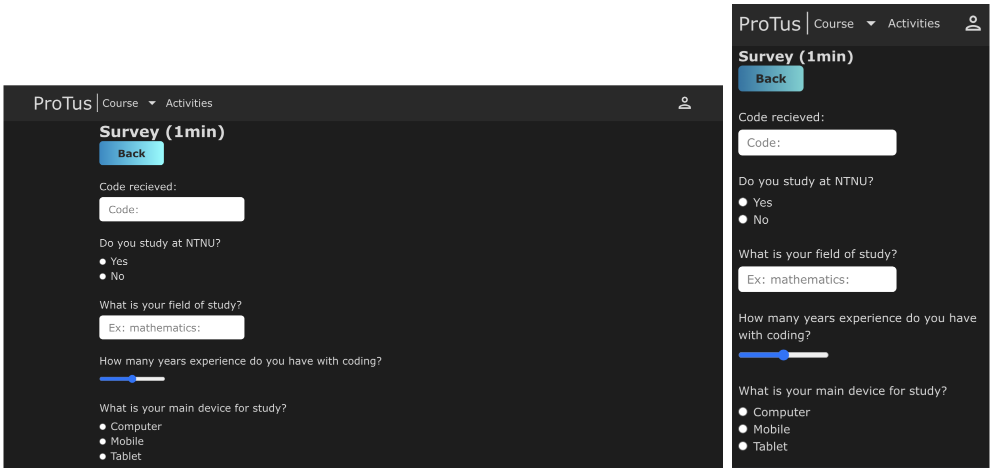

Figure B.12: New-user page input survey

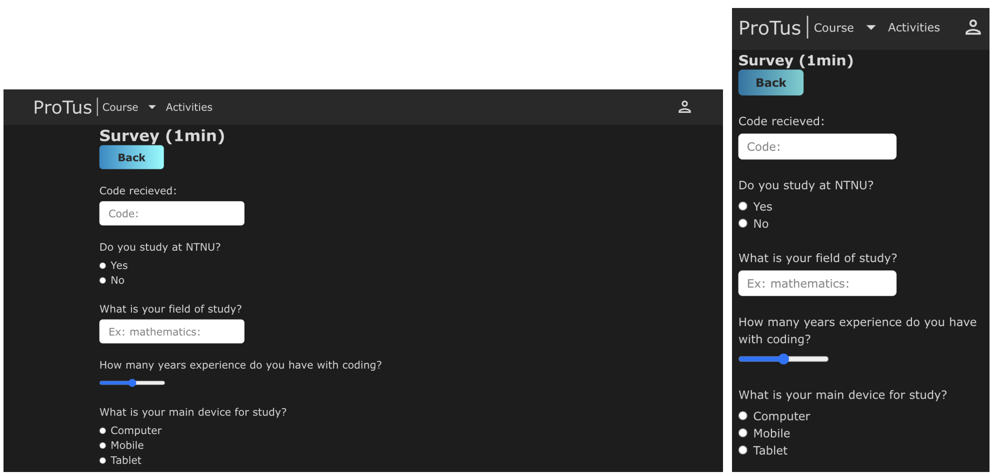

Figure B.13: New-user page input survey

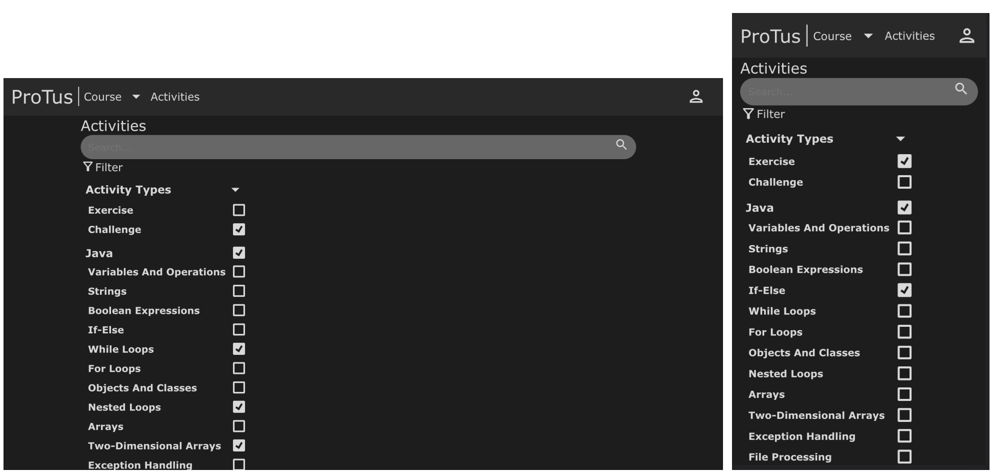

Figure B.14: Activities page using filter

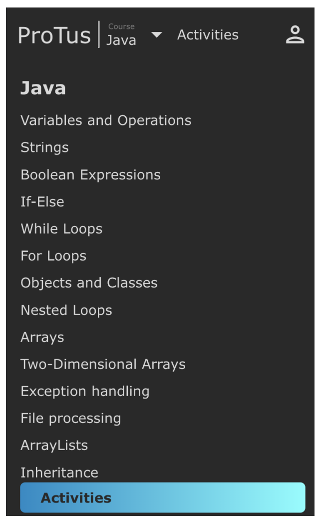

Figure B.15: Sidebar on mobile

NTNU

Norwegian University of
Science and Technology