Mathias Fossum & Johanne Tronstad

Predicting learning outcomes and motivating student achievement in interactive learning environments: guidelines for learning design

Master's thesis in Informatics: Interaction Design, Game and Learning Technology
Supervisor: Boban Vesin
Co-supervisor: Michail Giannakos
June 2023

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Mathias Fossum & Johanne Tronstad

Predicting learning outcomes and motivating student achievement in interactive learning environments: guidelines for learning design

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

The field of e-learning has received significant attention over the last few years. The effectiveness of e-learning platforms lies in whether they can give the students appropriate guidance through their learning journey while keeping them engaged and motivated. One of the key challenges involves determining the optimal combination of learning content, engagement tactics, and feedback mechanisms that maximize the student's learning outcomes and foster engagement. Traditional e-learning platforms typically offer the students a one-size-fits-all approach, lacking definite solutions for designing learner analytics dashboards. The thesis contributes with a new LA dashboard that offers customization options and facilitates solving object-oriented programming tasks with corresponding progress tracking. The application was tested during a one-month experiment on students (n=73) from the University of South-Eastern Norway. They generated 4630 exercise data entries. The following data analysis discovered which low-level on-task predictors influence the learning outcome. While strong correlations were found, the explanatory power (R^2) contributed to a low explanation of variance in learning outcomes. Moreover, the study found that those who chose to participate in a competitive environment tended to show higher learning outcomes. Finally, the study provides guidelines for designing effective learner analytics dashboards. By providing insights into these two aspects, the thesis aims to contribute to the e-learning field and proposes new directions for future research.

SAMMENDRAG

De siste årene har e-læring fått betydelig oppmerksomhet. Gode e-læringsplattformer kjennetegnes ved å kunne gi studentene tilstrekkelig veiledning for å kunne fasilitere aktiv læring. En av de viktigste utfordringene involverer å identifisere den optimale kombinasjonen av læringsinnhold og tilbakemeldingsmekanismer som både maksimerer studentenes læringsresultater og fremmer engasjement. Tradisjonelle e-læringsplattformer tilbyr som regel upersonaliserte løsninger, uten klare retningslinjer for å designe gode brukerstyrte dashboards. Oppgaven bidrar med et nytt e-læring dashboard som tillater tilpasningsalternativer til studenten, og oppgaver i objektorientert programmering med progresjonssporing. Programmet ble testet under et eksperiment på en måned på studenter (n=73) fra Universitetet i Sørøst-Norge. Gjennom testperioden løste de 4630 oppgaver til sammen. Under dataanalysen ble det oppdaget hvilke lavnivå predikatorer som påvirker læringsutbytte. Selv om det ble funnet sterke korrelasjoner, bidro determinasjonskoeffisienten (R^2) til en minimal variasjon i læringsresultatene. Videre fant studien at de som valgte å delta i et konkurransefremmende miljø, viste tendenser til å oppnå høyere nivå av læringsresultat. Til slutt gir studien retningslinjer for å designe effektive dashboards for læringsanalyse. Ved å gi innsikt i disse to aspektene, har oppgaven som mål å bidra til feltet e-læring og foreslå nye retninger for fremtidig forskning.

ACRONYMS

AUC Area under the Curve. 49, 52, 71

BKT Bayesian Knowledge Tracing. vii, 1, 2, 9, 10, 71, 81, 82, 84

BKT+F BKT+Forget. 9, 84

FSM Finite-State Machine. 9, 49

HMM Hidden Markov Model. 9

KT-IDEM Knowledge Tracing - Item Difficulty Effect Model. iv, 10, 49, 71

KT-PPS Knowledge Tracing - Prior Per student. 10

LMS Learning Management System. 7

MAPE Mean Absolute Percentage Error. v, 11, 48

MVP Minimum Viable Product. 16, 29

NaN Not a Number. 71

ProTuS Programming Tutoring System. 12

RMSE Root Mean Square Error. 49, 52

SUS System Usability Scale. 13, 47, 61, 78, 80

ACKNOWLEDGEMENTS

The team would sincerely like to thank our supervisor Boban Vesin for his tremendous effort in providing us with test candidates, knowledge, and direction during the master period. Boban always answered in detail and on time. We strongly recommend Boban to any master's students who consider one of his master's theses in the future.

We would like to express our gratitude to Katerina Mangaroska for providing us with critical insights during the statistical analysis and overall structure of the thesis.

We would also like to thank Anirudhan Badrinath at the University of California, Berkley for providing guidance and explanation on the KT-IDEM and Item Learning Effect models of pyBKT.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER

ONE

INTRODUCTION

This thesis articulates the development, testing, and data analysis of the learner analytic dashboard *Progresso*. During a one-month period, *Progresso* was used and tested by two object-oriented programming classes (n=73), first-year and second-year students, of The University of South-Eastern Norway. An analysis of parametric statistical tests and regression was conducted by identifying predictors of learning outcome estimated by Bayesian Knowledge Tracing (BKT). Lastly, it was identified if and which elements of a customizable learning environment contribute to increased engagement in students.

1.1 Problem description

The domain of this master's thesis is the field of e-learning. In e-learning, tutors and students effectively engage in learning despite being physically separated or lacking direct one-on-one interaction, using digital learning platforms [1]. E-learning offers unprecedented accessibility by enabling students to access learning content at their convenience [2]. Data collection capabilities that track progress and engagement in real-time may contribute to additional insight uncaught by the tutor [3][4]. Lastly, flexibility allows students to self-assess, work at their own pace, and tailor the learning environment to their needs [5][6].

The flexibility of e-learning extends into the research field of personalization. Personalization aims to create learning environments that are tailored to the student's preferences or dynamically adapt based on the student's interaction [7]. These adaptions could for instance be flexible customization options that convey the progress, performance, or learning outcome metrics of the student. Such customization options have in a previous study entailed what data-visualization components students want in a dashboard [5].

In summary, e-learning platforms should equip students with the appropriate guidance to successfully master a set of skills [8]. Estimating the learning outcome is hard because knowledge is an unobservable construct [9]. One can assess the skill mastery, by inferring the probability of a student knowing a skill based on observable data [10], such as a student's correct and wrong exercise answers. Bayesian Knowledge Tracing (BKT) was introduced in 1995 [10], but this thesis uses a new implementation of BKT that incorporates several newly developed model adoptions of traditional BKT [11].

1.2 Motivation

One of the major challenges in e-learning involves determining which combination of learning content, engagement tactics, and feedback mechanisms will not only maximize learning outcomes but also maintain satisfactory levels of student engagement with the platform [2]. Current e-learning platforms often adopt a uniform approach, focusing on boosting engagement, motivation, or skill mastery, without adequately exploring the relationship between these elements [12]. Students have many preferred learning environments and appreciate performance metrics differently [7]. As such, there exists no definite solution for designing a learner analytics dashboard.

High-level dimensions of learning outcomes such as course grade, institutional strategy, and teaching strategy, as well as the student's own learning strategy, have been used to determine what maximizes learning outcomes [13]. However, there is still a limited exploration of low-level on-task metrics, such as exercise attempt count and the duration taken by students to complete an exercise, in terms of their potential in explaining learning outcomes [14].

Gamification elements, such as leaderboards, have been employed to enhance motivation and engagement in e-learning [15]. While gamification elements have been found to be predictors of motivation and engagement, it is unclear if a relationship exists between a competitive learning environment and learning outcome [16].

Much research has been devoted to formulating and recommending customized learning paths [17][18][19]. Learning paths refer to a sequence of educational steps designed to maximize the student's potential learning outcome [18]. Such paths are typically recommended on an individual basis [19]. However, this master thesis shifts focus, exploring whether the initial choice of learning content ultimately predicts a higher learning outcome. This may reveal the initial attitude the student has towards learning by assessing the student's angle of attack.

1.3 Research Questions

In light of the problem description and motivation, two research questions were made that combine the topics of skill mastery, gamification in learner analytics dashboards, and student engagement. The thesis aims to answer the following research questions;

- **Research Question 1:** What predictors influence the learning outcome in an e-learning environment?
  - **RQ 1.1:** How does the number of attempts influence skill mastery?
  - **RQ 1.2:** What is the impact of on-task duration on skill mastery?
  - **RQ 1.3:** What difference in skill mastery is seen between first-year and second-year students?
  - **RQ 1.4:** Does the first step in the learning path influence skill mastery?
  - **RQ 1.5:** How does leaderboard participation contribute to skill mastery?
- **Research Question 2:** How can we design learner analytics dashboards in order to boost student engagement?

1.4 Research Methods

The research process adopted for the study is highlighted in Figure 1.4.1 and follows Oates' research framework [20]. First, a literature review was initiated to determine and uncover the state-of-the-art and research gaps in e-learning. The results formulated the research questions. The research strategy was a mixture of *design and creation* and *experiment*. The design and creation were the processes of creating a fully functional e-learning artifact that would allow the experiment to take part. The data generation was obtained by the experiment where the *documents* consisted of the quantitative data collected by the artifact and the *post-questionnaire* was the students' interpretation of the artifact after the experiment trial was over.

During the experiment phase, a one-month period in March 2023, two classes of first- (n=25) and second-year students (n=48) interacted with the e-learning platform. They freely chose exercises from a Java learning content provider and tailored the e-platform based on their preferences during onboarding. The data generated by the platform was used to answer Research Question 1 quantitatively by regression and parametric statistical tests. After the experiment phase, a post-questionnaire was sent out to the participants of the experiment, and the results were studied qualitatively to answer research question 2.
The chosen strategies, data generation, and analysis methods are highlighted in Figure 1.4.1 and further elaborated on in Chapter 4.4.

1.5 Contributions

This thesis proposes a novel e-learning platform: *Progresso*. Unlike other e-learning platforms, Progresso offers an optional competitive learning environment while still allowing a strong sense of autonomy, should the student prefer [21][22]. It does so by offering the students the freedom to customize their learning environment with selectable data

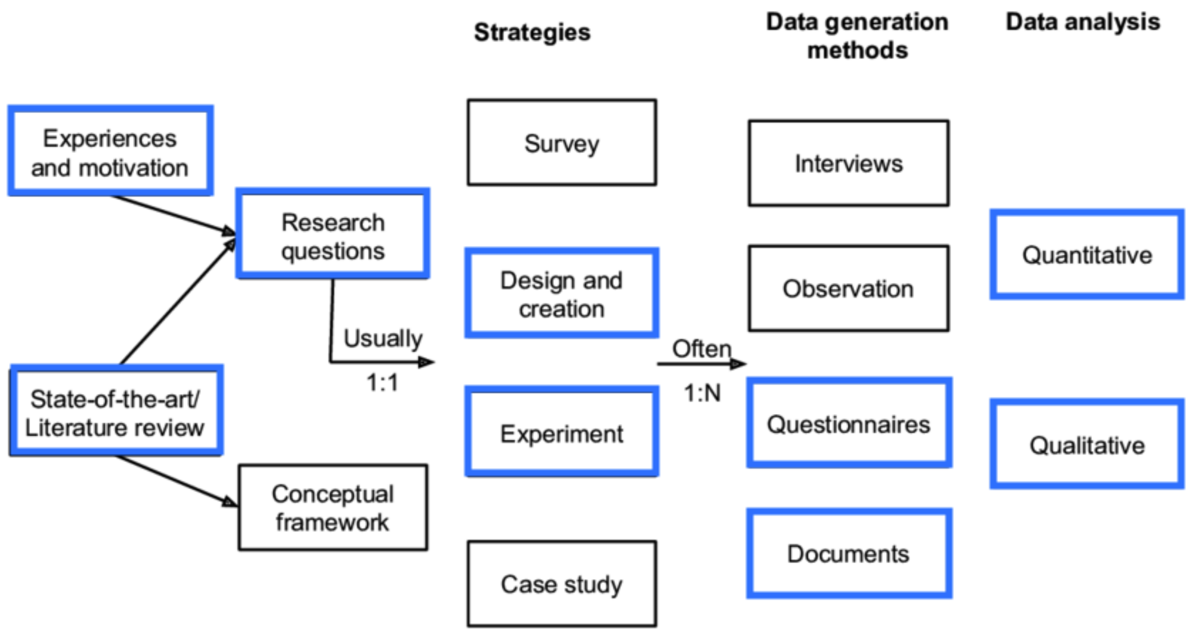


Figure 1.4.1: Oates' research process

visualization components. A leaderboard acts as the gamification element that fosters competition and, to some, a dimension to self-assess their performance relative to their peers. However, participation is not mandatory. Additionally, the study investigates potential correlations between student engagement, motivation, and performance data collected via the e-learning platform.

The study contributes with quantitative and qualitative empirical studies and a new customizable learner analytics dashboard with an optional competitive or non-competitive learning environment. Progresso stands out by integrating elements of self-regulation theory with a goal scheduling and planning component, while also incorporating principles of cognitive load theory by offering knowledge scaffolding in the learning content. The study also recommends new guidelines for successfully crafting learner analytics dashboards and, in detail, explains which web technologies are appropriate for e-learning. Finally, new approaches and directions are given for future research.

1.6 Thesis Outline

The overall outline of the thesis follows the process of answering the research questions. Chapter 2 Background & Related Work introduces the prior research on the topic. Chapter 3 Design & Implementation explains in detail the necessary steps and decisions during the development phase of the e-learning platform. Chapter 4 Methodology describes the method employed to answer the research questions either qualitatively or quantitatively.

It also describes what data aggregation and analysis methods were used. Chapter 5 Results presents the results of the experiment's quantitative data and questionnaire in detail. Chapter 6 Discussion interprets the findings and connects them to the literature review. Finally, Chapter 7 Conclusion & Future Work summarizes the results of the data analysis and projects what the future research might look like.

CHAPTER

TWO

BACKGROUND & RELATED WORK

This section aims to provide a comprehensive overview of the research area and current knowledge body. Section 2.1 presents relevant definitions of concepts related to the research area. Section 2.2 provides insight into relevant theories used to explain various aspects of human behaviour relevant to the research field. Lastly, section 2.3 presents a literature review with a thorough examination of the field and an outlining of previous work that together forms the established knowledge. Combined, the chapter serves as a foundation for understanding the context and significance of the study.

2.1 Definitions

The key definitions of important terms and concepts that form the basis of the research are represented in this section. Presenting these definitions aim to help the reader get a comprehensive understanding of the relevant terms used throughout the thesis.

2.1.1 Learning Technology

Learning technology - or educational technology - is the study and practice of facilitating learning and improving performance by creating, using, and managing appropriate technological processes and resources [23]. Learning technology is used in educational environments to assist and facilitate learning activities, enhance teaching strategies, and give students access to information and resources to help them realize their full learning potential [23].

Technology for learning has the potential to improve education's efficacy, efficiency, and accessibility [23]. An emerging research field of educational tehcnology is the integration of learner analytics [13]. This is a way of further enhancing educational practices, and can be done to further understand the activities of the learners and personalize the learning accordingly [13].

2.1.2 Learner Analytics

Learner analytics, as defined by Siemens in "Penetrating the Fog: Analytics in Learning and Education", refers to the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs [24]. It should help learners to take ownership and reflect upon their learning [25]. This field offers educators new opportunities to understand their students better and make more effective use of limited resources. The research community focused on learner analytics is expanding rapidly due to the increasing availability of resources gathered through Learning Management System (LMS)s [26].

2.2 Theories

This section aims to provide insight into human behavior by exploring two theories that examine various aspects of cognition, motivation, and development. The examination of these theories provide valuable perspectives that contribute to the research field.

2.2.1 Self-Regulation Theory

Self-Regulation Theory is a theory that focuses on how individuals can control their own thoughts in order to reach their desired goals [27]. It highlights the process in which individuals choose, monitor, and restructure their goals in order to succeed. It also explores how individuals can contribute to their own motivation, behavior, and development through self-regulation [27]. Research has found that students who have the ability to self-regulate often experience increased learning outcome [27]. However, most students struggle to apply self-regulation methods and would need assistance and guidance to develop these methods and be encouraged by others to not lose interest in the process [28].

2.2.2 Cognitive Load Theory

Cognitive Load Theory, originally introduced by Sweller in 1988, explores the limitations of our working memory's processing capacity [29]. Sweller emphasizes the importance of avoiding overloading the brain with irrelevant information that does not directly relate to the learning process. When our brain processes information, it utilizes structures called schemas to categorize it into long-term memory. This allows us to increase the amount of information that can be held at once [30]. The presence of well-developed schemas distinguishes experts from novices in problem-solving skills [29]. Introducing prerequisite skills before tackling a more complex theme will assist the learners in establishing these schemas, which could ease their learning of more difficult topics. It is crucial to identify the learner's level of expertise and provide them with suitable information at the appropriate level [31]. Additionally, minimizing the gap between the current level of knowledge and the desired goal is essential. One approach to narrow this gap is by providing examples and problems with partial solutions first before progressing to more challenging exercises [30]. Mousavi et al. suggest another approach; to incorporate auditory information alongside visual information, taking advantage of the working memory having independent processors for visual and auditory input [32].

2.3 Previous Research

This section offers a thorough examination of existing literature within the field, which highlights the current state of knowledge and identifies research gaps. To identify the relevant work, the researchers searched the following databases: *Google Scholar, Science Direct, IEEExplore, ACM* and *ERIC*, with the following search terms: *'e-learning', 'learning analytics', 'learner dashboard', 'gamification', 'leaderboard', 'customization', 'personalization', 'knowledge estimation'*, *learning outcome* and *'knowledge tracing'*. The literature review was conducted with a focus on sourcing papers published post-2019 whenever such recent materials were available.
After conducting the search with the chosen terms and databases, the top most relevant articles were briefly scanned and put in a list of potential candidates. During the next phase, the articles were thoroughly scanned for relevance, credibility, and validity. The remaining articles underwent a full review and are included in this section.

2.3.1 Predictors of Learning Outcome

The e-learning field has seen significant advancements in using data-driven methods to estimate the learning outcome and predict the skill mastery of students [13]. Predictive analysis of learning outcomes depends on many factors not yet fully understood [33]. Existing studies on predicting learning outcomes center around the concept of the knowledge space, largely drawing inferences from students' learning strategies and motivation [34][35][36]. Conversely, there is a noticeable gap in research evaluating and predicting students' skill mastery by the quantitative measures of research question 1.

Duration
Previous research indicates that a student's learning outcome significantly depends on the learning strategy they choose to employ [35][36]. Learning strategy is often contextualized via how much time the student dedicates to learning [35][37]. A weak correlation between study time and learning outcome has been found in several studies [37][14]. In contrast, another analysis concludes that the high-level quantity of study is a poor predictor of academic success [38]. In conclusion, the low-level time spent on an exercise designed to target elements of a more compound skill exhibits unique properties that are not often considered in traditional academic studies.

Learning Path
Another estimate that potentially can work as a predictor of learning outcome is the student's chosen learning path. Traditional teaching has typically followed a fixed sequence where they offer tasks and exercises to learners [17]. However, over the recent years,

researchers have given considerable attention to identifying the learning path that yields the highest learning outcome [17][7][18]. Basu et al. emphasize that no fixed learning path is suitable for every learner [18]. By finding the ideal learning path, the cognitive load of the students could be reduced, which could improve the efficiency and quality of the learning [7][1]. While previous studies have identified the significance of personalized learning paths, this study takes a novel perspective by examining the skill mastery of students who have chosen a specific start of their learning path. It could provide valuable insight to identify whether there is a significant difference between these students and their peers, and it could possibly reveal patterns, preferences, and strategies that contribute to their success. Are there any learning paths that are better suited to achieve a higher level of mastery, or is it necessary to personalize the learning paths to fit every student's needs?

Attempts
Another commonly used predictor to explain skill mastery and learning outcomes is the observation of the student's attempts on a sequence of exercises. Latent knowledge estimation observes knowledge as a latent variable by inferring the unobserved construct of student knowledge from observable data [9]. Various models have been developed and tested with promising results [1]. Traditional Elo rating used in determining skill in chess has been applied to determine skill mastery and then recommend coding exercises to that skill proficiency[8]. Bayesian Knowledge Tracing is another model that has been used in intelligent tutoring systems to predict the skill mastery of students based on their submitted correct and wrong attempts [9][10]. However, it is argued that the complexity of the exercise, the context in which it's given, and the instructional approach used can all significantly influence this data [39]

2.3.2 Predicting skill mastery with BKT

Bayesian Knowledge Tracing is a type of latent knowledge estimation that accepts a binary observation, as discussed above [10]. BKT utilizes a Hidden Markov Model (HMM) to predict skill mastery by the sequence of correct and wrong answers on exercises [9]. It aims to measure the changing knowledge state and estimate when the student transit from the not-learned state to the learned state at a time interval t. BKT infers the knowledge state by assuming the current knowledge state results from all prior observations. Figure 2.3.1 shows the Finite-State Machine (FSM) of an observation. The parameters at each observation are shown in Table 2.3.1. The model is inaccurate on compound knowledge, but satisfactory results have been seen on atomic skills associated with compound knowledge [9].

Standard BKT makes several assumptions about latent student knowledge. First of all, when the skill is first learned, it is never forgotten (state k cannot transition from 1 to 0). Secondly, the relationship between the known and not-known states is binary. Thirdly, skills with overlapping domains are not accounted for, meaning the student can not combine different skills when applying themselves to a new skill. Some variants of BKT try to mitigate these assumptions. Mohammad Khajak et al. propose BKT+Forget (BKT+F)

Probability	**Description**
$p(L_0)$	The initial probability of the student mastering the skill before any attempts.
$p(t)$	The probability of a student transitioning from not known to know after an opportunity to apply a skill
$p(s)$	The probability of the student answers incorrectly, despite knowing the skill (slip)
$p(g)$	The probability that the student answered correctly, despite not knowing the skill (guess)

Table 2.3.1: BKT Variables and Descriptions

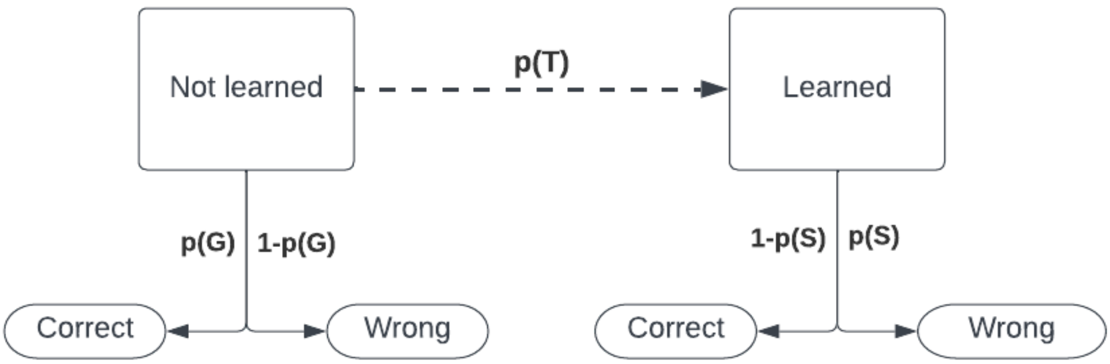
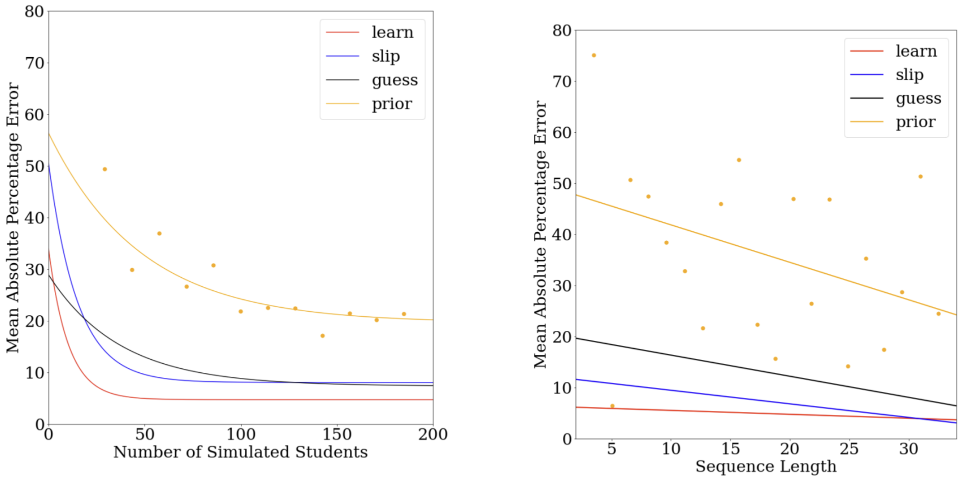


Figure 2.3.1: The finite state machine in Knowledge tracing

that accommodates the probability of the student forgetting a skill since the last time they practiced [40][41]. Knowledge Tracing - Item Difficulty Effect Model (KT-IDEM) proposes deriving guess and slip parameters from the exercise observations rather than student observations. When the model predicts skill mastery after an observation, the guess and slip parameters belong to the exercise's past interactions/observations. In a comprehensive evaluation, this approach proved superior to the standard BKT model in 9 out of 10 datasets [42]. The significance of item order has shown promise to determine the overlap between skills [43][44]. Lastly, Knowledge Tracing - Prior Per student (KT-PPS) proposes individualized BKT prior parameters [45].

pyBKT is a Python library used to model student knowledge by intertwining traditional BKT and optionally any of the models mentioned above [11]. The authors of pyBKT determined that 15 was a good sequence length to reduce worst-case mastery estimation accuracy and that 50 was a "reasonable sample size to achieve convergence to canonical parameter values with any average student sequence length" [11]. There seems to be less benefit in reducing model fitting error by increasing the sequence length than in increasing the sample size in Figure 2.3.2. The prior parameter is more sensitive to error at sample sizes less than 50. Moreover, the mastery prediction accuracy is asymptotic at around 15 in Figure 2.3.3. Both Figure 2.3.2 and 2.3.3 are from the original pyBKT paper [11].



(a) MAPE of parameter vs sample size **(b)** MAPE of parameter vs sequence length

Figure 2.3.2: Reported MAPE by the authors of pyBKT

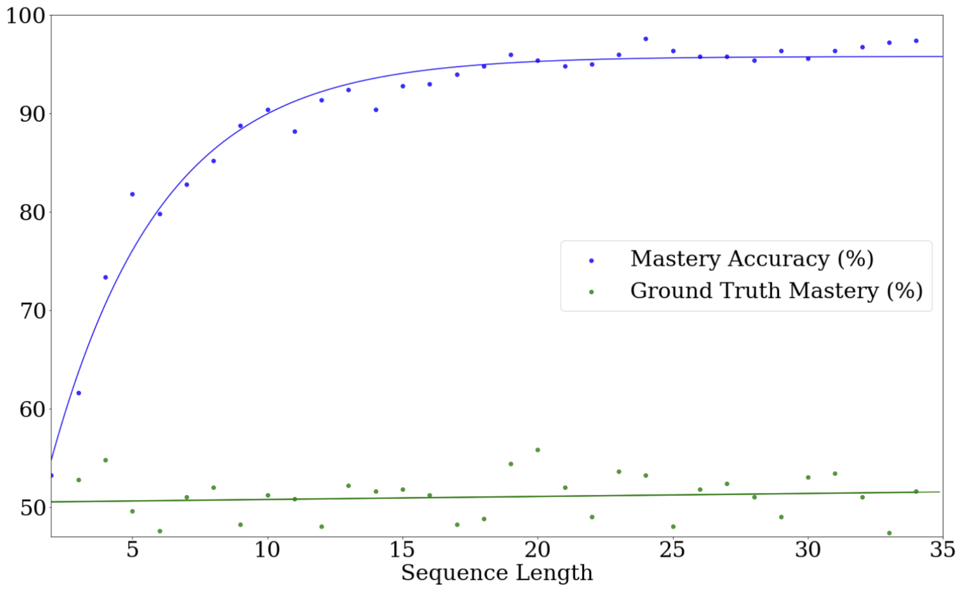


Figure 2.3.3: Accuracy of mastery prediction vs. sequence length reported by pyBKT authors

2.3.3 Learner centered design

There has been a growing emphasis on student-centered design in education in recent years [5]. This approach acknowledges that learners' needs and preferences vary and that engagement and motivation are essential to successful learning outcomes [5]. Traditionally, learner analytics dashboards have been designed with university staff members as the primary audience with minimal emphasis given to actively involving students [46]. According to Cho et al., a well-designed user interface can mean the difference between continued usage and technology rejection [47]. User-friendliness, clear instructions, purposeful structure, and satisfactory design are all critical aspects [47]. In designing effective learning dashboards, it is vital to consider the needs and preferences of the learners themselves. Student engagement is crucial to develop practical and useful dashboards [5].

Roberts et al. identify features that students would want in a dashboard. Some of the most important are; study habits, data on the course (such as difficulty), and feedback on how well they perform compared to their peers. In the latter, privacy was a crucial element [5]. They were also interested in seeing general statistics on their performance. Park et al. found that all participants were positive regarding statistics about their scores [22]. Vesin et al., in their work with an intelligent tutoring system, ProTuS, found that the users wanted visualizations like pie charts. They would also want the system to visualize their aggregated progression over time and to see the percentage of improvement [21].

"Confronting" dashboards emphasizing subpar performance should be avoided [5]. According to Schumacher et al., students would want reminders of deadlines and to-do lists, analyses of their current knowledge levels, and recommendations for effective learning. They were divided on whether they wanted analyses comparing them to their peers [6].

Personalization refers to any pedagogical action considering the student's individual and personal learning needs [19]. Notably, recent studies have measured the effects of learning with personalization tools that foster self-awareness, self-assessment, and autonomous learning [21]. Personalization also encompasses customization by allowing students to create self-designed learning environments that accommodate their individual needs [19][48][49]. Customization embodies a variety of metrics, but recent studies have primarily focused on how customization heavily impacts the students' learning activity over the course of e-learning engagement [19]. In addition, the student is positively stimulated by showing detailed information about their learning activity trajectory [49][19].

Customization of dashboard features could ensure that students have access to the information and tools most relevant and valuable to them. By enabling learners to customize their dashboards, they can avoid spending excessive time searching for desired functionality or information. Providing the learners with display elements keeps them interested and increases their motivation [47]. Several studies found that students would be interested in being able to customize their dashboard [5][6]. However, they state whether customization increases perceived academic control and leads to positive learning benefits should be further researched [5].

It is helpful to employ measures such as the System Usability Scale (SUS) to determine the effectiveness of learning dashboards. SUS is a widely used tool for evaluating software systems' usability and user experience [50]. Whether the system is perceived as useful and easy to use by the learners is crucial to avoid market rejection [47]. By using such metrics, researchers and educators can gain valuable insights into the efficacy of various dashboard designs and features and pinpoint areas for improvement coming from the students themselves.

The design of learning dashboards has significantly improved over recent years [5], but there are still some questions about what the "right" information is to show to the learners and the best way to visualize it. In addition, few studies have investigated the relationship between visualizations and how students react and understand them [22].

2.3.4 Gamification

Gamification is a term that has gained significant attention in recent years due to its potential to enhance engagement and motivation in various contexts [51]. Simply put, gamification refers to using game design elements and mechanics in non-game contexts [52]. It involves using concepts like competition, prizes, feedback, and progression to enhance the motivation and engagement of non-game activities. A significant portion of studies reports positive learning outcomes from using gamification [53][54]. Visualizing in the form of progress bars is seen as motivating and is linked to goal achievement [54]. The type of social interaction that occurs due to gamification, e.g. collaboration and competition, is regarded as essential factors [55].

Competetive learning environments can significantly increase the attentiveness, social pressure [55] and engagement of students [54][56]. Leaderboards specifically can be seen not only as competition but also as a form of comparing your performance to others and receiving feedback on your performance [54]. Park et al. found that students perceived it as useful to be able to compare their positions with their peers [22]. Souza et al. pointed out that not only are leaderboards useful for comparing oneself to peers but also for social recognition [57]. Armstrong et al. concluded that gamification elements make the learning process more satisfying, however, with equal knowledge outcome [58]. Vesin et al. suggest that incorporating a competition module into the dashboard can allow students to compare their skill levels and help achieve interactivity [21].

Despite its potential benefits, gamification has its criticisms and challenges, and several studies in a literature review reported null or mixed results [53]. Critics argue that gamification may not always lead to sustained engagement or meaningful behavior change, and that it can impose unnecessary competition that distracts from learning [15]. Additionally, it is argued that gamification elements shift the focus from actual learning to just passing the tests or getting a high score [56]. Competitive elements can also work destructively by causing feelings of irrelevance and oppression [55]. They might affect different students in severely different ways, for example, if they occur between learners with widely different skills [55]. Individuals also have different learning styles, which could affect the gamification experience [53] as well as their prior experience and attitude towards gamification to begin with [52][59]. Performance may also affect the enjoyment of competition, especially leaderboards [54]. Cheong et al. state that some students may feel uneasy about being identified as a low performer on a leaderboard [54]. As a solution to this issue, Bai et al. suggest that the instructor should consider only releasing the top-performing students on the leaderboard [16]. They also found that students ranked in the bottom third of a leaderboard prefer to be anonymous. Kim et al. conclude that gamification elements probably work best as supplementary tools rather than replacements [59]. Studies also emphasize the importance of the design because gamification itself does not automatically generate motivation and engagement [59][54]. There have been considerable research efforts in the gamification field. However, evidence of the effectiveness has yet to be provided [55][60].

Gamification can also be viewed in the context of learning outcome [51]. For this particular study, we will look at leaderboards and their effects on learning outcome. While previous studies have primarily focused on the overall effects of leaderboards, less attention has been given to leaderboards' effect on learning outcome and course engagement [16]. According to Bai et al., students in the top third of the leaderboard demonstrated better learning performance than their peers, along with higher levels of engagement [16]. However, the study was conducted in an East Asian culture where attitudes on public comparison and saving face are different [16].

CHAPTER

THREE

DESIGN & IMPLEMENTATION

The research questions outlined in Section 1.2 are characterized by two significant research inquiries. The first question explores the influence of several predictors of skill mastery. The second question is dedicated to understanding how to optimize the design of the learner analytic dashboard to improve its perceived usefulness and, by extension, increase student engagement. As such, an application must be designed that accommodates functionality that allows the RQs to be answered. Throughout the implementation phase, these research inquiries would serve as guiding principles, shaping the project's trajectory and informing the iterative development and refinement of the application. This chapter introduces the development and design process of *Progresso*. Progresso aims to create a fully functional, customizable learner analytics dashboard. The team aimed to reach an application state beyond a minimum viable product. As part of two university courses, Progresso should be functioning seamlessly, capable of efficiently managing and facilitating resources for two distinct university classes.

3.1 Stakeholders

The stakeholder is a party that has a stake interest in or stands to be influenced by the development, operations, and outcomes of Progresso.

Students
Students are the primary users of Progresso. It should offer a user-friendly interface to perform and review coding exercises and self-assess their performance. Moreover, the platform should entail an intuitive and engaging learning environment that prevents students from becoming frustrated with using the application.

Teachers
The teacher is the connection between the development team and the students. The teacher's stake in Progresso is that they must vouch for Progresso as an effective learning tool. Consequently, the effective orchestration of the teacher/development team relation-

ship constitutes a pivotal element in the successful development of Progresso, given its potential to impact the project's strategic and operational decisions.

Development Team
As the application's creators, the development team maintains, updates, and improves the system based on students' feedback and technical requirements. They must ensure the application's reliability, security, and scalability. It is also required of the development team to create something that is desirable for the aforementioned stakeholders. As such, they must be thoroughly educated on the requirements the teacher has for the application and understand the needs and preferences of students.

3.2 Functional Requirements

The functional requirements were crafted with the research questions in mind. Based on the proposed functional requirements in the thesis description (Appendix A), the team created a list of functional requirements during the starting phase of the project. The functional requirements capture the intended behavior of the system and refer to the specific functions and capabilities the application is required to fulfill to perform [61]. The features were prioritized in order of importance, five being the most important making out the Minimum Viable Product (MVP) and 1 being the least important. Each requirement was also given a complexity scaling from 1 to 5 based on how complex it would be to implement. The final list can be seen in Table 3.3.1.

3.3 Non-functional requirements

A non-functional requirement is a characteristic of a system used to evaluate if the system fulfills the expectations of its stakeholders beyond its primary function. [62, p 59]. The three most important quality attributes for the project were identified early on in the planning stage of the project and are listed below. The quality requirements Modifibaility, Scalability, and Deployability were identified as most important for successfully crafting Progresso. The quality requirements were necessary to give the team the best possible foundation for completing the functional requirements in Section 3.1, but also the integrity of the application during usage as well as future development.

Modifiability
Modifiability is a quality attribute proposed by Bass et. al [62, p 154]. Modifiability was decided as a key non-functional requirement to minimize technical debt. First of all, the team had nowhere near a complete plan for the application early on. Identified changes could occur throughout the development phase. As such, the modifiability attribute allowed the team to adapt the application to a changing environment. Moreover, the supervisor informed the team that it was highly likely that Progresso would be further developed in a future master's thesis.

ID	Requirement	Description	Priority	Complexity
1	Authentication	Access control. Log in and register	5	3
2	Customization	The users should be able to choose the content they want to see on their dashboards	5	3
3	Programming exercises	The user should have access to Java exercises	5	5
4	Skillmastery prediction	Users should be notified when a skill is considered learned	5	5
5	Feedback	Users should continuously receive feedback on their progress	4	4
6	Leaderboard	The user should be able to opt in and out of the leaderboard	4	3
7	LTI Integration	When the system grades the student, the grade should be reported to the LMS	4	5
8	Progress Analytics	The user should be able to see their own progress, results and visited/solved content	4	4
9	Recommended content	The user should be served recommended content	4	3
10	Badges/rewards	Users should be rewarded when they progress	3	2
11	Admin	The teacher should add/remove learning content, and manage users	2	4
12	Planning tool	Users should be able to plan their work and set deadlines	2	1
13	Communication Options	Users should be able to communicate between exercises	1	5

Table 3.3.1: Functional requirements with priority and complexity

Scalability
The team aspired to create an application not solely dedicated to the test-phase, but one that could also stand alone and be adaptable to a wide range of similar user scenarios. The contrary would be to make a "hard-coded" application that would not adapt to any scenario other than the test phase applicable to this master thesis. Scalability is achieved using technologies that rapidly accommodate new changes or scale based on demand [62] and the technology stack in Section 3.5.2 was decided because of their high scalability attributes.

Deployability
The team ensured a strong emphasis on the quality attribute deployability. By designing the software architecture around deployability, the team could deploy an efficient deployment pipeline that promoted new code changes to production in case of bugs reported by the students, logs, or other bug discovery procedures. The team anticipated that during the live test phase of Progresso, certain mechanics of the application uncaught by prior acceptance testing would impact the user experience. In such events, it is time critical to focus on resolving the issue and not become distracted by a development environment unsuited for fast-paced changes.

3.4 Development Tools

The development of a project requires various tools which play a critical role in achieving the desired result. In this section, the chosen means are presented.

3.4.1 Figma

Figma is a web-based design tool that allows users to create high-fidelity wireframes, mockups, and prototypes [63]. This tool was used to quickly create, test, and change a functioning prototype.

3.4.2 Craft

Craft is a web-based project management tool that allows teams to efficiently organize tasks and documents and create to-do lists [64]. The collaborative features made it easy for the team to collaborate, share feedback, and stay informed about the project's status.

3.4.3 Miro

Miro is a collaborative online whiteboard platform that is particularly useful for designers who need to brainstorm, ideate, and collaborate on projects with team members [65]. This was particularly useful in the starting phase of the project and was used to brainstorm and ideate, as well as after the user interface test phase to organize the feedback.

3.4.4 GitHub

During the development phase of the project, we utilized the collaborative software development platform GitHub to facilitate team communication as well as version control [66]. GitHub allowed the team to store and manage our code and documentation, making it simple to track changes and collaborate on code development.

3.5 Software Architecture

This section explains the architecture chosen to successfully implement the application proposed by the functional and non-functional requirements. Naturally, several important architectural decisions were made through the seven-month-long development phase. However, only critical decisions relevant to the application's core functionality and required by the functional requirements are detailed in this chapter. Before diving into the architecture, it is recommended that the reader familiarizes themselves with the data models in Appendix F as they are frequently referenced in this section. Figure 3.5.1 shows a holistic component diagram that facilitates a clear visual representation of the interactions and relationships between the various components.

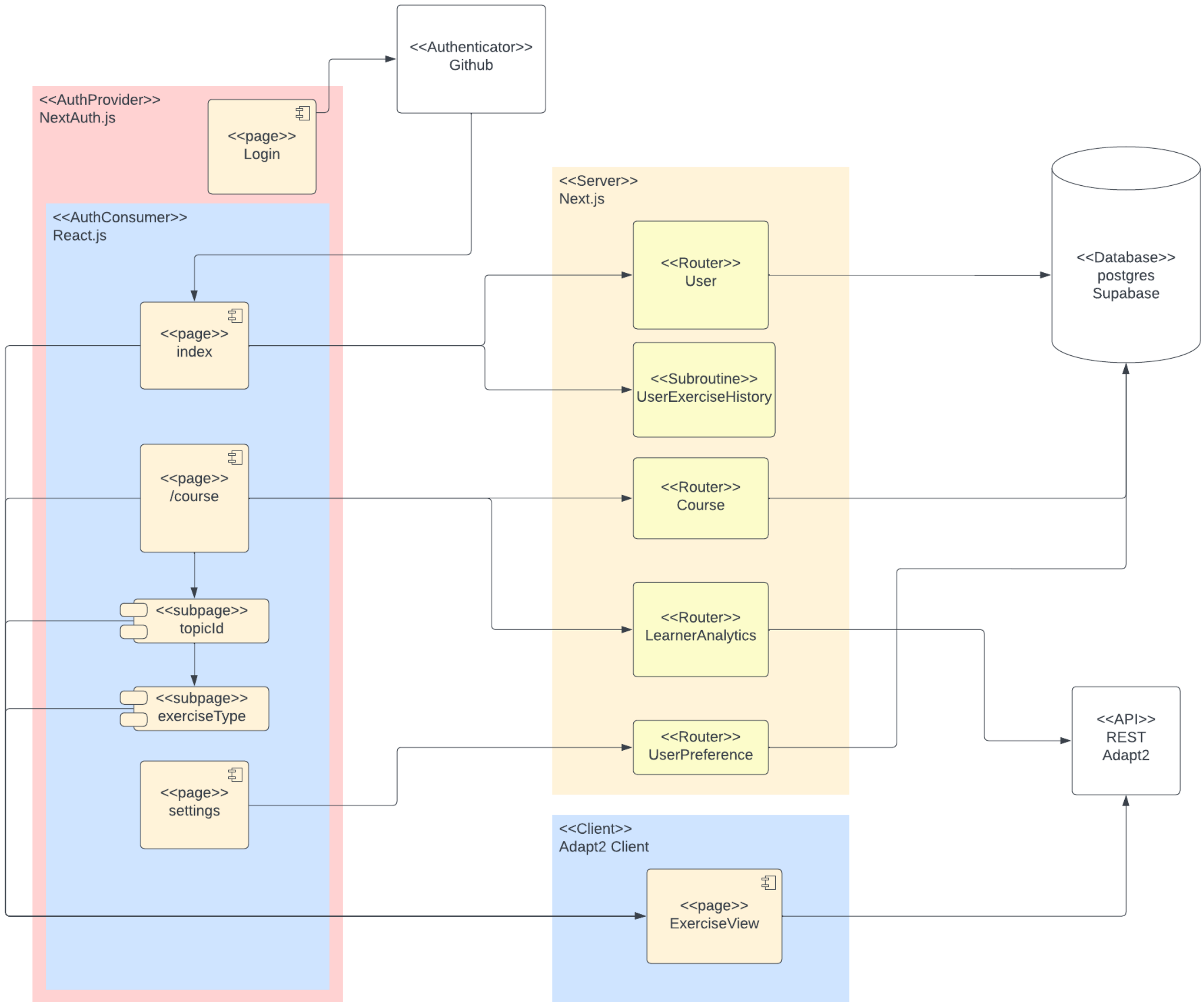


Figure 3.5.1: Component diagram of the application

3.5.1 Learning content provider

Functional Requirement 3 required that the user should be served with Java exercises. Adapt2 is a learning content provider made mandatory by the supervisors. Adapt2 is an API that serves exercises across multiple different programming languages [67]. Adapt2 tracks the user's interaction with the exercise, such as the number of attempts, success rate, and the sequence of failure and success. The exercise repository consists of Java exercise content across 14 different topics. The exercises are divided into three types, examples, challenges, and coding exercises. Examples do not test knowledge but explain the topic through code examples. The student can initiate a walkthrough of the code by clicking the next or previous buttons. See Figure 3.5.2. The challenge type is a multiple-choice question where the student must drag the correct code into a preexisting code block, see Figure 3.5.3. The coding exercise gives the student a problem, and the student writes the code themselves. The compiler runs a set of unit tests and determines whether the attempt was incorrect, partially correct, or correct. See Figure 3.5.4.

The API stores information on the actions and interactions the user has performed on the tasks. Most important is the sequence of correct and incorrect attempts for the exercise, indicating "0" as incorrect or partially correct and "1" as correct. Partially correct is considered incorrect. A student's sequence of "0001" means they got the exercise correct on the fourth try. The analytics of the API is hosted in the US, and the University of Pittsburgh provides challenges and examples. The University of Toronto provides the coding exercises. The API is divided into two, depending on which parameters are appended to the URL. The team used two versions. One is the analytics part storing user interaction, progress, and information. The other part is static information about the topics and exercises.

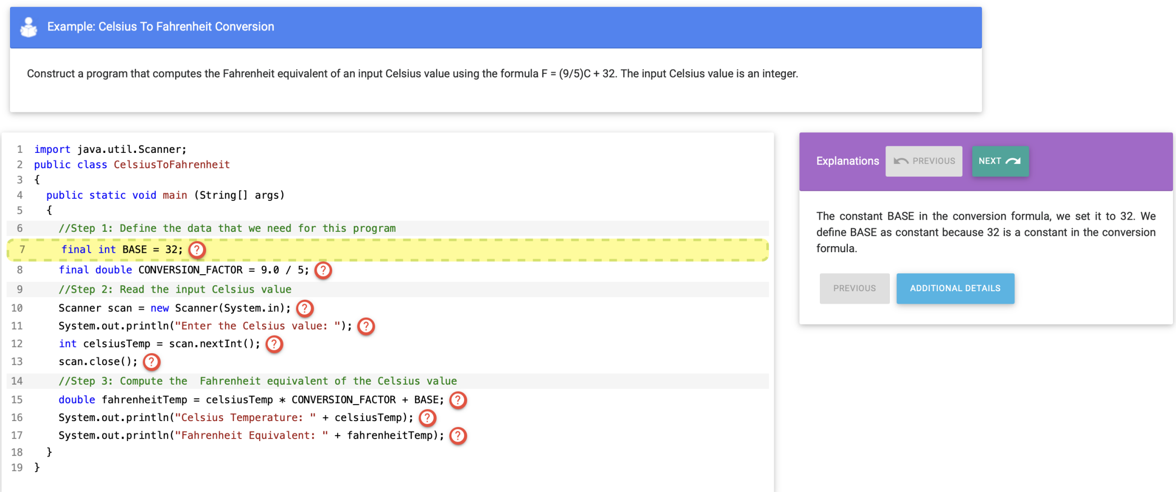


Figure 3.5.2: Example of an example exercise

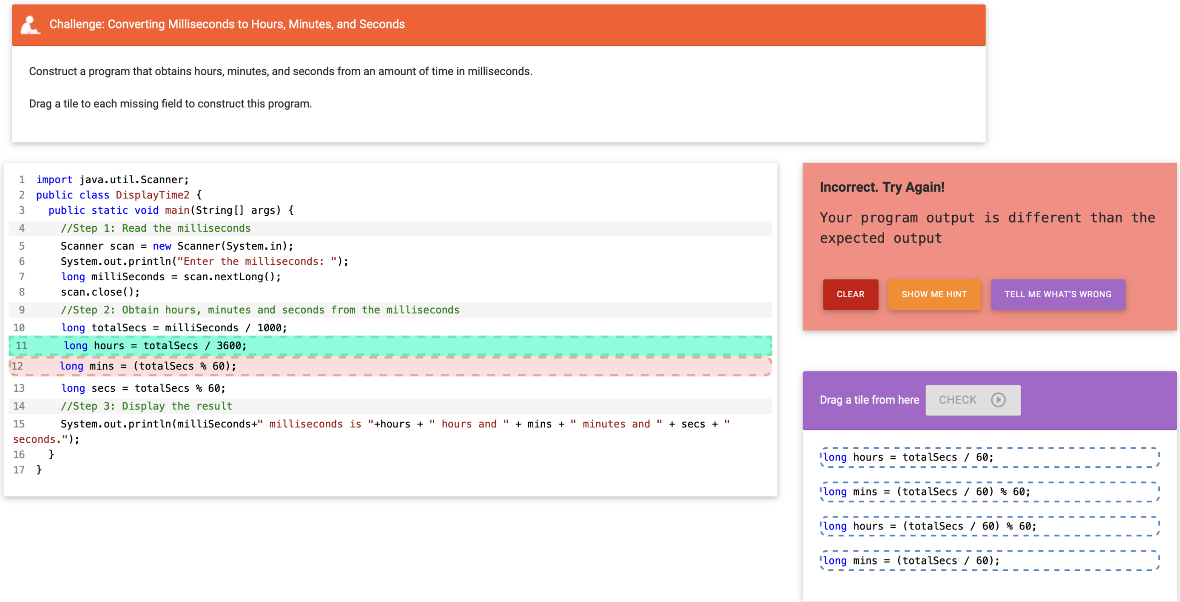


Figure 3.5.3: Example of a challenge exercise

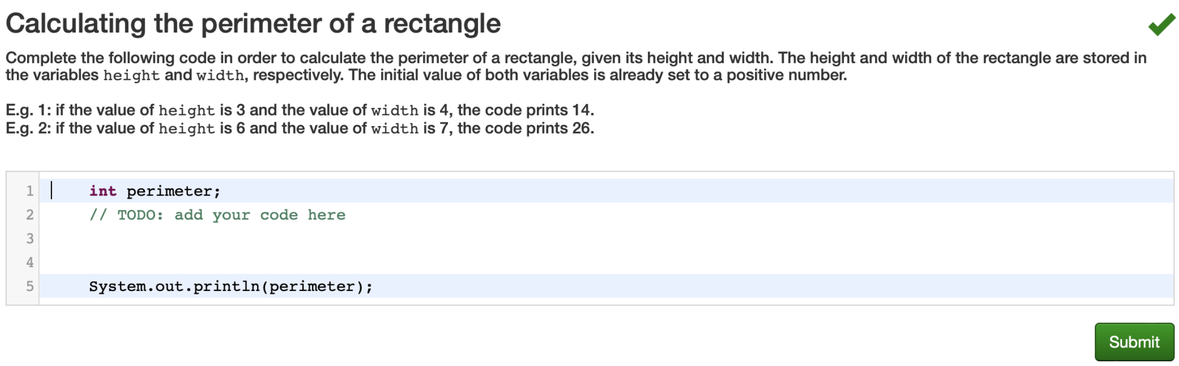


Figure 3.5.4: Example of a coding exercise

One drawback of adapt2 was its cold-boot time of $\approx$ 2 seconds. As a solution, static information of the API, such as courses, topics, and exercises, was extracted and stored in a Postgres database provided by the open-source database project Supabase [68]. The database and frontend application was co-located in Germany to reduce propagation delay between client and server. Another flaw of adapt2 is its Object key-value structure. The key-value pairs do not conform to the JSON object literal standards defined by RFC8259 [69]. As a consequence, the API was not parseable for Progresso. The solution was to use an alternative version of JSON that accepts object keys as bare strings (without quotes) [70]. As a result, Progresso can also accept partially broken learner APIs.

On-task duration is a dimension necessary to answer RQ1.2. Later in the development phase, it became apparent that adapt2 lacked this critical feature. To store the timestamp of when an exercise was opened and completed, the team implemented a caching technique that would also apply to any other API that does not store timestamps on event changes. Figure 3.5.5 shows the events of how the client determines when to save the *completedAt* field after a user submits a successful exercise attempt in a sequence diagram. Appendix E shows the code relevant to the technique.

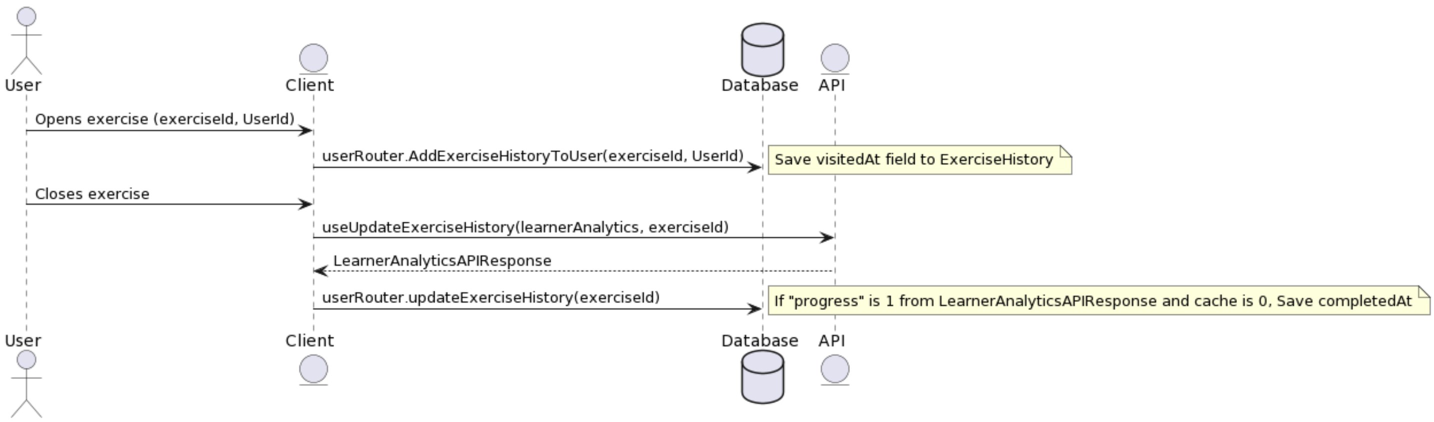


Figure 3.5.5: Sequence diagram of caching technique

Some limitations were apparent with this technique. Firstly, it created a lot of business logic on the client, which ideally could have been solved on the server. Secondly, if the user completes the exercise several days later, the time interval between *visitedAt* and *completedAt* would not be accurate. This is further discussed in the Application limitations in Section 6.4.2.

3.5.2 Infrastructure

The team created an infrastructure that was highly compliant with the non-functional requirements of scalability, deployability and modifiability. Vercel, a cloud hosting deployment provider, was used to deploy the application to production quickly. This platform provided an instant, automated, and scalable deployment pipeline. Vercel's unique capability to host both frontend and serverless functions drastically simplified the deployment process. It also ensured a faster *time-to-market*, with a developer-friendly workflow that allowed for seamless handling of deployment tasks. Moreover, using Vercel's deployment model greatly satisfied the non-functional requirement of modifiability. With its automatic CI/CD pipelines, any updates or modifications to the application could be easily incorporated and pushed to production. In addition, Vercel's intuitive developer tools, such as the Preview Deployment feature, provided a sandboxed environment for testing and validating modifications before deploying them to production. This reinforced the team's ability to deliver a reliable, robust, and highly modifiable application. One can argue that the automatic pipeline also fulfilled the deployability requirement because it quickly made code changes available to the production environment. Figure 3.5.6 shows the sequence diagram of the deployment flow when code changes are pushed to the *main* branch of the source code on git. Vercel as a deployment provider also satisfied the scalability non-functional requirement. Vercel ran the application "on the edge", meaning geographically close to the end-user.

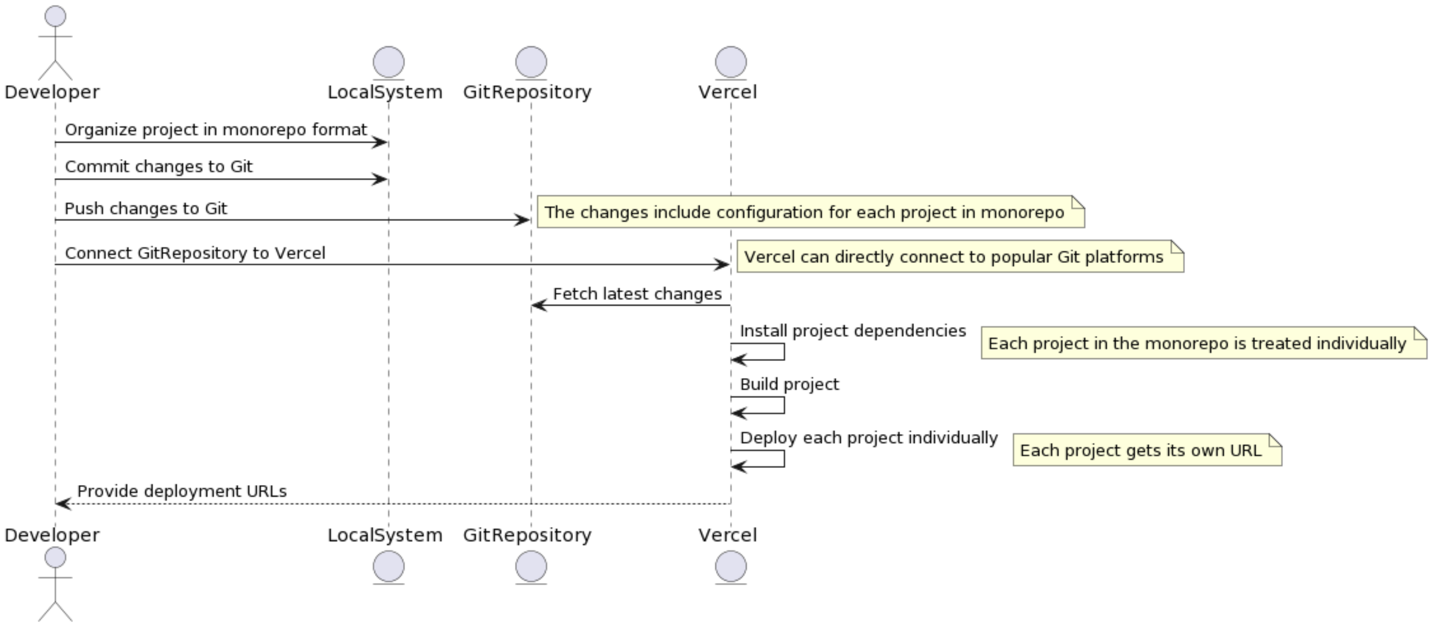


Figure 3.5.6: Sequence diagram of source code to production

3.6 Technology stack

The team had solid prior experience in different web technologies from NTNU courses, internships, and part-time jobs in the industry. The team wanted to build on existing knowledge but simultaneously challenge themselves with new technologies. The non-functional requirements were especially important when picking the technology stack. Recently, a web technology stack called T3 has seen an increasingly high adoption rate in the open-source community [71]. The foundation consists of Next.js, Prisma, tRPC, Zod, and Tailwind CSS. Next.js is a popular open-source framework for building server-side rendered React applications with built-in routing and static site generation features [72]. Prisma ORM is a powerful database toolkit that simplifies database access and manipulation, providing an intuitive abstraction layer of SQL for querying, writing, and migrating data in various database systems [73]. tRPC is a lightweight and efficient TypeScript framework that simplifies building robust and type-safe APIs by automatically generating client and server code, reducing boilerplate, and improving developer productivity [74]. Zod is a TypeScript-first runtime validation library that enables developers to define and enforce strict data schemas and type validations for runtime data integrity and consistency [75]. Finally, Tailwind CSS is a highly customizable, utility-first CSS framework for rapidly building custom user interfaces in web applications [76].

3.6.1 Next.js

Next.js is a full-stack framework that integrates a frontend written in React.js and a server in typescript. The main features are the routing and page protocols that allow the developer to handle response and request objects between the client and server without writing independent front- and backends. These attributes solidify the modifiability non-functional requirement because the developer can easily adapt and change the application based on requirements or new features.

The most distinguished pattern in Next.js that inherited concepts from the modifiability quality attribute was the consumer-provider pattern in the React Context API [77]. Contexts were used extensively due to their state-sharing nature across frontend components. Most notable was the session object reference in multiple places on both client and server sides. Instead of passing the session object through multiple layers in the data hierarchy, known as "prop drilling", the session was available client-wide. The session object contained helpful information about the currently logged-in user. Moreover, the session context was made available on the server instead of checking that the user was authorized in every client component. If the client queried a protected procedure on the router and the user was unauthorized, it would redirect to the login endpoint. Frontend components that rely on information from the session are regarded as consumers, and the server that sends the session information is the provider.

New features do not have to consider the authorization layer when adding future requirements. The developer must only consider whether the fetch data is sensitive and should be in a protected router procedure. This is a solid advantage compared to other applications that have to check for a valid session each time data is fetched.

3.6.2 Prisma

Additionally, the team recognized the need for an effective database management system. Prisma is an object-relational mapper that provides its own API for CRUD operations. Database entities are declared in a *schema.prisma* file and can automatically generate types for each entity and DTOs for the different CRUD operations. In contrast, traditional SQL must be manually written, and type inference would not be available automatically. Prisma is a solid contributor to the modifiability requirement because the automatic generation allows adding of new database entities with rapid integration into the code base.

Users were onboarded by signing in to the application using their GitHub account. This action created a User record in the database with their email. The team sent the necessary onboarding instructions with a unique ID and invitation to participate in the research project to this email. As such, **no sensitive user data** was given to the team without the user's consent.

Prisma supports several popular database systems, such as Postgres [73]. Supabase was decided as the database provider because of its generous free tier plan, which included statistics on database usage, but most importantly, table editors for all entities used by the application [68]. The team could monitor the data to ensure that satisfactory data was generated for the data analysis phase after the test-phase.

3.6.3 Typesafe Remote Procedure Calls (tRPC)

Type safety refers to enforcing strict type checking and ensuring that operations are performed only on values of appropriate types [78]. Typescript provides this feature to catch type-related errors at compile-time or runtime and prevent unintended behavior or bugs caused by incompatible types. In line with this, tRPC is a data fetching framework written in Typescript [74]. tRPC utilizes typescript types to ensure static types at compile time. This is most notable at runtime when honoring the API contract between the client and server or server-to-server using a typesafe implementation. In the context of Progresso, tRPC was selected to achieve static typing of analytics data between the adapt2 API and the Progresso backend, and between the Progresso backend and frontend. tRPC accomplishes this by using a Zod schema, which enforces static type inference and parses the JSON data obtained from the API into an object of a specified type. Should the JSON data not conform to the expected type, the server identifies and flags the incompatible key-value pair. This error detection mechanism ensures that the frontend inherits the data types from the Progresso server through static type inference.

Web applications like Progresso often grapple with the challenge of synchronizing the state between the client and server. This is especially significant in Progresso, which features real-time user analytics data. To tackle this, tRPC employs a batching technique that optimizes network requests by grouping multiple requests into a single batch [74]. This method not only dramatically reduces overhead but also enhances the user experience. The different data visualization components in the dashboard may request information on the server from different origins in the application. Two solutions were present for this issue. The first solution was to fetch the data in the parent component and pass the data as "props" to the child components. If the child components relied on different parts of the data or an aggregated state, these altercations had to be done in the child component. The other solution was to make the child components' data independent and let them call their data directly from a procedure exposed by the router. The benefit of the first solution is that a single network call is made for all child components. However, if the data became stale for any child components, the data had to be invalidated and, as such, trigger a loading state in all other components. The second solution was best for Progresso because it prevents unnecessary layout shifts and rerendering of the DOM. This solution also reduces the impact of the high round-trip time of the learner analytics from the learning content provider.

Figures 3.6.1 and 3.6.2 offer insights into how Zod schema and tRPC respectively contribute to ensuring type safety and improving modifiability. They highlight the strength of the tRPC framework in managing data requests, batching, caching, and error detection, thus significantly enhancing the user experience. Figure 3.6.1 shows how a single exercise is checked for conformity should the adapt2 object structure suddenly change in the future. Figure 3.6.2 shows a protected procedure that passes a data-transferable object. The procedure stores the time and date when a user completes an exercise.

```
export const activitySchema = z.object({
  relatedTopic: z.string(),
  activityName: z.string(),
  activityId: z.string(),
  url: z.string(),
  visited: z.boolean(),
  attempts: z.number(),
  successRate: z.number(),
  t: z.number(),
  aSeq: z.string(),
  sequencing: z.number(),
  type: z.enum([type.EXAMPLE, type.CODING,
type.CHALLENGE]),
})
```

Figure 3.6.1: An *activitySchema* validates the exercises structure

```
updateExerciseHistory: protectedProcedure
    .input(z.object({ activityId: z.string() }))
    .mutation(async ({ ctx, input }) => {
      const activityAnalytics = (
        await learnerActivity(ctx.prisma, ctx.session.user)
      ).activityAnalytics;

      const attempts = [
        ...activityAnalytics.challenges,
        ...activityAnalytics.examples,
        ...activityAnalytics.coding,
      ].find((e) => e.activityId === input.activityId)?.attempts;

      return await ctx.prisma.exerciseHistory.update({
        where: {
          userExerciseHistoryOnActivityResource: {
            userId: ctx.session.user.id,
            activityResourceId: input.activityId,
          },
        },
        data: {
          completedAt: new Date(),
          attempts: attempts,
        },
      });
    }),
```

Figure 3.6.2: A *protectedProcedure* passes a data-transferable object. The procedure is used to store the time and date when a user completes an exercise

In summary, Figure 3.6.2 exemplifies how easy it is to distinguish between *publicProcedure* and *protectedProcedure*. It is only a matter of changing the procedure object. Line 1 declares that the *updateExerciseHistory* is a *protectedProcedure*, meaning that the user must be authenticated to invoke the function. For convenience, line 4 calls the *learnerAnalytics* router to fetch the user's attempts on the exercise so we have the duration and attempt count stored in the Progresso database.

3.6.4 Tailwind CSS

As part of the T3 stack, Tailwind CSS is a highly customizable and utility-first framework [76]. The framework proved to be an invaluable component of the project's front-end development because it made it possible to swiftly compose the UI components and achieve the desired visual style. The extensive range of utility classes provided by Tailwind enabled the team to fine-tune the visual style of each component with precision, as well as meet every design requirement.

3.7 Design

This section elaborates on the design phase of the development, which started off with pen and paper sketches and ended up with a fully functioning product. The section is divided into subsections based on the design iterations. Based on the concept of customization emphasized in the background and according to Functional Requirement 2, customizable components were designed. In Section 3.8, the rationale for these customizable components is presented.

3.7.1 Low fidelity sketches

During the first iteration, the team drew low-fidelity user interface sketches. These sketches were made using pen and paper and worked as a minimum-effort starting point to visualize and validate ideas that could be easily changed. Seeing that the requirements of the dashboard changed throughout the project, not all components were made at this stage but were developed continuously throughout the iterations. An example of one of the low-fidelity sketches can be seen in Figure 3.7.1.

No user testing was done at this stage due to the project's time limitations, but the researchers used the sketches to analyze perceived usage and quickly draw up and alter ideas.

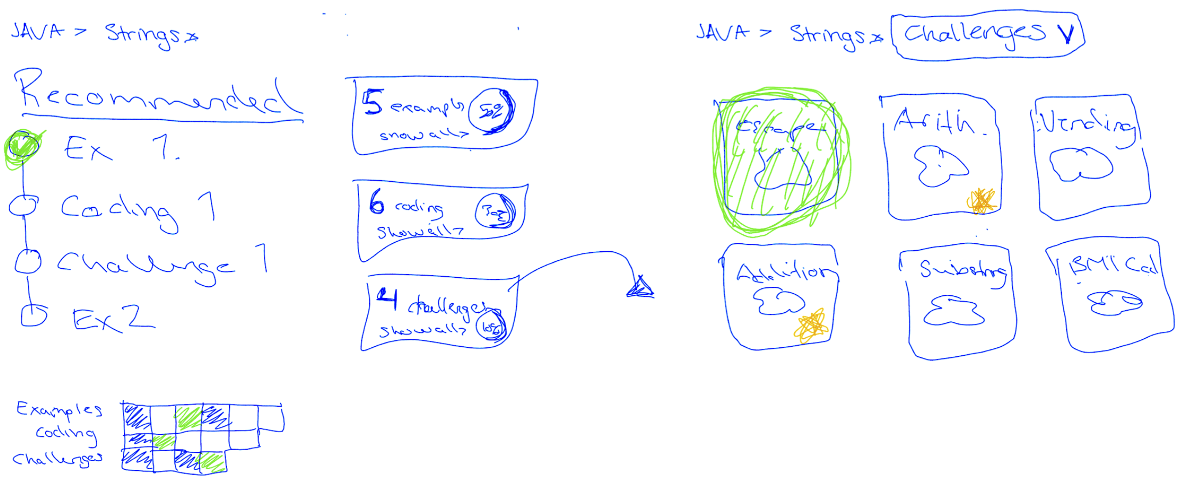


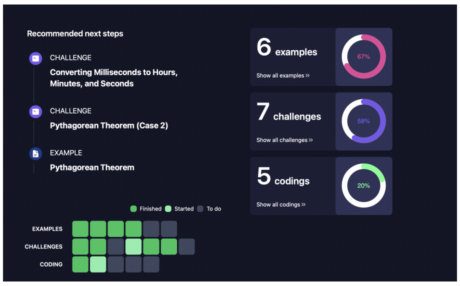
Figure 3.7.1: Low fidelity sketch of the topic page

3.7.2 Proof of Concept

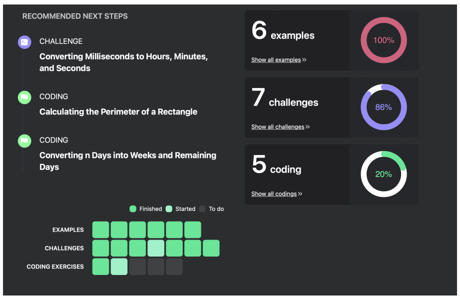
During the next design development phase, Figma was used to create a prototype. This was the first step of all the new features developed throughout the process. This step was helpful because it saved time by avoiding reworking the design later in the process as well as being able to simulate user actions and test the user experience.

At this stage, a small-scale (n=5) user test was conducted with a non-probabilistic convenience sample readily available to the team. The application was still at an early stage, which would make it resource-inefficient and time-consuming to conduct a user test on a representative demographic. However, this user test helped identify usability issues and improve the overall user experience. The user test feedback resulted in the first MVP's development. Some of the most significant feedback received and used for improvement are listed below.

- Some users disliked the color choice for the dark mode version. Based on this feedback, a new color scheme was picked for the application. The change can be seen in Figure 3.7.2
- The application lacked an efficient way of navigating between the topics and between the different task categories (examples, challenges, and coding exercises). Based on this feedback a top navigation bar was designed and developed to ease the navigation of the application. The change can be seen in Figure 3.7.3.

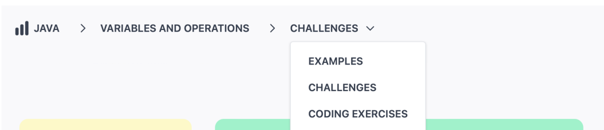


(a) Before

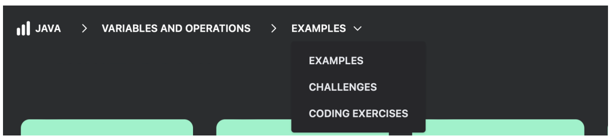


(b) After

Figure 3.7.2: Color scheme before and after user testing



(a) After (light mode)



(b) After (dark mode)

Figure 3.7.3: Top bar navigation added after user feedback

3.7.3 MVP

At this stage, the actual application was developed using the prototype made in Figma. New user tests were conducted on fifth-year students from NTNU campus Gløshaugen and were more in-depth than the previous tests. The team ultimately tested on six students, each of whom was given their own user ID in the system and instructed to do a series of tasks and then offer feedback. The feedback was organized in a Miro board, which can be seen in its entirety in Appendix G.1. The user tests provided the team with valuable insight and knowledge and were used to develop the application into the final product.

The most prominent feedback and resulting changes are listed below.

- Some users missed information about the different topics that would help them to learn the basics about the topic before starting on exercises. Based on this feedback an information box about each topic was designed and put on top of each topic page. See Figure 3.7.4.
- Some issues appeared related to the accordion view of the different topics. The links navigating to different parts of the system were "hidden", and it proved counterintuitive to navigate. The users did not understand that clicking on the name gave a different outcome than clicking on the arrow. Based on this feedback a new design for the accordion was made. See Figure 3.7.5.
- The initial onboarding page presented an overwhelming amount of information on a single page, requiring the user to scroll excessively and making it hard to absorb the information effectively. Consequently, the onboarding page got redesigned, with the information split into three pages for improved user experience. The change can be seen in Figure 3.7.6.

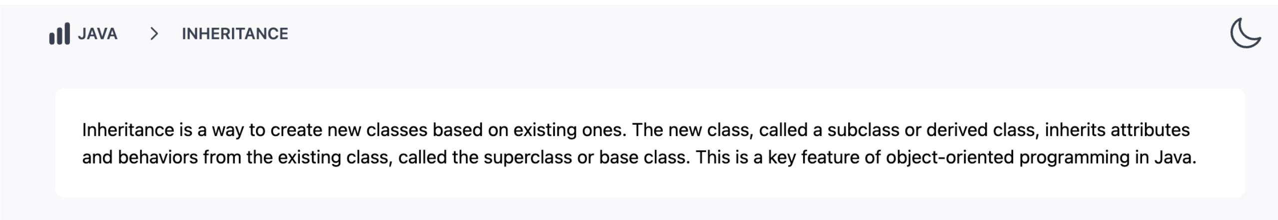
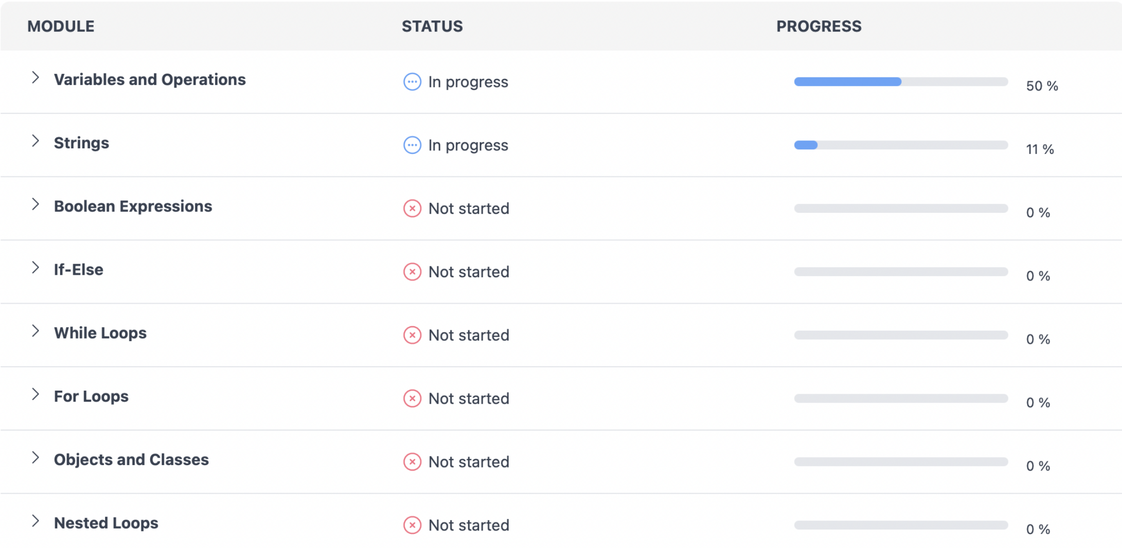
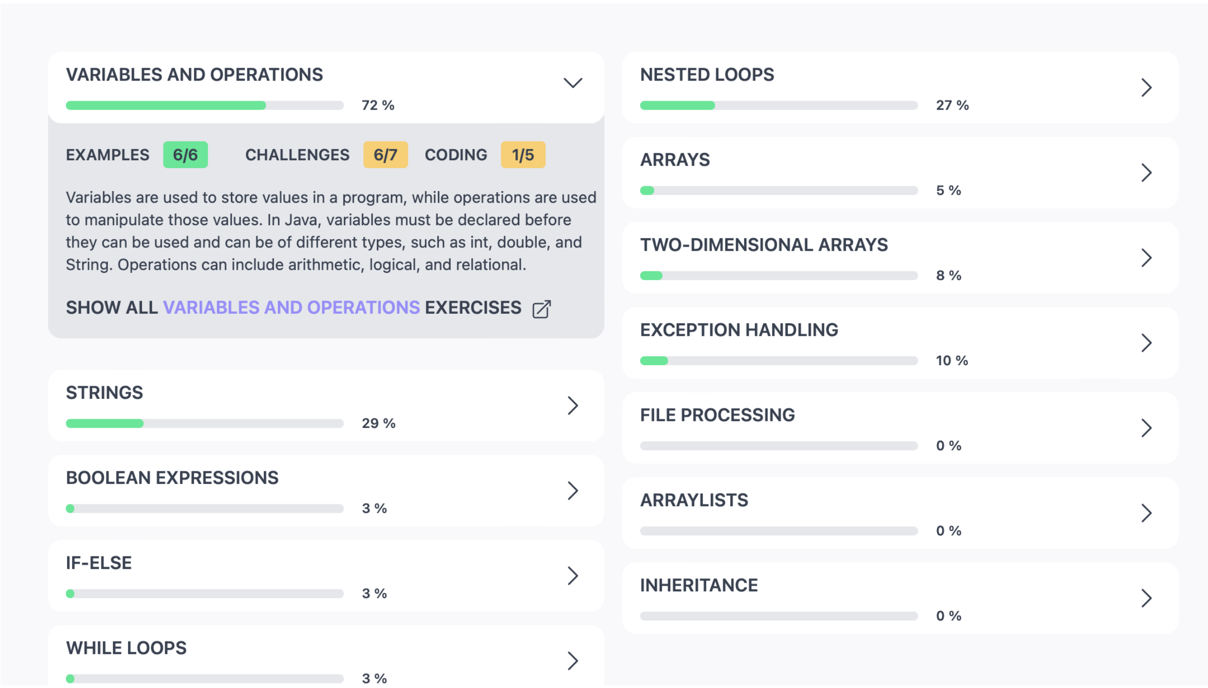


Figure 3.7.4: Text box added to each topic page

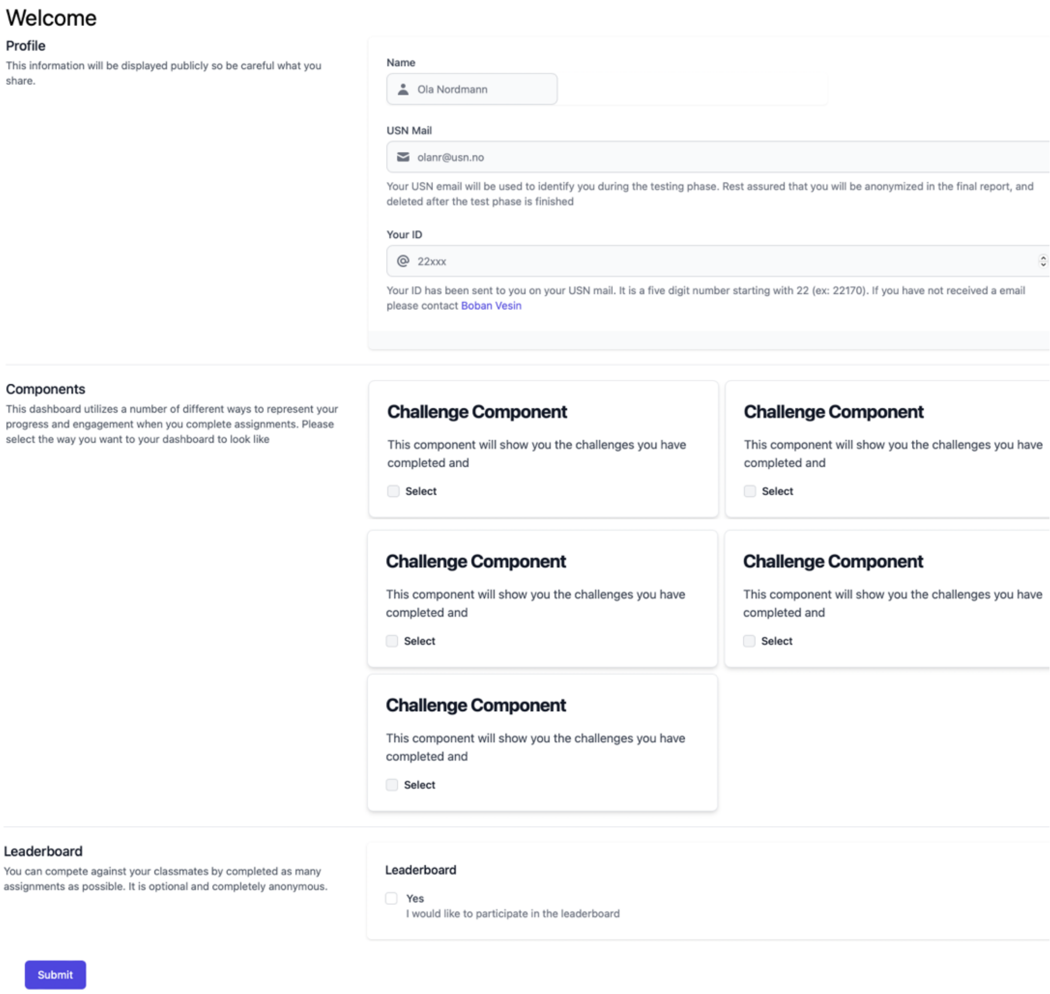


(a) Before

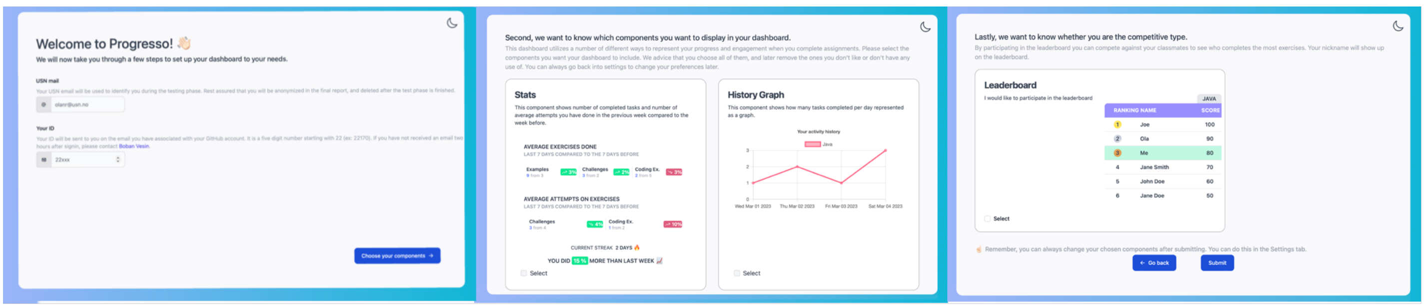


(b) After

Figure 3.7.5: Topic accordion overview before and after user testing



(a) Before



(b) After

Figure 3.7.6: Onboarding page before and after user feedback.

3.7.4 Final product

After the final user test and re-iteration, the application was finalized for the experiment test phase. After the test phase started, only minor changes and bug fixes were conducted on the design. In this section, the final product in its entirety is represented. Figure 3.7.11 shows a total overview of the application pages and how to interact with them. The final product includes a dark mode function, however, only light mode will be showcased in this section. The full-size user interface can be seen in both light- and dark-mode in Appendix H. The final color scheme used for the application can be seen in Figure 3.7.7. This color scheme was used for both light mode and dark mode, to keep the transition between the two as seamless as possible.

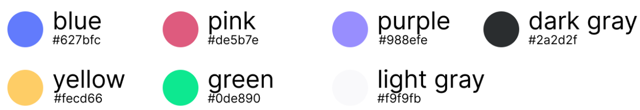


Figure 3.7.7: Color scheme

3.7.4.1 Onboarding

When the users first enter the Progresso application they have to undergo an on-boarding process. On the first page, which can be seen in Figure 3.7.8, the users have to enter their student email as well as their Progresso ID (which was provided via email beforehand). On the next page, as seen in Figure 3.7.9, the users have the option to choose their own components based on their wants and needs. The configuration can be changed later. The last page of the on-boarding process is where the users can decide whether they want to join the leaderboard or not. If they do, they will have to choose a name that will be shown on the leaderboard. This final page can be seen in Figure 3.7.10.

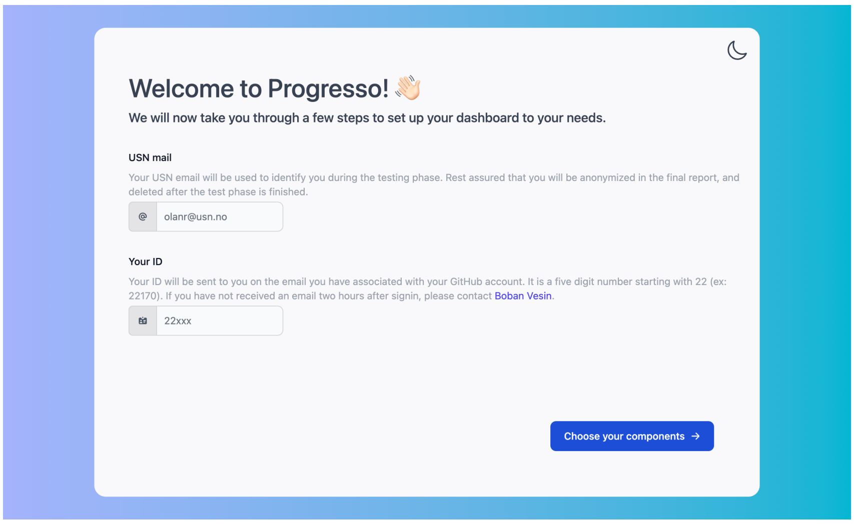


Figure 3.7.8: First page of the on-boarding process

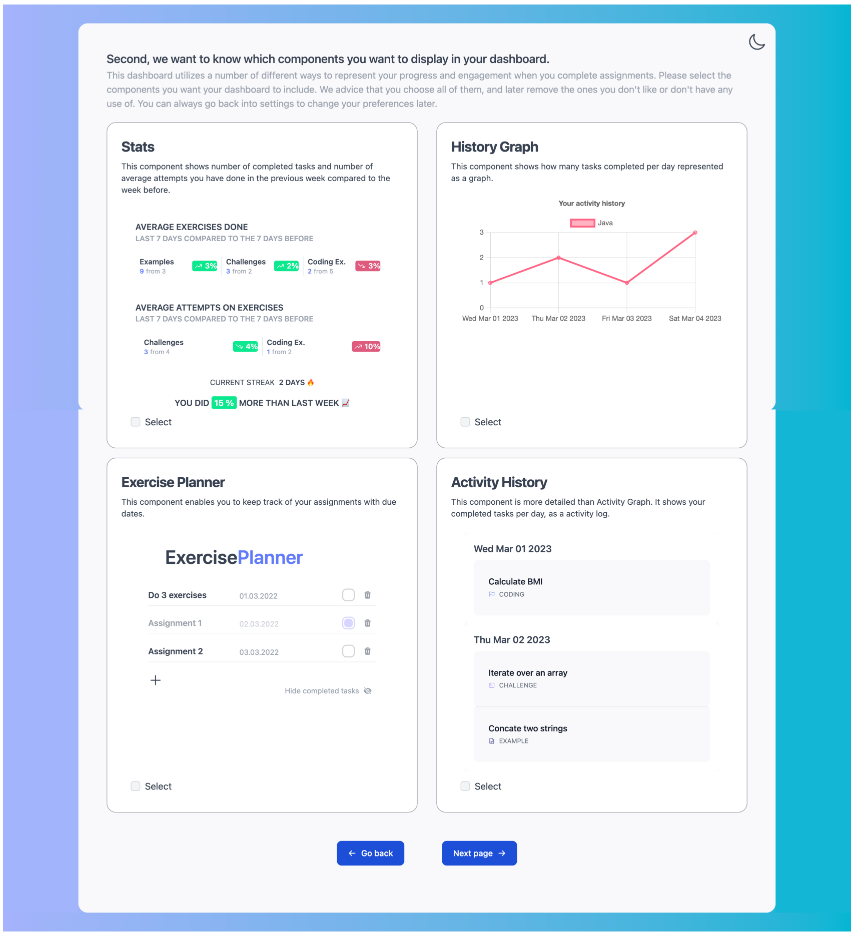


Figure 3.7.9: Second page of the on-boarding process

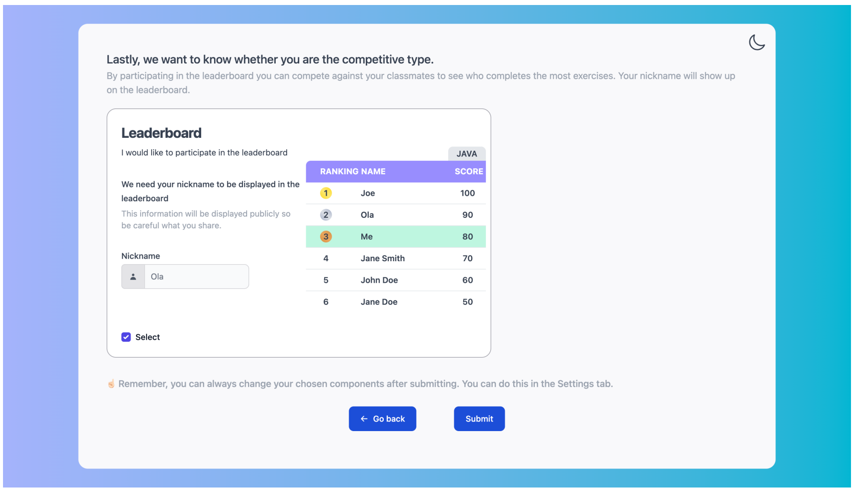


Figure 3.7.10: Third page of the on-boarding process

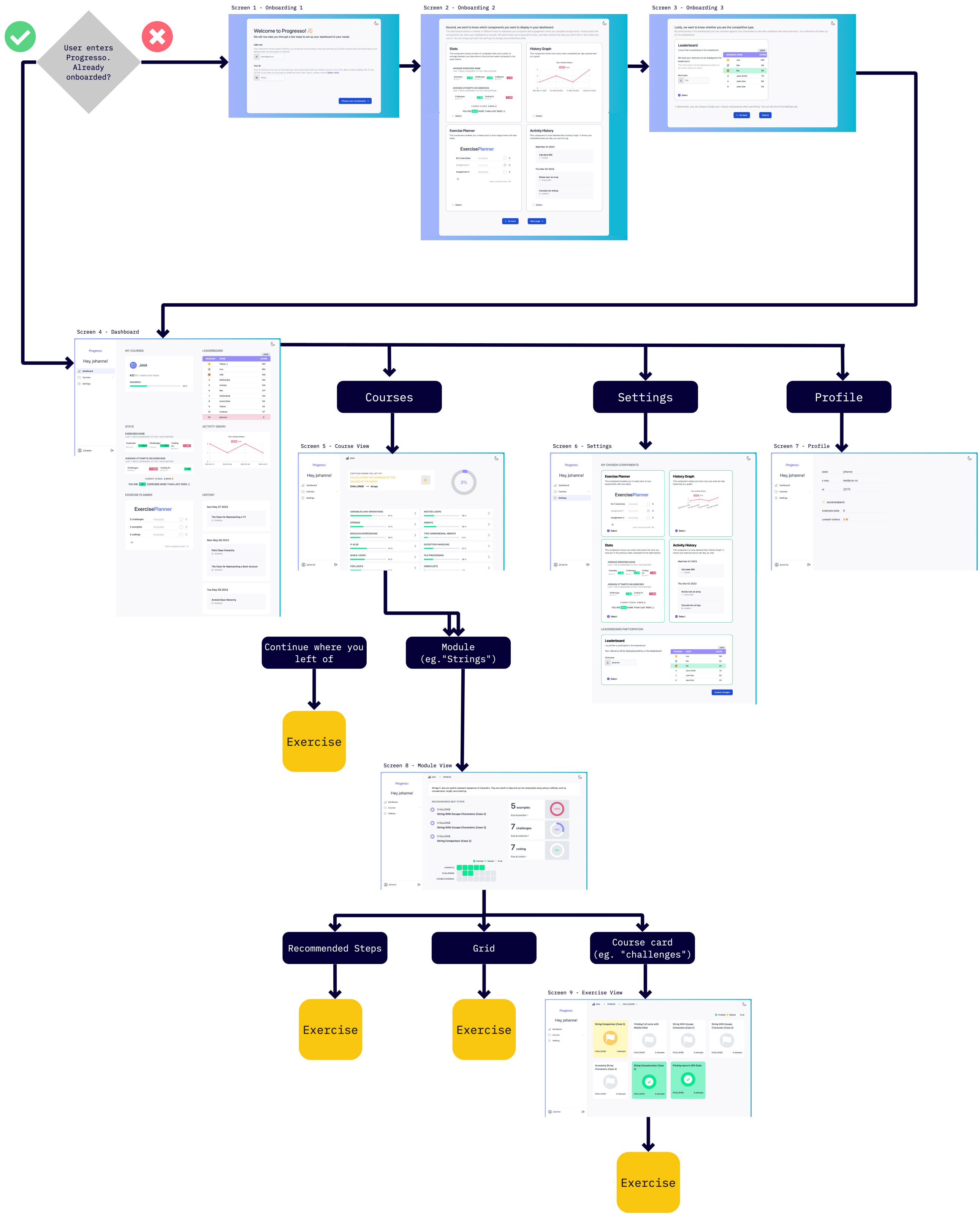
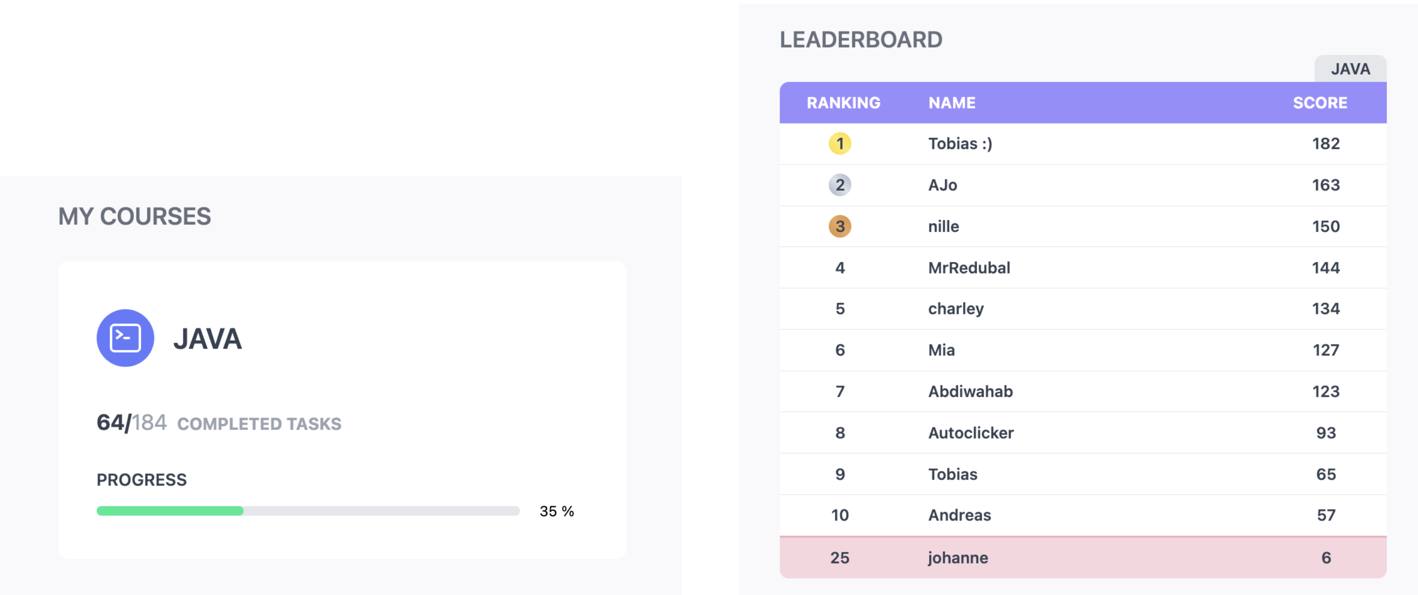


Figure 3.7.11: User flow

3.7.4.2 Dashboard

After on-boarding, the user is redirected to the customized dashboard. The dashboard consists of one permanent component, in addition to five optional components that the users choose themselves. The permanent component is a *course card* that shows the user their progress on their course. It also works as a link directly to the course page. The component can be seen in Figure 3.7.12 a. The five optional components are listed below. The complete dashboard, in both light and dark mode, can be seen in Appendix H.

Leaderboard
The first optional component is the leaderboard. The leaderboard showcases the top 10 users that have done the most exercises. Each completed exercise yields one point. Should the user be outside the top 10, the position is shown on the bottom line. The leaderboard can be seen in Figure 3.7.12 b.



(a) Course Card component (b) Leaderboard component

Figure 3.7.12: Dashboard components

Statistics
The statistics component showcases statistics of the user's performance. The first row shows how many exercises the user has done in the current week compared to the week before, in numbers as well as arrows pointing up or down. These are separated into examples, challenges, and coding exercises. The second row shows the user's average number of attempts in the current week compared to the week before, separated into challenges and coding exercises. On the bottom, the users can see their current streak (how many days they have been doing exercises in a row), as well as whether they have done more

exercises than the week before. The statistics component can be seen in Figure 3.7.13 a.

Activity Graph

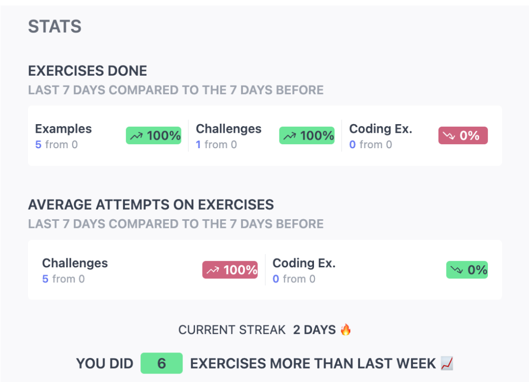
The activity graph is a line chart that visualizes the number of exercises that the user has done each day. The component can be seen in Figure 3.7.13 b.

Exercise Planner

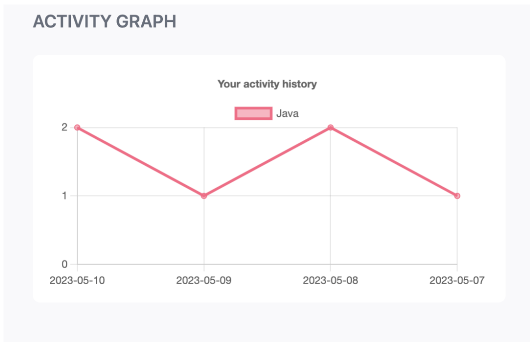
The exercise planner is a planning tool that allows the user to put in their to-do's and set a deadline. When the task is done, they can tick off the to-do. As seen in Figure 3.7.13 c, the user has the option to show/hide the tasks they have already done.

Activity History

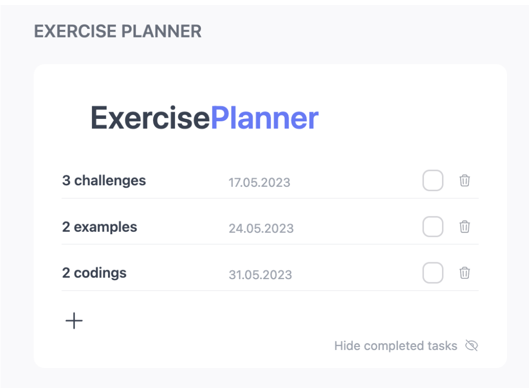
In the activity history component, the users get a list of every exercise they have done sorted after time completed, in a more detailed manner than in the activity graph. The component can be seen in Figure 3.7.13 d.



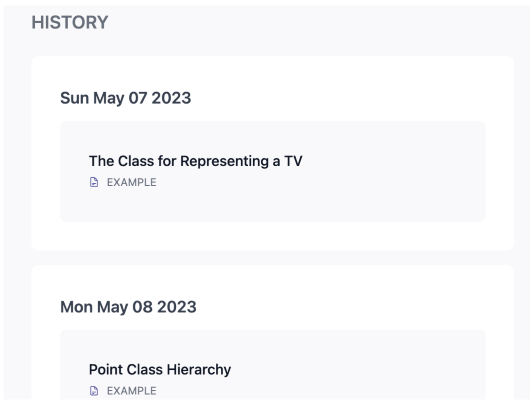
(a) Statistics component



(b) Activity Graph component



(c) Exercise Planner component



(d) Activity History component

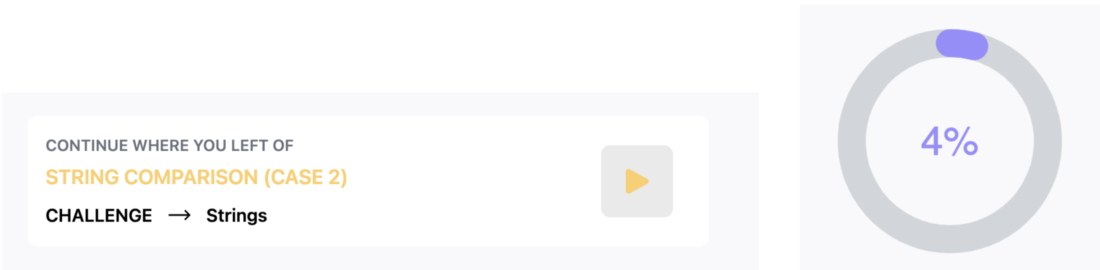
Figure 3.7.13: Dashboard components

3.7.4.3 Course Page

The course page consists of three components. The first component is a continue where you left of button. This will take the user straight to the last exercise he/she did not complete. The component can be seen in Figure 3.7.14 a.

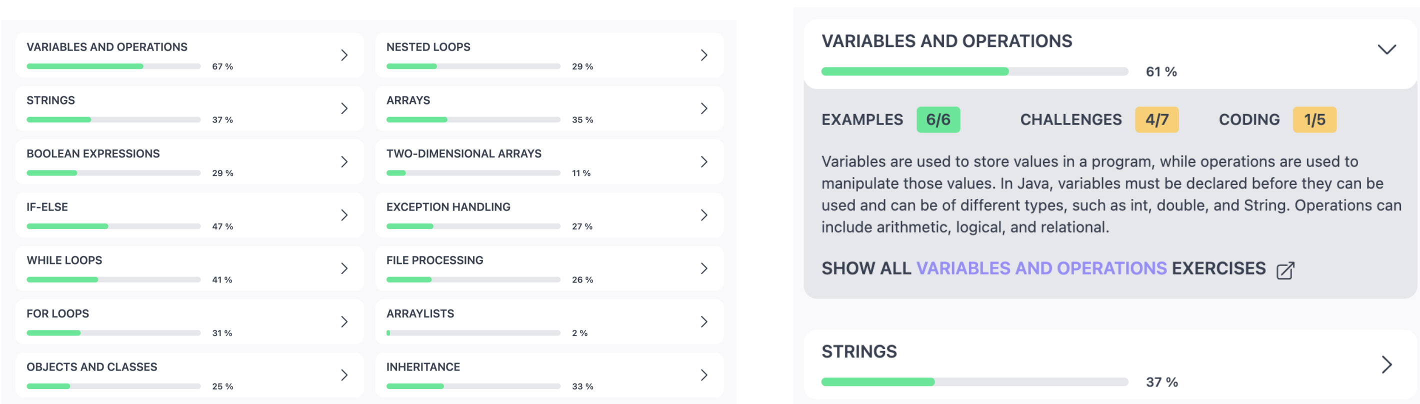
The second component is a progress donut chart that shows the user their overall progress in the course. The donut chart can be seen in Figure 3.7.14 b.

The third component is an accordion that showcases all the topics of the course. The accordion can be seen in figure 3.7.14 a. When the user clicks on a topic, the topic information is shown. This includes how many exercises the user has done in each of the exercise types (with links to the respective exercise pages), an informative text about the topic as well as a link to the topic page. The open accordion can be seen in figure 3.7.15 b.



(a) Continue where you left of **(b)** Progress Donut Chart

Figure 3.7.14: Course page components

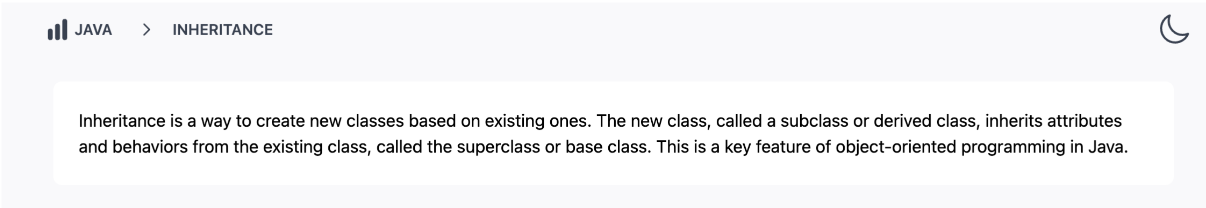


(a) Accordion component (b) Accordion component - open

Figure 3.7.15: Course page components

3.7.4.4 Topic Page

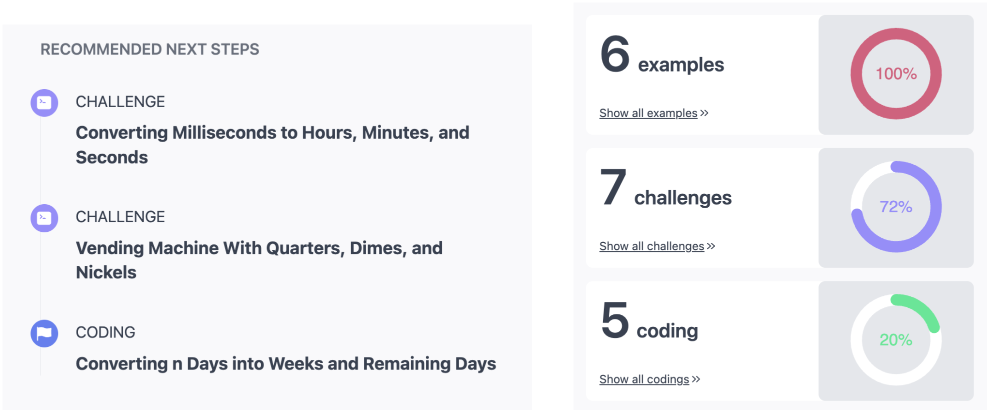
The topic page consists of four components. On top of the page, there is a text box with a short description of the topic. See Figure 3.7.16 a. Second, there is a Recommended Exercises component that showcases the user's next three recommended exercises within the topic. See Figure 3.7.17 a. The third element is an Exercise cards component that showcases each of the three types of exercises, how many of each the user has done, and how many are left. There's also a donut chart visualizing the progress within each exercise type. The component can be seen in Figure 3.7.17 b. The last component on this page is a grid map showcasing each exercise within the topic. The exercises that are done have a dark green color, the exercises that are started but not finished have a lighter green color, and the exercises not started have a grey color. By hovering the squares the exercise's name pops up, and by clicking the square the user is taken straight to the exercise. The grid component can be seen in Figure 3.7.16 b.



(a) Informative text box

(b) Exercise Grid - hovered

Figure 3.7.16: Topic page components



(a) Recommended exercises (b) Exercise cards

Figure 3.7.17: Topic page components

3.7.4.5 Exercise Page

The exercise page contains the exercises within the chosen topic. Each exercise has a card containing basic information - the name of the exercise, how many attempts the user has used, and which type of exercise it is (example, challenge or coding exercise). The exercise cards are color coded. Green means they are finished, yellow meana they are started, and grey means they are yet to be done. The exercise cards can be seen in Figure 3.7.18.

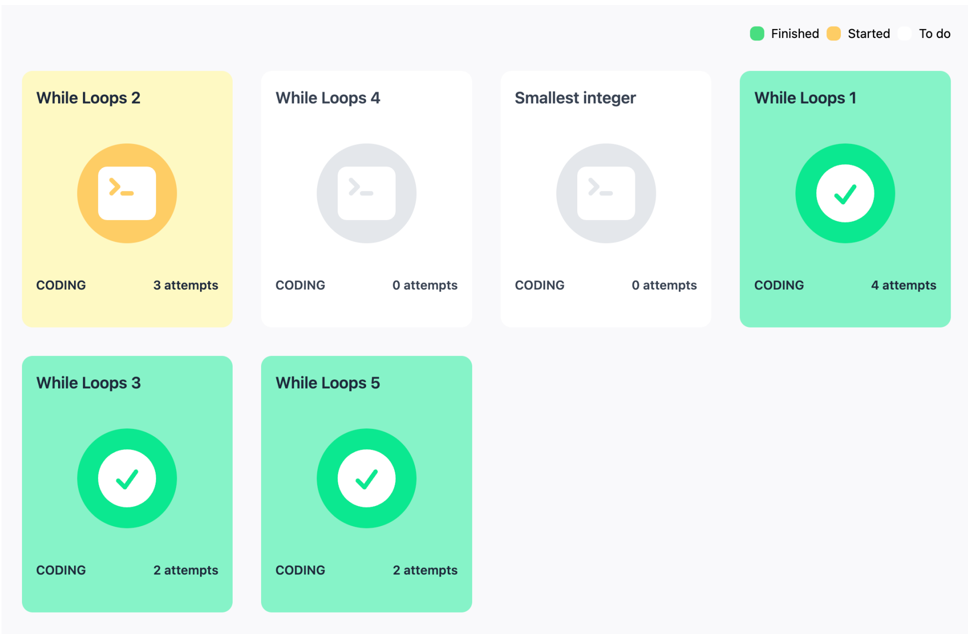


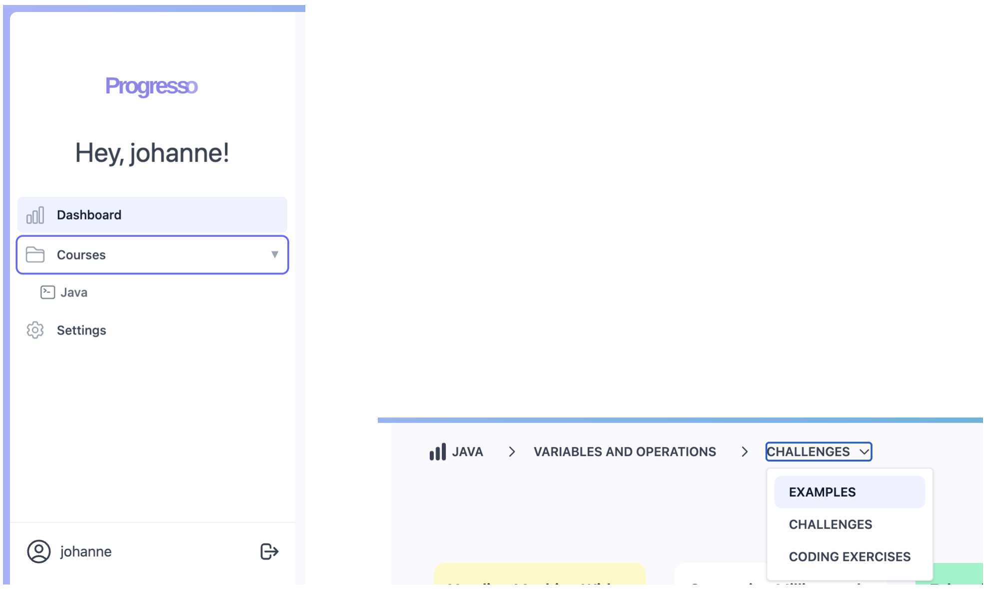
Figure 3.7.18: Exercise cards

3.7.4.6 Settings

On the settings page, the users can customize their dashboards, and change the components they originally chose during the onboarding process. The users can simply tick on or off the components they would like to include in the dashboard. The settings page can be seen in Figure H.8 in Appendix H.

3.7.4.7 Navigation

Navigating the application is done mainly via the sidebar. The sidebar contains links to all the main pages of the application. This can be seen in Figure 3.7.19 a. When the user enters the course page a top bar is visible on the top of the page. This can be used to navigate between the course page, the topic page, and the exercise page. See Figure 3.7.19 b. When the user is on the exercise page he can use the top bar to navigate between the different exercise types.



(a) Sidebar (b) Topbar

Figure 3.7.19: Application navigation

3.8 Rationale behind dashboard components

Table 3.8.1 shows the rationale behind the decision to create each component based on prior research and established theories.

Component	**Bakground**
Leaderboard	The leaderboard was included both as a way for the students to compare their performance to their peers and for fostering competition. As mentioned in Subsection 2.3.3, students perceived it as useful to be able to compare their positions with their peers [22]. Studies on leaderboards have provided conflicted findings, so the leaderboard was included to investigate these effects as well.
Statistics	The statistics component was included because previous studies have pointed out that students were positive regarding seeing statistics about their scores and performance, as mentioned in Subsection 2.3.3. Kupczynski et al. suggests that there may be a correlation between frequency of learning and performance [79]. The inclusion of the *streak* element is meant to serve as a motivation to keep up this frequency.
Activity Graph	The inclusion of the activity graph was based on feedback from previous studies where students have expressed that they would want to see their aggregated progression over time in a visual representation [8], similar to the statistics component. In the activity graph they get to see how many exercises they are doing each day compared to their past performance, and much like the inclusion of the *streak* element it is meant to motivate the student to keep up this frequency.
Exercise Planner	Self-regulation theory, as mentioned in Subsection 2.2.1, motivated the inclusion of the exercise planner. Goal-setting is a key part of this theory, and states that individuals who actively evaluate and adjust their goals are more likely to achieve success [27]. The exercise planner is meant to work as a tool for the students to assist them in this goal setting. It can also contribute to the students' sense of responsibility and accomplishment, which they can experience when ticking off a mark in the planner.
Activity History	Recognizing that students' needs and preferences vary greatly, as mentioned in Subsection 2.3.3, the activity history component was included for the learners who prefer non-visual representations and alternative ways to see their past performance. The activity history components represent similar data as the activity graph, but including both allows the students to choose in which format they would like to view this data.

Table 3.8.1: Dashboard components in Progresso and their rationale

3.9 Evaluation

The team created 13 functional requirements as a comprehensive overview of the entire Progresso ecosystem. However, not all requirements were fulfilled. FR 4, 7, 10, 11, and 13 were not implemented due to time constraints. The team focused on the FRs that were directly tied to the research questions. In FR4, the skill mastery (learning outcome) was calculated, but the data was not communicated to the student in the UI. FR7 was intended to report the mastery prediction and integrate it into the University's LMS using the LTI protocol so the tutor could identify students who needed follow-up. FR11 was partially completed. The teacher could have logged in to Supabase to organize the course through the table inspectors of the database. However, a standalone UI was not made. In the current version of Progresso, the tutor would have to login into the admin panel of Supabase and open the database tables to make changes. FR13 was meant to act as a feedback mechanism where students could give each other hints about an exercise. This FR necessitated a real-time database which would have increased the complexity of the application tremendously.

Functional Requirement 9 required that users should be served with some type of recommended content. Recommended exercises were offered as a black box feature of the learning content API. Adapt2 had zero documentation, so the team can only speculate on the extent and effect of recommended exercises.

CHAPTER

FOUR

METHODOLOGY

4.1 Progresso Participants

The participants of the experiment were students of the University of South-Eastern Norway. In total, 73 students participated in the study. 25 (34,2%) of the study's participants were taking a 1st year-level introductory Java programming course, while the remaining 48 (65,8%) were taking a 2nd year Java programming course.

4.2 Questionnaire Participants

Out of the 73 students that participated in the study, a total of 33 answered the post-questionnaire. 8 of the students were from the 1st year level, and 24 were from the 2nd year level. 34% of the respondents were female, and 66% were male. The age ranged from 20 to 50. The sample of participants can be seen in table 4.2.1.

Year of study	N	Mean age	Female	Male
1	8	25	4	4
2	24	28	7	17

Table 4.2.1: Sample of participants in questionnaire

4.3 Setting

The participants were informed about the study and the developed product during a lecture in their respective programming courses. After the presentation, the students were given three assignments due over a period of four weeks. After they had each registered as users in the Progresso app, the team sent each participant a mail with their unique IDs and an informative letter that can be seen in Appendix C. None of the participants had any prior experience with the application. The assignments consisted of several programming tasks that had to be done through the system with a given deadline. See Appendix D for the assignments. The tasks were completed on the participants' personal computers. This allowed the team to monitor the participants' decisions and progress away from a potentially stressful testing environment. The questionnaire was sent out to all participants after the four-week testing phase of the application. There were drawn three winners of 500 NOK gift cards each to motivate the participants to respond.

4.4 Research Strategy

To approach the research question, a research strategy is needed. Out of the strategies mentioned by Oates, two were chosen as the most suitable.

- **Design and Creation** - The design and creation strategy was chosen because the nature of this study requires the development of an IT system. To be considered as research, such systems must include academic qualities such as analysis, explanation, and critical evaluation [20]. This system - the Progresso app - was used as the basis for the research, and the contribution to knowledge is based on how the system is used and not the system itself.
- **Experiment** - The experiment strategy is used to investigate cause and effect relationships, and to answer the research questions. The experiment strategy was chosen to test the Progresso application on the two classes. The strategy included collecting data, behavioral counts as well as self-reported responses through the questionnaire.

4.5 Data Generation

Data generation is the field where data or evidence is produced [20]. This data can be either quantitative or qualitative, and both types were produced during this research. Out of Oates' data generation methods, three were chosen. Using more than one data generation method allows the researchers to look at the data from different angles and enhance the validity of the findings. This approach is called method triangulation.

The combined data generation of Progresso, adapt2, and the questionnaire contains more than necessary to answer the research questions. In Section 4.5.1, the data collected by Progresso and Adapt2 is presented. In Section 4.5.2, the data collected by the questionnaire is presented. For the full reference of data collected, visit Appendix E and E.1 for the questionnaire, Appendix I for data collected on challenges type and J for data collected on coding exercise type.

4.5.1 Variables

Table 4.5.1 shows the combined variables collected by both Adapt2 and Progresso that were used to answer the research questions. The variables are unique to two datasets. One dataset for the student's challenges and one for coding exercises. Note that since the students had the option to opt in and out of components and the leaderboard, the final configuration was considered.

Variable	Description	Source	Type	Scale
attempts	Average attempts per student	Adapt2	Ratio	(0, 15]
duration	Average duration spent on exercise (seconds)	Progresso	Interval	Challenge (0,900] Coding (0,3600]
exercisehistory	Whether the student chose the exercise history component	Progresso	Categorical	True/False
exercise_first	What exercise type the student picked first	Adapt2	Categorical	example, challenge, coding
exercise_id	Exercise id	Adapt2	Nominal	
exercise_type	Type of exercise	Adapt2	Categorical	example, challenge, coding
historygraph	Whether the student chose the historygraph component	Progresso	Categorical	True/False
leaderboard	Whether the student participated in the leaderboard	Progresso	Categorical	True/False
skill_mastery	Prediction of whether the student has mastered the skill	Adapt2	Interval	[0,1]
stats	Whether the student chose the stats component	Progresso	Categorical	True/False
success_rate	Ratio between correct attempt and attempts	Adapt2	Interval	[0,1]
todo	Whether the student chose the exercise planner component	Progresso	Categorical	True/False

Table 4.5.1: Data collected

4.5.2 Questionnaire

To evaluate the participants' opinions regarding the system's usability and design, a questionnaire was chosen. It was a self-administered questionnaire made using Microsoft Forms and was sent out immediately after the experiment phase concluded. The first part was the standard System Usability Scale questions. The remaining parts were set of questions determined by the team to most accurately collect data regarding system usability, motivation, engagement, and design, as well as attitude towards each choosable component. The questionnaire was mostly closed - meaning that the respondents had to choose from a range of pre-defined answers [20] - except for the last question, which asked the respondent for any extra comment, allowing them to give feedback that was not covered by the questionnaire. A 5-point Likert scale was chosen for most questions to avoid confusing the participants and avoid the questionnaire design looking cluttered [20]. In addition to asking the standardized SUS questions, the team made five more categories of questions to investigate further and dig into the students' opinions on the customizability, components, and the prefered order of selected exercises. The complete list of themes covered by the questionnaire is as follows;

- **System Usability Scale Schema** As mentioned in Section 2.3.3, SUS was chosen due to it being a simple usability scale with a high level of validity [50]. SUS is categorized using a 5-point Likert scale, ranging from "Strongly disagree" to "Strongly agree".

- **Customizability** Questions concerning the customizability of the application. The questions were answered using a 5-point Likert scale.

- **Dashboard components** In this section each component of the system was singled out, and respondents had to rate each on a scale of 1 to 5, as well as answer questions using a 5-point Likert scale on each of them. The participants were only asked questions regarding the components they had actually chosen to use in their dashboards to avoid too much unnecessary clutter.

- **Order** The participants were asked to rank the order in how they preferred selecting their exercises, as well as to rank the type of exercises (examples, challenges, coding exercises) they selected the most down to the exercises they selected the least.

- **Overall design** Questions were asked about the overall design and color choices of the application and the on-boarding process. These were answered with a 5-point Likert scale.

- **General feedback** Finally, the participants got the opportunity to write additional comments and feedback at the end of the questionnaire in hopes of covering any remaining thoughts they might have. Answering this last question was optional.

The questions combined provided a thorough insight into how the users perceived the user interface of the system. The complete questionnaire can be seen in Appendix E.

4.6 Data Analysis

Data analysis is the field in which relations, patterns, and themes are identified [20]. The data is either analyzed quantitatively - based on numbers - or qualitatively - non-numeric. Given the chosen data collection methods, both quantitative and qualitative analysis were needed. In this section, the data analysis methods used are presented.

4.6.1 Quantitative analysis

After the test-phase was completed, the team had data collected on the adapt2 API and their own database. The two data-sets were merged into one. The variables are illustrated in Table 4.5.1. The data went through filtering based on the exclusion criteria below. The quantitative data analysis consisted of independent two-sample t-tests and a linear regression. The confidence level of $p<0.05$ was considered for all tests and models.

1. The exercise must have a timestamp to sequence them correctly.
2. The student must have opened and attempted the exercise at least once.
3. The exercise type must be of challenge or coding exercise.
4. Challenges not answered correctly on the first try is considered incorrect response.
5. The skill per student must have a combined sequence length of at least 15 derived by exercise attempts.
6. The exercise must have an attempt count of a maximum of 15.
7. Duration must be below one hour for coding and 15 minutes for challenges.

The inclusion criteria were necessary to satisfy the requirements of pyBKT [11]. Sequences of less than 15 per skill per student and a total sample size of less than 50 would result in unsatisfactory model accuracy. The sample size of the first-year class was 25, well below the threshold of 50. However, the model performs with a Mean Absolute Percentage Error (MAPE) <25% for all parameters except priors in Figure 2.3.2. As such, one can determine individual guess, slip, and learn rates for first-year students but not individual priors. Outliers were identified by plotting duration and attempts in a scatterplot. A reasonable cut-off point was determined to be any attempts above 15, where it is assumed that random guessing has started to occur. Above 15 minutes for challenges and 1 hour for coding was found to be a reasonable cut-off for the duration metric.

Research Question 1
To answer RQ1, *What predictors influence the learning outcome in a learning environment?* and its following sub-questions, one must determine the student's skill mastery. For this analysis, the team utilized two measures. The first one was the skill mastery

prediction determined by pyBKT. The team had enough data per pyBKT requirements to utilize two models of pyBKT. KT-IDEM was used to individualize the guess and slip parameters per exercise type. Ideally, the KT-IDEM should have taken the exercise id to determine each exercise's slip and guess probabilities. Preliminary testing of this model setup showed overfitting in the first-year students. However, estimating the slip and guess per exercise type per skill yielded a much greater number of sequences. Learning rates were generated per student using the Item Learning Effect model. The average skill mastery was determined by summing the student's skill prediction per skill and dividing it by the number of skills for which the student had a valid mastery prediction. The model fitting was validated by cross-validation with Area under the Curve (AUC) and Root Mean Square Error (RMSE) as metrics.

The second metric was the average success rate per student. The average success rate was obtained by dividing all success rates from attempted exercise divided by the number of attempted exercises. Note how the success rate differs from skill mastery. The success rate represents the ratio of correct to incorrect attempts, while skill mastery predicts the student's potential to transit from the not-known to known state of the FSM in 2.3.1.

Sample	**Categorical Variable**	**Dataset**
attempts	class	Challenge
skill_mastery	class	Challenge
attempts	leaderboard	Challenge
skill_mastery	leaderboard	Challenge
attempts	exercise_first	Challenge
skill_mastery	exercise_first	Challenge
successRate	class	Coding
skill_mastery	class	Coding
successRate	leaderboard	Coding
skill_mastery	leaderboard	Coding
successRate	exercise_first	Coding
skill_mastery	exercise_first	Coding

Table 4.6.1: Two sampled independent t-tests

The data analysis consisted of conducting independent two-sampled t-tests. Two environments were considered, exercise of type challenge and coding. These were considered independent of each other because they infer knowledge from the student differently. For each t-test, each sample was checked for normality by running a Shapiro-Wilks test in version 27 of SPSS. Additionally, if the Levene's test yields a p-value greater than 0.05, it is generally acceptable to assume equal variances for subsequent statistical tests that assume equal variances [80, p 729]. Different categorical variables were run between continuous variables from Table 4.5.1. Only continuous variables from a normal distribution were considered. Table 4.6.1 depicts the different test scenarios. The attempt variable is the average attempts across all the student's exercises either correctly or incorrectly answered. The duration is the average duration for correctly answered exercises by subtracting the *completedAt* field with the *visitedAt* field from the Progresso data generation.

After the t-tests were completed, a backward linear regression analysis was used to discover which predictor explains most of the variance in the obtained skill mastery and success rate. The backward method can potentially minimize suppressor effects because it considers the effect of each predictor by starting with a full model and eliminating variables one by one, the analysis keeps those variables that maintain significance in the presence of others, helping to mitigate suppressor effects [80, p 532]. The regression was done using version 27 of statistics program SPSS.

Research Question 2

To answer the second research question, *How can we design learner analytics dashboards in order to boost student engagement?*, a combination of qualitative and quantitative methods was employed. The quantitative aspect involved analyzing the data collected from the questionnaire responses. In this analysis, tables and charts were used to present the data in a concise and visually informative manner [20]. For visualizing the questionnaire responses, bar charts were chosen as they allowed the researchers to explore the data and identify patterns while also facilitating easy comprehension for the reader. To create these bar charts, mean values were calculated for each question, considering that the number of respondents for each question varied.

4.6.2 Qualitative analysis

Research Question 2

The qualitative data used to address RQ2 was obtained through the final open-ended question of the questionnaire, which sought general feedback. It is worth noting that this question was optional, resulting in limited data availability. To analyze this qualitative data, an inductive approach was followed, wherein the categories were derived from the observed data itself rather than being predetermined [20]. Firstly, the researchers thoroughly reviewed the feedback to gain an overall understanding of the responses. Next, key themes that appeared relevant to the research question were identified, while feedback unrelated to the research purpose was set aside. This method is called a thematical analysis. Next, the relevant feedback was organized according to the identified themes. For presentation purposes, the data was cleaned without altering the actual quotes of the respondents. Based on the limited nature of the data no further data analysis processes were needed.

CHAPTER
FIVE

RESULTS

The results chapter introduces the data analysis based on the methodology decided upon in the former chapter. The first section, 5.1, shows the results of the data analysis collected from the data generated by the artifact, Progresso. The second section, 5.2, shows the respondents' answers from the post-questionnaire after the experiment ended.

5.1 Progresso

Derived from the quantitative data analysis prompted by research question 1, the team collected a dataset consisting of 4630 entries, of which 1187, 1516, and 1227 examples, challenges, and coding exercises, respectively. Only challenges and coding exercises were considered since examples did not test knowledge. One dataset for challenges and one dataset for coding exercises were collected, where each entry was the information gathered per user. The variables are explained in Table 2.3.1 of the Methodology chapter.

5.1.1 Component usage data

The Upset diagram in Figure 5.1.1 shows the intersections between the selected component sets [81]. Of those who chose the exercise planner component, two students added todos at least once but had no more than two todos simultaneously. The stats component was the most selected, followed by the exercise planner.

Nine students did not choose any data visualization components. Five of them made a single configuration, asserting that they maintained the "zero" dashboard setup they selected during the onboarding step throughout the test phase. In contrast, two students changed their configurations three times, while the remaining two made ten changes.

The categorical variables concerning which dashboard components the students selected are not included in the following analysis because no significant results were found between the means of any of the dependent variables.

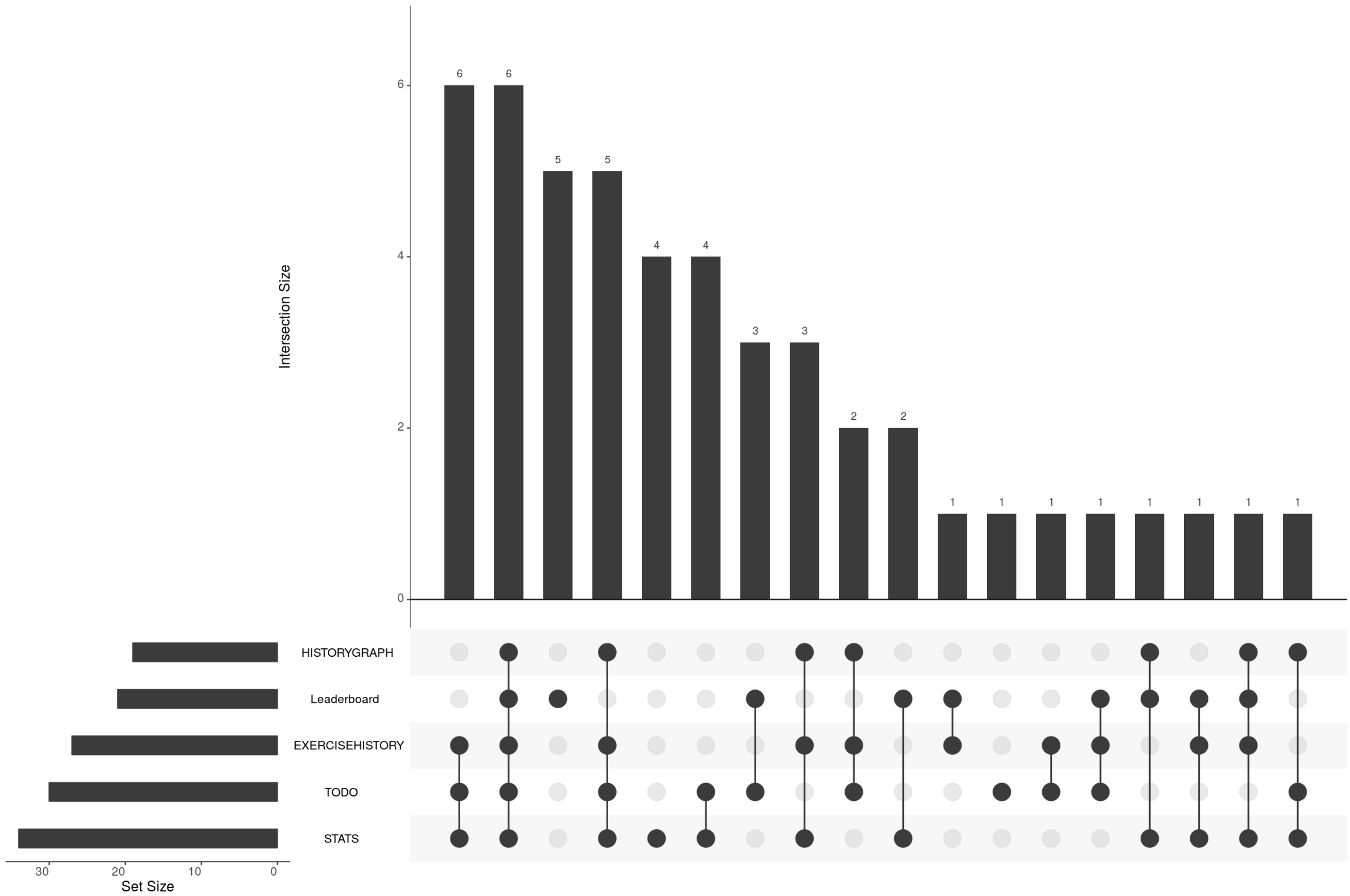


Figure 5.1.1: Upset diagram of the intersections between the selected component sets. The set size of those who selected zero components had a cardinality of 9.

5.1.2 Latent Knowledge Estimation

PyBKT was used to estimate the latent knowledge of the first- and second-year students. As per Section 4.5, the knowledge sequences were mined from adapt2 API, and the timestamps from the Progresso database were merged to achieve the correct sequences for each student. To determine the model fit that obtained the highest accuracy, 85 different model combinations were tested. The different models obtained different accuracy levels determined by Area under the Curve (AUC) and Root Mean Square Error (RMSE). The model that achieved the highest evaluation metric of AUC and lowest RMSE was chosen. Table 5.1.1 shows the AUC and RMSE for the pyBKT model used for determining the skill mastery per student. The setup was identical to the cross-validation reported by pyBKT using the standard model with five folds [82].

Skill	AUC	RMSE
Variables and Operations	0.680	0.434
Strings	0.678	0.424
Arrays	0.601	0.469
If-Else	0.679	0.430
While Loops	0.587	0.447
ArrayLists	0.380	0.447
Inheritance	0.488	0.442
Objects and Classes	0.622	0.404
Two-Dimensional Arrays	0.597	0.435
Nested Loops	0.317	0.430
For Loops	0.610	0.487
Boolean Expressions	0.638	0.456
Exception handling	NaN	0.348
File processing	NaN	0.456
Average	**0.671**	**0.398**

Table 5.1.1: Cross-validation for standard pyBKT model using five folds

5.1.3 Independent t-test

Table 4.6.1 from the Methodology chapter shows the cases for the independent t-test conducted. The t-test assumes that the data is approximately normally distributed and that variances are homogeneous. Table 5.1.2 assesses the assumptions of the t-test by providing information on the normality of the data. Levene's test is seen in each t-test table below.

Variable	Statistic	df	Sig.
avg_duration	.920	66	<.001
avg_success_rate	.958	66	.025
skill_mastery	.985	66	.619
avg_attempts	.973	66	.159

Table 5.1.2: Shapiro-Wilk Tests of Normality for challenge dataset

5.1.3.1 Challenge dataset

Class

Group statistics for the challenge sample with the class as the grouping is seen in Table 5.1.3. Table 5.1.4 shows the two sampled independent t-tests. Levene's statistic shows whether an equal variance is assumed or not.

Variable	Class	N	Mean	Std. Deviation
avg_success_rate	First Year	20	.48575	.176485
	Second Year	46	.53291	.175133
avg_attempts	First Year	20	3.1820	.95995
	Second Year	46	2.9337	1.04210
skill_mastery	First Year	20	.47575	.216706
	Second Year	46	.47148	.199122

Table 5.1.3: Group Statistics for challenge dataset with the class as group

Variable	Levene's Test	F	Sig.	t	df	2-tailed p
avg_attempts	E.V assumed	.473	.494	.910	64	.366
skill_mastery	E.V assumed	.309	.580	.078	64	.938

Table 5.1.4: Independent Samples t-test for challenge dataset between classes

Leaderboard

The other sample case is the leaderboard participation as the group and the other continuous variables. The descriptive statistics are seen in Table 5.1.5 and the t-test in Table 5.1.6.

Variable	Leaderboard	N	Mean	Std. Deviation
avg_attempts	TRUE	24	2.9933	1.02197
	FALSE	42	3.0179	1.02655
avg_success_rate	TRUE	24	.54296	.175676
	FALSE	42	.50471	.176058
skill_mastery	TRUE	24	.53871	.196252
	FALSE	42	.43510	.199202

Table 5.1.5: Group Statistics for the samples in the challenge dataset with leaderboard participation as grouping

Variable	Levene's Test	F	Sig.	t	df	2-tailed p
avg_attempts	E.V assumed	.002	.964	-.094	64	.926
skill_mastery	E.V assumed	.042	.838	2.044	64	.045

Table 5.1.6: Independent samples t-test for challenge dataset between leaderboard participation

Exercise First (Initial step of Learning Path)
The last grouping is the type of exercise completed first. The group statistics can be viewed in Table 5.1.7 Here an ANOVA analysis would be the practical analysis because there are three different categories when it comes to which exercise type the student picked first. However, no student decided to start their first observation with the type *challenge*. As such, the fallback is a two-sampled independent t-test, shown in Table 5.1.8.

Variable	Exercise First	N	Mean	Std. Deviation
avg_attempts	Example	53	3.0664	1.02821
	Coding	13	2.5390	.83107
skill_mastery	Example	53	.46098	.198573
	Coding	13	.55350	.178157

Table 5.1.7: Group Statistics for samples found in the challenge dataset with exercise first as grouping

Variable	Levene's Test	F	Sig.	t	df	2-tailed p
avg_attempts	E.V assumed	1.866	.177	1.527	64	.132
skill_mastery	E.V assumed	.045	.838	-1371	64	.175

Table 5.1.8: Independent samples t-test for challenge dataset between exercise first

No statistically significant differences between the means are found in the challenge dataset when grouping by class belonging with a confidence level of $p(< 0.05)$. One statistically significant difference is found between state prediction and leaderboard participation with a p-value of 0.045 found in Table 5.1.6. The mean skill mastery of the students who participated in the leaderboard was 0.54 while the mean of non-participating was 0.44. A medium effect size of *cohens d* 0.502. However, the difference is non-significant between the means when comparing the exercise type of the first observation of the students.

5.1.3.2 Coding dataset

One can fulfill the same test environment as in the challenge dataset by looking at the coding dataset. Table 5.1.9 show which variables are assumed to be normally distributed.

Variable	Statistic	df	Sig. (p<0.05)
avg_duration	.747	64	<.001
avg_attempts	.940	64	.004
avg_success_rate	.985	64	.609
skill_mastery	.980	64	.395

Table 5.1.9: Shapiro-Wilk Tests of normality for coding dataset

Class
In Table 5.1.11, it is observed that the mean between success rate and state prediction is statistically significant. The mean for success rate and state prediction is higher for the second-year students as per Table 5.1.10. The effect size using Cohen's d was 0.988 for success rate and -0.302 for skill mastery.

Variable	Class	N	Mean	Std. Deviation
avg_success_rate	First Year	22	.41891	.232080
	Second Year	44	.62111	.189780
skill_mastery	First Year	22	.34495	.222378
	Second Year	44	.57109	.233220

Table 5.1.10: Group Statistics for class group in coding dataset

Variable	Levene's Test	F	Sig.	t	df	2-tailed p
avg_success_rate	E.V assumed	1.369	.246	-3.784	64	<.001
skill_mastery	E.V assumed	.098	.755	-3.770	64	<.001

Table 5.1.11: Independent Samples t-test for coding dataset between class

Leaderboard
Table 5.1.12 shows the group statistics with leaderboard grouping. No significant results are found for either of the samples, see Table 5.1.13.

Variable	Leaderboard	N	Mean	Std. Deviation
avg_success_rate	TRUE	23	.55643	.233899
	FALSE	43	.55226	.222304
skill_mastery	TRUE	23	.49043	.257430
	FALSE	43	.49853	.252059

Table 5.1.12: Group Statistics for coding with leaderboard grouping

Variable	Levene's Test	F	Sig.	t	df	2-tailed p
avg_success_rate	E.V assumed	.017	.898	.071	64	.943
skill_mastery	E.V assumed	.078	.781	-.123	64	.902

Table 5.1.13: Independent Samples t-test for coding dataset with leaderboard grouping

Exercise first (Initial step of Learning Path)
Table 5.1.14 showcases the group statistics with exercise first grouping. No other significant differences between the means of skill prediction and skill mastery is found when grouping by type of exercise done first, as seen in Table 5.1.15.

Variable	Exercise First	N	Mean	Std. Deviation
avg_success_rate	Example	53	.56334	.029618
	Coding	13	.44330	.081515
skill_mastery	Example	53	.50157	.244045
	Coding	13	.60480	.171895

Table 5.1.14: Group Statistics for coding with exercise first grouping

Variable	Levene's Test	F	Sig.	t	df	2-tailed p
avg_success_rate	E.V assumed	.282	.598	1.566	64	.123
skill_mastery	E.V assumed	1.945	.168	-1.275	64	.207

Table 5.1.15: Independent samples t-test for coding dataset between exercise first

5.1.4 Linear Regression

A linear regression model was performed to determine the predictor that explains the most variance in skill achievement. Separate linear regression was done on the challenges - and the coding dataset.

5.1.4.1 Challenges dataset

Table 5.1.16 shows the descriptives of the model, which tells us the mean and variability (standard deviation) of each variable in the model.

	Mean	Std.Dev.	N
skill_mastery	0.47015	0.197990	66
avg_attempts	3.0089	1.01706	66
avg_duration	122.5012	75.32872	66
avg_success_rate	0.52326	0.173173	66

Table 5.1.16: Descriptives of linear regression - challenges

The correlation matrix in Table 5.1.17 shows the Pearson correlation coefficient between every pair of variables, in addition to the one-tailed significance of each correlation. We see that every correlation is significant except for the correlation between the skill_mastery and the avg_duration (p = 0.098). We see that skill_mastery and avg_attempts have a strong correlation (r = -0.679, p = 0.00). skill_mastery and avg_success_rate have a strong correlation as well (r = 0.790, p = 0.00).

		Variables			
		skill_mastery	attempts	duration	success_rate
Pearson Corr.	skill_mastery	1.000	-0.679	0.161	0.790
	attempts	-0.679	1.000	-0.253	-0.909
	duration	0.161	-0.253	1.000	0.247
	success_rate	0.790	-0.909	0.247	1.000
Sig. (1-tailed)	skill_mastery	.	$< .001$	0.098	$< .001$
	attempts	0.000	.	0.020	0.000
	duration	0.098	0.020	.	0.023
	success_rate	0.000	0.000	0.023	.

* all predictors are average. avg_ removed for space reasons

Table 5.1.17: Pearson Correlation of challenges dataset

Table 5.1.18 describes the overall fit of the model. In this table, three models are presented due to the utilization of the backward method. With the backward model, all predictors are initially included, and then start removing the least significant variable. The model is then re-evaluated using the remaining predictors, and the contribution of these predictors is re-evaluated accordingly.

The R column presents the multiple correlation coefficient, indicating the correlation between the predictors and the average state prediction. It demonstrates the relationship between the mentioned predictors and the outcome variable. The column R^2 accounts for how much of the variability in the outcome can be explained by the predictors. When only including avg_success_rate this predictor accounts for 62.5% of the variation in skill state prediction. When including all the variables the model accounts for 63.5% of the variation.

The adjusted R^2 column is a modified version of R^2 and adds precision and reliability. It takes into account the influence of additional independent variables, which can potentially affect the accuracy of R-squared measurements. Ideally, we would like the adjusted R^2 to be similar to the R^2. We see that the difference in the table is very small (0.625-0.619 = 0.006), indicating that the cross-validity of the model is very good.

Model	R	R^2	Adjusted R^2	Std. Error of the Estimate	Durbin-Watson
1	.797[a]	.635	.617	.122558	-
2	.796[b]	.634	.622	.121712	-
3	.790[c]	.625	.619	.122230	1.689

a Predictors: (Constant), avg_success_rate, avg_duration, avg_attempts
b Predictors: (Constant), avg_success_rate, avg_attempts
c Predictors: (Constant), avg_success_rate
d Dependent Variable: skill_mastery

Table 5.1.18: Model Summary[d] of challenge dataset

Table 5.1.19 displays the model parameters for all three steps in the model. The table reveals the evaluation of each variable through backward elimination. Among all variables, only avg_success_rate received a significance lower than 0.05, which is why this was the only remaining value in the model. The first column of the table provides estimates for the B-values, representing the contribution of each predictor to the model. The magnitude of the B reflects the extent to which each predictor influences the outcome. In this case, a B value of 0.904 signifies that a 1-unit increase in avg_success_rate is associated with an expected increase of 0.904 units in average_skill_mastery.

		Unstd. Coefficients		Std. Coefficients		
Model		*B*	Std. Error	*Beta*	*t*	*Sig.*
1	(Constant)	-.249	.215		-1.156	.252
	avg_attempts	.043	.036	.222	1.205	.233
	avg_duration	-7.619E-5	.000	-.029	-.365	.716
	avg_success_rate	1.143	.211	1.000	5.429	$< .001$
2	(Constant)	-.259	.212		-1.224	.226
	avg_attempts	.044	.036	.227	1.243	.218
	avg_success_rate	1.140	.209	.997	5.456	$< .001$
3	(Constant)	-.003	.048	-	-.056	.955
	avg_success_rate	.904	.088	.790	10.322	$< .001$

Note. Dependent Variable: average_skill_mastery

Table 5.1.19: Coefficients of challenge dataset

5.1.4.2 Coding dataset

Table 5.1.20 shows the descriptives of the variables when the coding dataset was used.

	Mean	Std.Dev.	N
skill_mastery	0.50989	0.238917	64
avg_success_rate	0.54041	0.214724	64
avg_duration	335.4122	326.33291	64
avg_attempts	2.8972	1.17370	64

Table 5.1.20: Descriptives of linear regression - coding

In Table 5.1.21 we see that both the correlation between skill_mastery and avg_duration (r = -0.308, p = 0.007), and the correlation between skill_mastery and avg_attempts (r = -0.632, p = $<$ .001), are significant. The correlation between avg_attempts and avg_duration (r = 0.562, p = 0.000) is significant as well.

		Variables			
		skill_mastery	success_rate	duration	attempts
Pearson Corr.	skill_mastery	1.000	-0.203	-0.308	-0.632
	success_rate	-0.203	1.000	-0.086	-0.038
	duration	-0.308	0.086	1.000	0.562
	attempts	-0.632	-0.038	0.562	1.000
Sig. (1-tailed)	skill_mastery	.	0.054	0.007	< .001
	success_rate	0.054	.	0.250	0.382
	duration	0.007	0.250	.	0.000
	attempts	0.000	0.382	0.000	.

* all predictors are average. avg_ removed for space reasons

Table 5.1.21: Pearson Correlation of coding dataset

In Table 5.1.22 we see the overall fit of the model. For the last model the R^2 value is 0.451, meaning that avg_attempts and avg_success_rate accounts for 45.1% of the variation in skill_mastery. Seeing that the first model that contained avg_duration as well has a R^2 value of 0.458 we see that avg_duration only accounts for 0.7% of the variation (0.458 - 0.451 = 0.007).

Model	R	R^2	Adjusted R^2	Std. Error of the Estimate	Durbin-Watson
1	.677[a]	.458	.431	.180223	
2	.671[b]	.451	.433	.179958	2.029

a Predictors: (Constant), avg_attempts, avg_success_rate, avg_duration
b Predictors: (Constant), avg_attempts, avg_success_rate
c Dependent Variable: skill_mastery

Table 5.1.22: Model Summary[c] of coding dataset

Table 5.1.23 displays the model parameters for both steps in the model. Both avg_success_rate and avg_attempts received a significance lower than 0.05, which is why these were the remaining values in the model. From the B values we see that a 1-unit increase in avg_success_rate contributes to an expected decrease of 0.253 units in skill_mastery. For avg_attempts we see that for every 1-unit increase, we expect a decrease in skill_mastery of 0.130 units.

		Unstd. Coefficients		Std. Coefficients		
	Model	*B*	Std. Error	*Beta*	*t*	*Sig.*
1	(Constant)	1.040	.086		12.029	< .001
	avg_success_rate	-.266	.107	-.239	-2.492	.015
	avg_duration	-7.681E-5	.000	.105	.906	.369
	avg_attempts	-.142	.024	-.700	-6.058	< .001
2	(Constant)	1.024	.085		12.117	< .001
	avg_success_rate	-.253	.106	-.228	-2.399	.020
	avg_attempts	-.130	.019	-.640	-6.742	< .001

Note. Dependent Variable: average_skill_mastery

Table 5.1.23: Coefficients of coding dataset

5.2 Questionnaire

In this section, the results gained from the data analysis of the questionnaire responses are presented. 33 students answered the questionnaire.

5.2.1 System Usability Score

The results from the System Usability Scale questions from the questionnaire are listed in Table 5.2.1. In the table the average score for each question is listed, as well as the total average score and the total average SUS Score (which is the total average score times 2,5). The total SUS Score for the application was 71,65.

Question	Score (N=32)
Frequent use	2,53
Unnecessarily complex	2,81
Easy to use	3,09
Would need assistance	3,41
Functions well integrated	2,66
Inconsistency	2,59
Quick to learn	3,09
Cumbersome/awkward	2,86
Confidence using	2,53
Required learning	3,06
Total	28,66
SUS Score	**71,65**

Table 5.2.1: System Usability Score (SUS)

5.2.2 Customization

In Figure 5.2.1 the results given in the questionnaire regarding the customization feature of the application are represented.

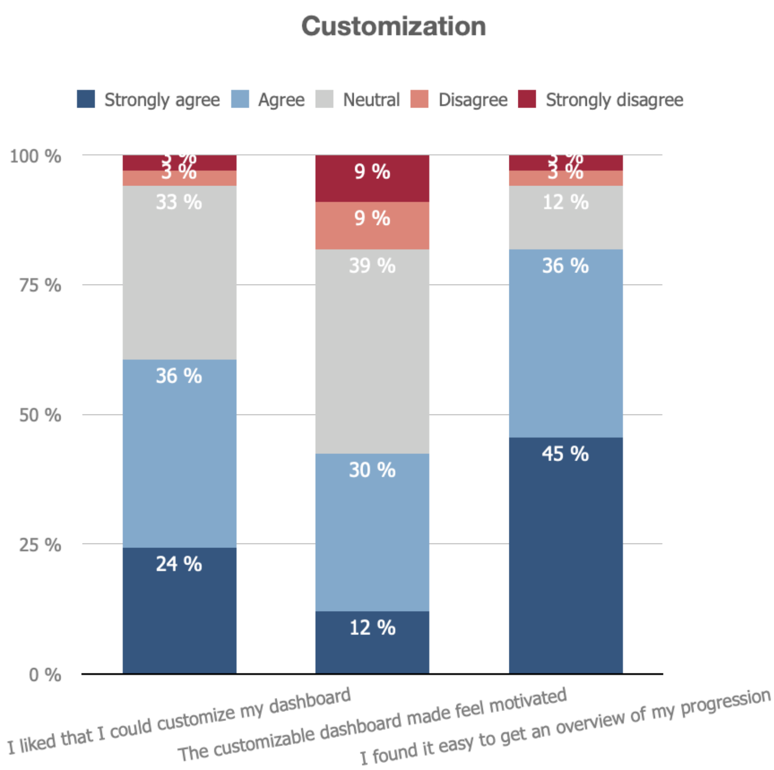


Figure 5.2.1: Questionnaire response (n=33) regarding customization

5.2.3 Components

In the questionnaire, the respondents were asked about their opinions on each component they chose to include in their personalized dashboard. Each optional component is presented in Section 3.7.4. The statements presented were as follows:

- **Q1** - I frequently used the component
- **Q2** - The component made me feel more engaged
- **Q3** - The component gave me a better overview of my progression
- **Q4** - I found it easy to extract relevant information from the component
- **Q5** - The component made me feel motivated

Table 5.2.2 shows how many of the respondents included each of the components in their customized dashboards, as well as the components' average rating out of 5 on perceived usefulness.

Component	**N**	**Perceived Usefulness**	**Std.Dev.**
Activity History	13	3.46	0.88
Exercise Graph	14	3.43	1.09
Exercise Planner	7	3.71	0.49
Statistics	18	3.72	0.83

Table 5.2.2: Perceived usefulness of each choosable component

The Figures 5.2.2, 5.2.3, 5.2.4, 5.2.5 show the feedback from the respondents on each of the different optional dashboard components.

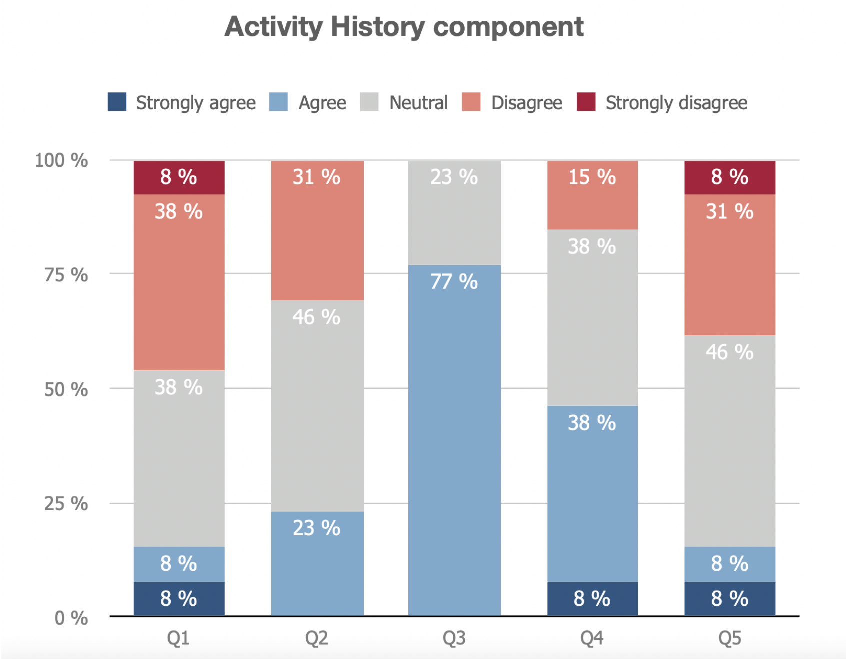


Figure 5.2.2: Questionnaire response (n=13) regarding the activity history component

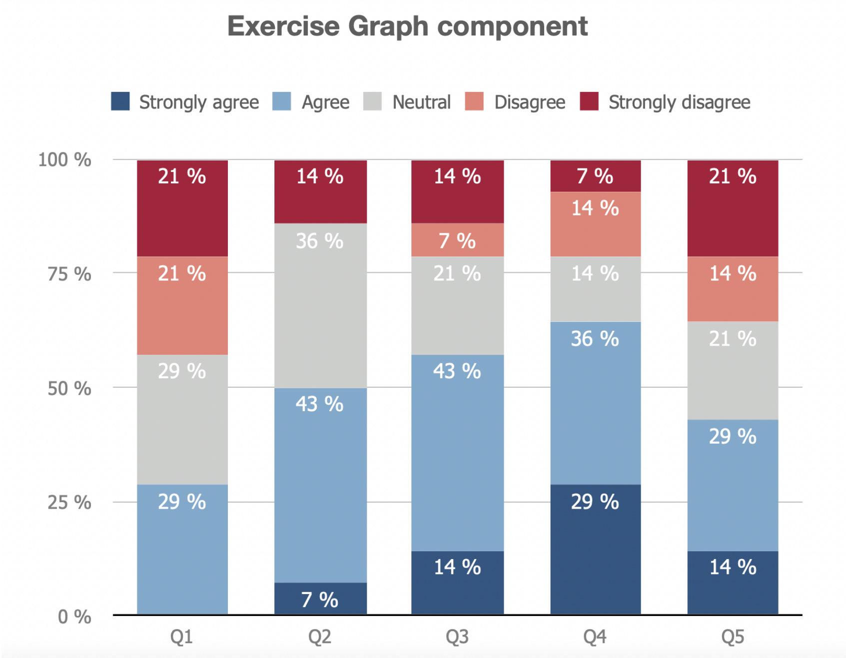


Figure 5.2.3: Questionnaire response (n=14) regarding the exercise graph component

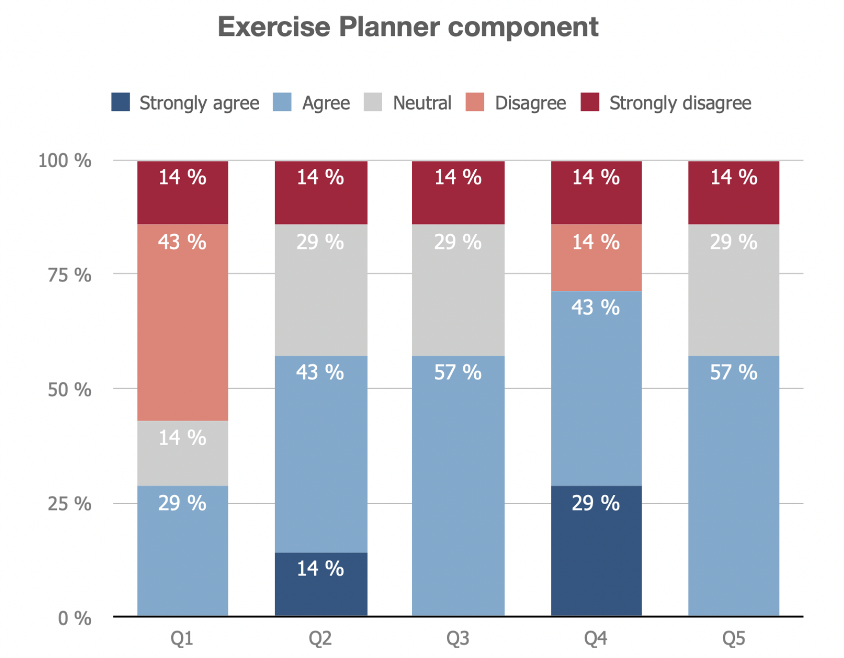


Figure 5.2.4: Questionnaire response (n=7) regarding the exercise planner component

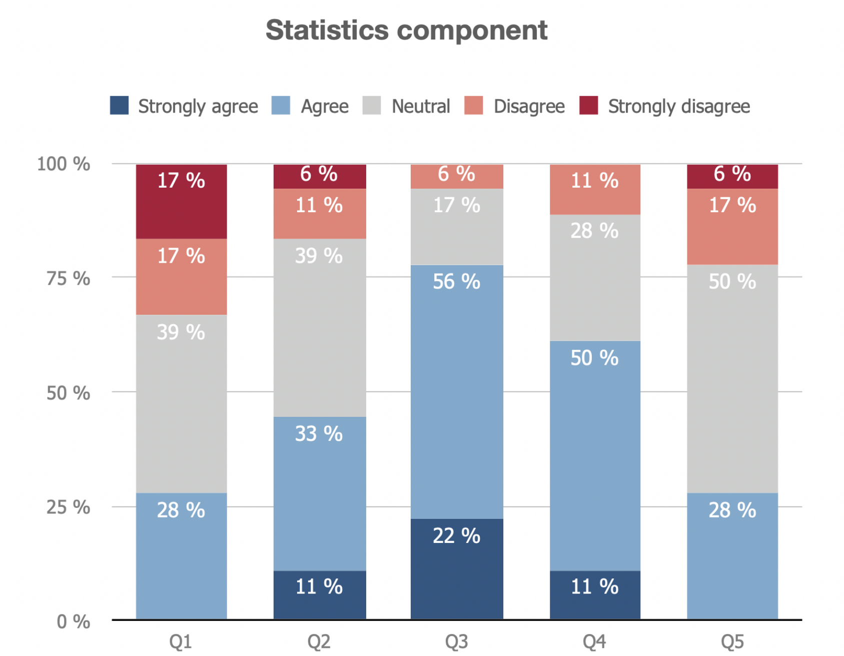


Figure 5.2.5: Questionnaire response (n=18) regarding the statistics component

Leaderboard

Table 5.2.3 shows how many of the respondents included the leaderboard in their customized dashboards, distinguishing between 1st-year and 2nd-year students. Additionally, the table displays the average rating, on a scale of 1 to 5, that the leaderboard received in terms of perceived usefulness. The students could change their configuration at their own discretion. Out of the 17 students, two initially opted in but later opted out. One student initially did not participate in the leaderboard but opted in later. The results from the questionnaire regarding the leaderboard are represented in Figure 5.2.6.

Component	N	1st year	2nd year	Perceived Usefulness	Std.Dev.
Leaderboard	17	4	13	3.88	1.17

Table 5.2.3: Perceived usefulness of leaderboard

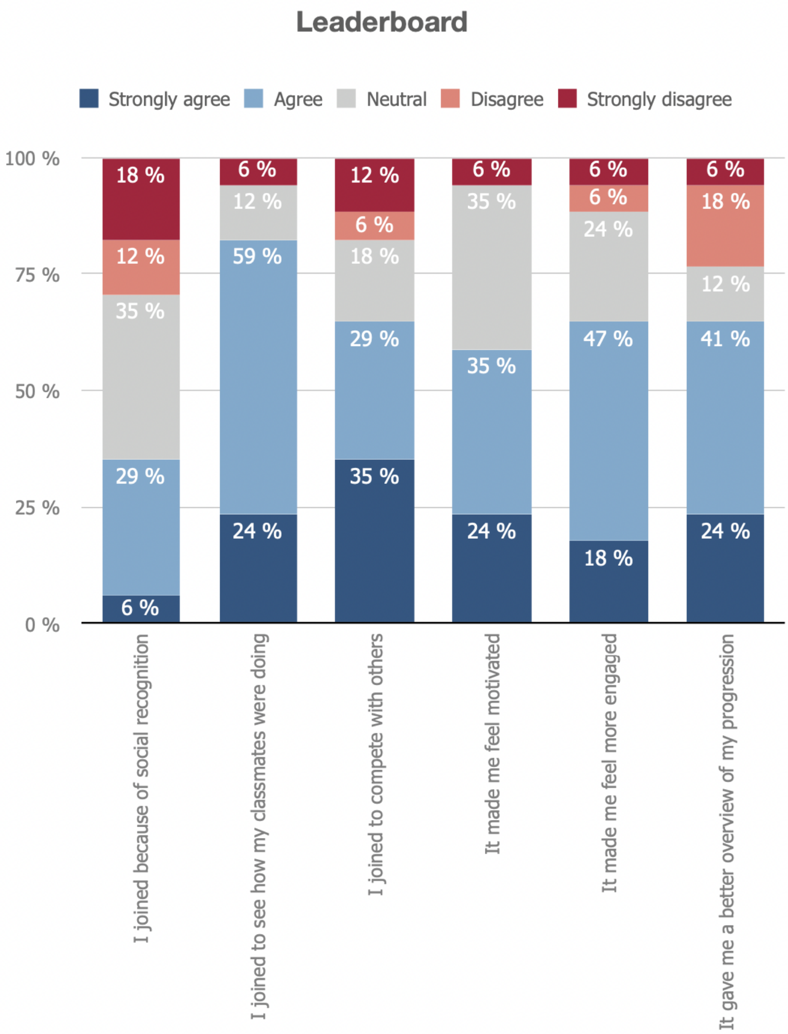


Figure 5.2.6: Questionnaire response (n=17) regarding the leaderboard

5.2.4 Exercise type order

The respondents were asked to sequence in which order they preferred to do *examples*, *challenges*, and *coding exercises*. Of the respondents, The results can be seen in Figure 5.2.7. Second, the respondents were asked to rank their reasoning when selecting exercises from first to last choice. The most popular first- and second choice was which topic the exercise belonged to. The least popular decision to pick an exercise was whether it belonged to the recommended section on the topic page. The results are charted in Figure 5.2.8.

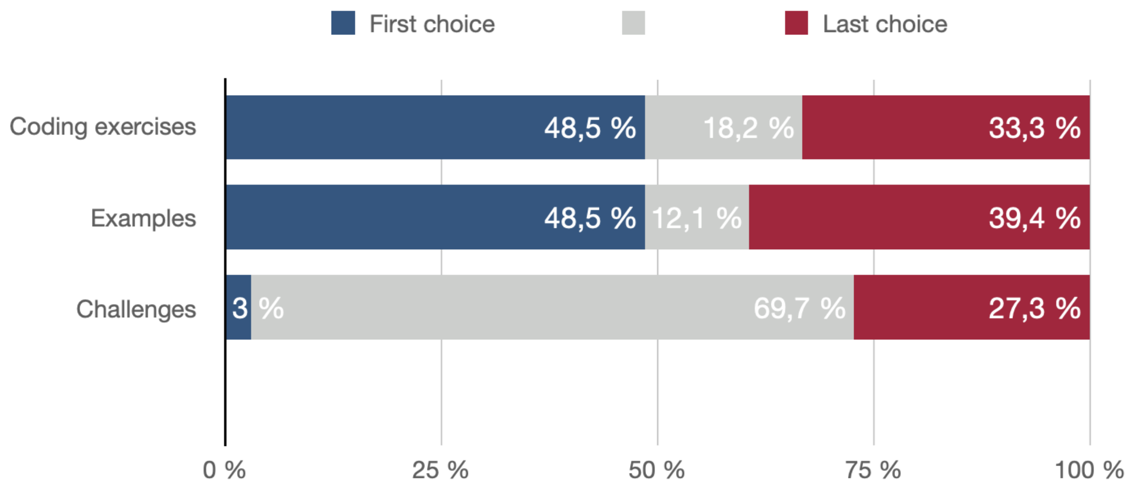


Figure 5.2.7: Preferred order when selecting exercises

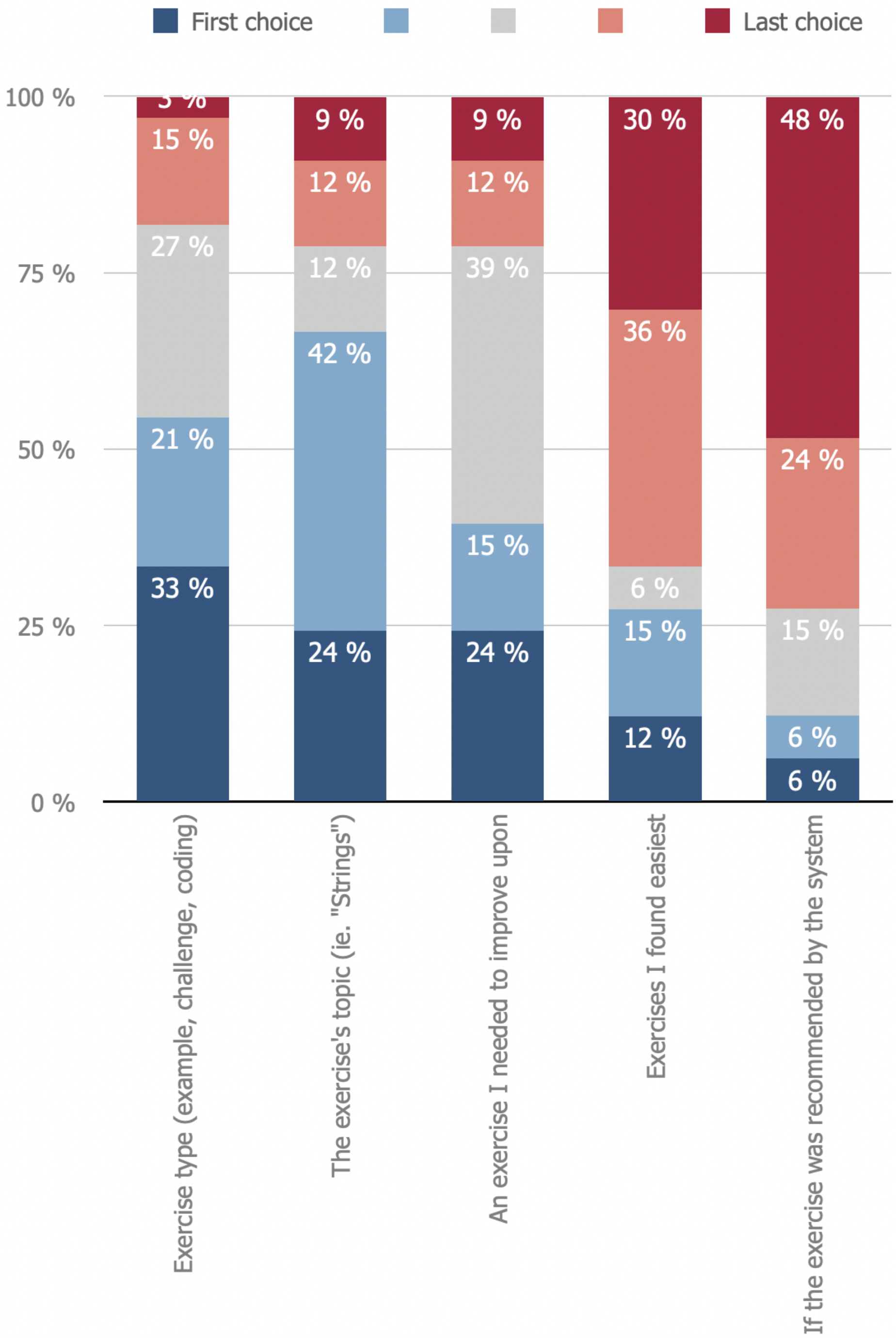


Figure 5.2.8: Reasoning when selecting exercises

5.2.5 General Feedback

The last question of the questionnaire was an open question where the respondents were given the opportunity to give any sort of comments regarding Progresso. Some of the feedback is stated in the section below.

A lot of the feedback received was targeted toward the exercises themselves and the exercise view;

- F1 - *"the coding sheet took time to understand"*
- F2 - *"the compiler had shitty output"*
- F3 - *"I wish it saved my code/solution for later"*
- F4 - *"I would like to see my answer to the questions because that would be helpful in other assignments"*
- F5 - *"it was cumbersome that the exercises were in a different tab than the application."*

Some responders were missing hints in the coding exercises;

- F6 - *"more hint in the coding exercises"*
- F7 - *"It would also be helpful to get feedback when you don't get it right, maybe some tips or something?"*
- F8 - *"it would be nice to be able to get tips/answer after you fail the exercise some amount of times"*

Some responders commented on wanting the system to include more exercises;

- F9 - *"I wish there were more tasks on writing an entire program"*
- F10 - *"There could be more or larger coding exercises. Some topics only had 1 or 0"*

One responder commented on the performance of the system;

- F11 - *"the loading time on some of the coding exercises made me impatient"*

Other responders experienced issues with bugs, for example;

- F12 - *"Leaderboard was not tracking my progress as it should. I had done everything, but the leaderboard was way behind."*
- F13 - *"bugs where one must submit the same answer several times before it accepts it as correct, and most of my progress was lost and I had to start over again"*
- F14 - *"I had a persistent problem where upon submitting code I would be told that I had to re-authenticate and reload the page"*

One user had issues with the text;

- F15 - *"I find it non-productive towards learning; it's too much text (most people would like a video explanation instead)"*
- F16 - *"Most people have a different mother tongue making it more difficult for them to understand the tasks"*

One responder wanted the system to provide more feedback;

- F17 - *"I would appreciate entire programs with feedback from the system"*

Regarding the customization one responder commented that;

- F18 - *"I felt like the optional components were too similar to each other"*

One specific comment made about the design of the system was that;

- F19 - *"the "continue where you left off" box, made me think I had some unfinished task because of the yellow color"*

Some of the positive feedback received included;

- F20 - *"Great system that I will use even after finishing my degree"*
- F21 - *"great system that definitely motivates"*
- F22 - *"Really nice program"*
- F23 - *"It was an easy-to-use interface, and I felt like it had all the right components to make it pleasant for the user"*

The complete log of feedback given by the responders can be found in Appendix E.1.

CHAPTER

SIX

DISCUSSION

This chapter aims to delve into the findings presented in Chapter 5, and examine the research questions in light of these obtained results. Section 6.1 aims to answer the first research question, while Section 6.2 aims to answer the second one. These are followed by Section 6.3, which explains the implications of the findings, and Section 6.4, which explains the various limitations that may have influenced the results.

6.1 Predictors of skill mastery

RQ1 calls for several measures of comparing skill mastery with categorical and continuous variables and is expressed as *"What predictors influence the learning outcome in a learning environment?"*.

One should first establish whether the validity of skill mastery determined by BKT is within acceptable limits. Classic BKT obtains an AUC of 0.73 when trained on the publicly available ASSISTments dataset [83]. pyBKT reported an AUC of 0.699 when using the KT-IDEM model [42] on the ASSISTments dataset. From the cross-validation process of the Progresso dataset, pyBKT obtains an average AUC of 0.671. Thus, on the Progresso dataset, the model fit is slightly less accurate than that of the ASSISTments dataset. However, one should note that the sample size of Progresso dataset is around 2743 records, while the version ASSISTments pyBKT used exceeded 500 000 records [83]. Thus, we can say that the validity of skill mastery in this study is similar to that of prior research. For extra safeguarding, the success rate, which measures learning outcomes differently, is introduced in addition to skill mastery for the subquestions of research question 1, when possible.

Table 5.1.1 shows that some skills obtained an AUC of less than 0.5, which suggests that the model classifies no better than random guessing. These skills were excluded when calculating the average skill mastery of each student. This also includes skills that obtained a Not a Number (NaN) AUC value due to insufficient sequence quantity in each fold.

Attempts
The team investigated whether the number of *attempts* has any significant impact on skill mastery. In previous research, skill mastery has been inferred by the ratio between correct and wrong answers [8] and Bayesian statistics [10]. However, little has been reported on the number of attempts submitted. Table 5.1.17 demonstrates the Pearson correlation in the challenge dataset. The results reveal a negative Pearson correlation (-0.679) between skill mastery and *attempts*, indicating that the number of *attempts* decreases as skill mastery increases. If one considers the success rate as a measure of mastery as well, the negative correlation (-0.909) points to the same result. Table 5.1.21 concerning correlation in the coding dataset further supports the observation from the challenge dataset by proving the same negative correlations between skill mastery and *attempts* (-0.632). From a broader perspective, this trend can be attributed to higher-performing students arriving at the correct answer more quickly, resulting in fewer *attempts*. However, it is important to note that correlation does not imply causation. Looking at the regression analysis in Table 5.1.18 for challenges suggests that removing the *attempts* in the backward step of the analysis causes a marginal 1% decrease in the explained variance in skill mastery. This minor shift is reflected in a correspondingly small change in the R^2 value.

However, the marginal change in R^2 upon removing the variable *attempts* from the model suggests that despite the strong correlation observed, *attempts* do not significantly contribute to the model's explanatory power over the variance in skill mastery. This seemingly paradoxical result could be attributed to one factor. The success rate may already account for the information *attempts* provided to the model. Hence, eliminating *attempts* doesn't reduce the explanatory power significantly since the success rate already encapsulates the relevant information. This interpretation aligns with the understanding that a higher number of attempts is typically associated with a lower success rate, given how success rates are calculated. Therefore, the explanatory power of skill mastery is most present in Model 3 of Table 5.1.17 when only the success rate is considered, making the inclusion of *attempts* redundant.

Duration
We also analyzed whether the duration of completing an exercise has an impact on skill mastery. No t-tests were conducted due to *duration* not being normally distributed. The correlation tables for challenge, as can be seen in Table 5.1.17, suggest a very weak positive correlation between duration and skill mastery. Similarly, the coding dataset in Table 5.1.21 suggests a weak positive correlation between *duration* and skill mastery. However, both linear regression models for the challenge and coding datasets exhibits less than a 1% decrease in explanatory power (R^2), indicating that duration does not severely affect skill prediction. The unstandardized beta coefficients for both datasets, as displayed in Table 5.1.19 and 5.1.23, show that a one-unit increase in skill mastery causes ≈ 0 increase in *duration*. One reasonable explanation for this observation is that weaker students have a small duration due to random guessing, and higher-performing students tend to conclude more quickly. Secondly, weaker students may obtain a longer duration because they

take more time to conclude, while higher-performing students do not randomly guess and spend a longer time but with a lower attempt count.

Another possible explanation for the weak correlation between *duration* and skill mastery is that spending a longer time on an exercise does not necessarily indicate increased learning. It is possible that not all exercises yield the same learning outcome, and therefore, the relationship between time and learning may vary across exercises. Additionally, since the study was performed in an uncontrolled learning environment, the recorded duration may not accurately reflect the true on-task duration. Another study that posited a controlled environment reported a $R^2 = 10\%$ explanatory power of duration on learning outcome [84]. Another analysis of on-task behavior also supports the notion that duration is a weak predictor of learning outcome [14].

This finding aligns with our previous understanding gained from the state-of-the-art findings reported in the background section that the length of a task alone does not exclusively contribute to learning. Rather, it requires a multifaceted involvement from the student, including attention, cognitive capacity, and cognitive comprehension [85]. Therefore, simply spending more time on a task does not guarantee proficiency in a certain skill.

Class

The team determined whether there is any relationship between skill mastery and class. The results in Table 5.1.10 indicates that the first-year students obtained a lower success rate (0.42) compared to the second-year students (0.57) in the coding dataset. This difference is statistically significant according to the t-test ($p<0.05$) presented in Table 5.1.11. Additionally, skill mastery was 0.34 for the first-year students and 0.57 for the second-year students, with a statistically significant result. Since all parameters of the BKT model were held constant for both classes, this difference in skill mastery and the success rate is unlikely due to any biases in how the model was applied. This supports the notion that the observed difference reflects a genuine difference in success rate and skill mastery between the two classes. In contrast, for the challenge dataset, the skill mastery was 0.476 for the first-year students and 0.472 for the second-year students, as can be seen in Table 5.1.3. However, this difference was not statistically significant ($p>0.05$). This could indicate that both groups might have encountered similar difficulties when faced with *challenge* tasks, regardless of their academic year. This assumption is supported by the fact that the challenge tasks have a 25% probability of guessing correctly, unlike the coding exercises where no multiple-choice options are given. In conclusion, the results indicate that LA dashboards that serve exercises that put a heavy cognitive load on the students will impact their probability of mastering the skill. This difference is especially visible when different-year students participate. The observed difference between first-year and second-year students highlights the need for the LA dashboard to even the skill gap through appropriate measures. One potential approach is to provide hints tailored to the first-year students' prior skill mastery.

Learning path

We examine whether the type of content visited first impacts skill mastery. Progresso allowed for three unique ways of starting their learning path by examples, challenges, or coding exercises. Prior research proposes how the entire learning path should be considered in order to identify which learning paths yield the highest learning outcome [7]. However, the team believes that the initial path initiation complements the existing research by examining the student's attitudes toward learning since the different exercise types inherently teach the skill differently with different cognitive loads. Based on the findings presented in Figure 5.2.7, it appears that students generally start with examples to acquire knowledge on the topic. They then proceed to conduct challenges to build confidence. Lastly, they perform coding exercises that impose a higher cognitive load. This type of learning path aligns with the cognitive load theory because they scaffold their prior knowledge based on the cognitive load the exercise possesses. However, this could mean that high-performing students skip the knowledge scaffolding provided by examples and challenges, and jump directly to the exercises that require more cognitive effort. In Table 5.1.7 we see that students who went straight to coding exercises demonstrated higher average skill mastery in the challenge dataset than those who started with examples. Similarly, in the same table, the before-task behavior of selecting coding exercises first also constitutes fewer attempts. However, the results are not statistically significant. This could be attributed to the fact that students in Progresso were handed out tasks to complete, where the first subtask was to complete examples. However, 13 students disregarded this instruction and jumped straight to later subtasks that involved coding exercises. Prior research has reported that self-regulated learners are more efficient in identifying learning content that yields the maximum learning outcome [27], which possibly can be attributed to the decision of the 13 students who chose coding exercise first. Nevertheless, we can not confidently say in our study that students who did coding exercises as their first observation have a higher probability of mastering a skill.

Leaderboard

Lastly, we examine the relationship between skill mastery and leaderboard participation. In the challenge dataset, students who participated in the leaderboard exhibited a higher skill mastery prediction compared to those who did not participate, as shown in Table 5.1.6. This result was statistically significant with a medium effect size. However, the corresponding t-test for the coding dataset did not prove a statistically significant result. One possible explanation for this disparity could be that when students are engaged in less cognitively demanding tasks, they may be more inclined to showcase their academic performance to their peers. In such circumstances, a leaderboard is an effective platform for students to demonstrate their mastery, especially when they are confident in their ability to answer correctly.

From the background section, it was found that leaderboards improve the learning process but does not impact learning outcome [58]. The results from the leaderboard participation in the challenge dataset indicate a different trend. However, it is important to consider that students who participate in the leaderboard may possess certain traits that contribute to their higher learning outcome and performance. For instance, students participating in the leaderboard might be inherently more motivated or competitive, possess a higher degree of self-regulation skills, or have a greater inherent proficiency for the subject matter, all of which could contribute to their enhanced performance, independent of their leaderboard participation.

Figure 5.2.6 could potentially highlight the possible reason behind students' decision to display their mastery on the leaderboard. 64% of the respondents agreed or strongly agreed that they joined the leaderboard to "compete with others". Students may be more motivated to engage in competition with their peers when they feel confident in their ability to achieve mastery [60]. Additionally, 83% of the respondents strongly agreed or agreed that they joined because they "wanted to see how their classmates were doing". As a consequence, the leaderboard could potentially serve as more than just a competitive space. It may also function as a social exchange platform where students compare their progress with their peers not from a place of competition but as part of a broader, interpersonal, and less adversarial level.

6.2 Designing learner analytics dashboards

The second research question concerns the design of learner analytic dashboards and how they can be optimized to enhance engagement. It's expressed as *"How can we design learner analytics dashboards in order to boost student engagement?"*.

Defining and measuring engagement present a challenge due to its multifaceted nature. It encompasses active participation, interest, and motivation, reflecting how invested the students are in what they do [86]. Engagement is a significant term because it is strongly associated with positive learning outcomes and achievement [86]. The questionnaire given to the students only gathered self-reported data about the students' perception of their own engagement and motivation. However, it provided some insight into which aspects of the learner analytic dashboard that engaged the most. By taking into account the students' opinions on usefulness, customization, and general feedback, these results can together contribute to the development of guidelines for designing effective learner analytic dashboards.

Customization

As stated in the background section, having a dashboard that can be customized to please every learner's unique needs seems highly valued. Figure 5.2.1 highlights that a substantial proportion (60%) of students agreed or strongly agreed with the ability to personalize their dashboards. Regarding motivation, only 18% of the participants disagreed or strongly disagreed with the notion that a customizable dashboard makes them feel more motivated. One student said that "I felt like it had all the right components to make it pleasant for the user". This finding aligns with previous studies, where several studies indicate that students express interest in being able to customize their learner dashboards. The questionnaire results further validate the positive reception of the customization opportunity. This positive outcome can be attributed to the concept of self-regulation discussed in Section 2.2.1, which emphasizes the significance of an individual's contribution to their own success. Autonomy plays a crucial role in self-regulation and performance [87], and having the chance to customize their own dashboard can greatly contribute to a sense of autonomy within the learning environment. The varying opinions of the respondents on the different components in the dashboard, as can be seen in the questionnaire results in Section 5.2, highlight the fact that each learner possesses unique needs, which again emphasizes the importance of customization as a fundamental element of a learner analytics dashboard.

In contrast, the most popular dashboard setup was, after all, with zero selected components from Figure 5.1.1. Nine students decided to configure their dashboards without any data visualization aids. Five of them never selected components from the start, while four selected at least one but deselected them later. The reason behind this change is not obvious. Statement F18 of the questionnaire could point to one explanation. "I felt like the optional components were too similar to each other" was made by one of the four students who first selected components but ultimately deselected all. This suggests that the objective of customizable LA dashboards shouldn't just achieve customization but rather the implementation of accurate components that serve a distinct purpose. However, it could also point to the fact that the students perceive anything but exercise completion as a distraction or simply do not want and value customization.

Engagement

As mentioned above the students gave self-reported data on their own engagement and motivation. It is also valuable to look at the usage of the different components, seeing that interest is a part of the concept of engagement [86]. Comparing the results of the different choosable components shows that they affect student engagement and motivation to varying degrees. The most chosen component is the statistics component (18 respondents), which aligns with the finding by Roberts et al. that students would want to see general statistics about their performance [5]. It also aligns with the research by Kupczynski et al. that there is a correlation between frequency of learning and performance [79], and that the "daily streak" element of the statistics component contributes as a motivation to keep up this frequency. Even though the statistics component was the most popular choice, only 44% of the respondents said it made them feel more engaged, and only 28% said they used the component frequently. It is difficult to assume the reason behind this lack of engagement and adoption. The stats component is the most data-rich, where all metrics collected are combined. Data overload could be one potential contributor. Another could be that duration and attempts are not necessarily good predictors of performance, as seen in the quantitative data analysis. As such, the students might not perceive them as particularly accurate in explaining their academic performance.

From the usage data gathered from Progresso, which can be seen in Figure 5.1.1, we see that thirty-one students, accounting for roughly half the population, chose the exercise planner component. However, only four students interacted with it. Very low engagement can be interpreted differently. Firstly, planning and setting goals have been reported to be a highly desired feature of LA dashboards [5][6]. Goal-setting forms a fundamental component of self-regulated learning, by establishing personal objectives, learners instigate self-focused feedback cycles, which they can use to assess their effectiveness and adjust their performance [27]. The exercise planner consisted of very basic features like a list of todos, a checkbox, and a deadline. One possible explanation for the low usage is the loose coupling between adding a todo and the learning content. For example, the student should have been able to create todos from a list of available learning content rather than having to type the exercise manually. If there is no apparent connection between the exercise planner component and the rest of the application, the students might choose to plan their course of action by alternative goal-setting and progress-tracking applications. Another possible solution could have been that the students set their learning goals with guidance from their teachers, seeing that self-regulation is not an easy task and not something that every student automatically does [28].

Perceived usefulness

In the questionnaire, the students were also asked about each component's perceived usefulness. According to Karahanna et al., perceived usefulness is defined as "the degree to which a person believes that use of a system would improve his or her performance" [88]. Their opinions on the perceived usefulness contribute to the suggested design guidelines in Table 6.2.1. From Table 5.2.2 and 5.2.3, we see that the leaderboard component received the highest score (3.88 out of 5). The students see the leaderboard as the most useful for their learning process. The statistics component (3.72) and the exercise planner (3.71) tightly follow the leaderboard and are also seen as useful to the students. The remaining two components also received relatively good scores on perceived usefulness, with the lowest being the exercise graph (3.43) and the activity history (3.46). Based on these scores, components related to the student's work history might be less critical when designing the optimal learner dashboard.

Leaderboard

The previous research on the topic of gamification and leaderboards is conflicted on whether leaderboards positively contribute to learner analytics dashboards. The results from the questionnaire show that the leaderboard component received the highest percentage of making students feel more engaged, with 65%. This confirms that the students, in general, are positive about having a leaderboard component included in their dashboards and confirms what several studies have found about competition significantly increasing engagement [54][56]. An essential factor to the leaderboard being so well received might be that the students had the opportunity to choose their names/be anonymous, seeing that studies have shown that students show skepticism to be easily identified on a leaderboard based on performance [54]. Another contributing factor to its popularity could be that only the top 10 performers were visible at any given time, allowing low-performing students to observe their peers without others seeing their scores.

When asked about their reasoning for joining the leaderboard, 83% agreed that it was to see how their classmates were performing, while 64% agreed that it was for the sake of competition. This confirms that students perceive it as valuable to be able to compare their positions with their peers, as mentioned in a previous study by Park et al. [22]. However, only 35% agreed that they joined the leaderboard for social recognition. Some studies have emphasized the importance of social recognition as a reason for including leaderboards in learner dashboards [57], but the findings in this study contribute to a different perspective.

Design

Studies have emphasized the importance of design in learner analytics dashboards. User-friendliness, clear instructions and a purposeful structure are highlighted aspects of the design process [47]. From the System Usability Scale in Table 5.2.1, we see that the Progresso application received a score of 71.65. According to Bangor et al. a SUS score above 70 is considered acceptable [89]. Among the different segments of the SUS, the highest score was obtained for the statement "I think I would need support to be able to use the system" (3.41), indicating the opposite - that they would not need support using the system. Followed by this statement were the statements "I thought Progresso was easy to use" (3.09), and "I would imagine that most people would learn how to use Progresso very quickly" (3.09). These high scores confirm that Progresso is highly user-friendly and quick to comprehend. Additionally, the feedback from the different components aligns with these findings, as over 60% of students agreed that it was easy to extract relevant information from the exercise graph, the exercise planner, as well as the statistics component. Some comments from the general feedback section of the questionnaire were targeted towards the design such as; F23 - "it was an easy-to-use interface", F20 - "great system that I will use even after finishing my degree" and F21 - "great system that definitely motivate".

The segment that received the lowest score from the SUS was "I think that I would like to use this system frequently" with a score of 2.53. Although this score is the lowest, it is still considered acceptable. One possible explanation for this result is that the experiment was conducted as part of a Java programming course, and the number of available tasks within the system was limited. As students progress through the course, they may desire more challenging exercises to further enhance their programming skills. Some feedback was given that could further confirm this explanation; F9 - "I wish there were more tasks on writing an entire program" and F10 - "there could be more or larger coding exercises. Some topics only had 1 or 0".

From the questionnaire, several comments were made about the application's performance. Statement F11 - "the loading time on some of the coding exercises made me impatient" suggests that the team failed to optimize the user interface by the chosen non-functional requirements. However, it is important to note that the statement concerns the learning content and not the dashboard by itself. Although this issue was outside the team's ability to fix, it shows that LA dashboards that integrate third-party content are not stronger than its weakest link. Statements F1 through F10 all point to weaknesses in the learning content, and students mistakenly perceive this content as part of the dashboard, despite Progresso only providing links to it. When designing a learner analytics dashboard it is crucial to ensure a seamless transition between dashboard and exercises. Therefore, it is for future development recommended to design and integrate tasks in the same style as the dashboard itself to maintain consistency and improve the overall user experience.

The findings of the study on how we can design learner analytics dashboards in order to boost student engagement can be summarized into guidelines that are listed in Table 6.2.1.

ID	**Dimension**	**Recommendation**
G1	Customizability	Having a dashboard that can be customized is highly valued and can contribute to a sense of autonomy within the learning environment. However, it is equally as important to implement accurate components with a distinct purpose.
G2	Feedback	The students would want the system to provide feedback when they are doing exercises, tailored to their respective levels of knowledge.
G3	Goal and plan scheduling	Goal-setting is a highly desired feature and forms a component of the self-regulated theory. Incorporate a goal-setting tool such as an exercise planner to emphasize students' self-regulation.
G4	Leaderboard	Inclusion of a leaderboard positively contributes to learner analytics dashboards. It enables students to compare their own progress to their peers, but also contributes to a competitive learning environment that engages students. An important factor for a successful leaderboard is to enable students to be anonymous.

Table 6.2.1: Guidelines on how to design a learner analytics dashboard

6.3 Implication of findings

The findings of this study have confirmed that competitive environments, specifically leaderboards, in this case, show higher skill mastery in students who participate in them. This has been shown to hold true for the challenge type. Previous research has been conflicted on whether competition positively contributes to learning. Therefore, the results of this study should encourage the future development of learner analytics dashboards to incorporate leaderboards for future research. Moreover, the skill mastery was higher for second-year students, which asserts the need for individualized learning environments to narrow the skill gap. It was impossible to distinguish students' skill mastery based on duration and attempts. As such, other metrics should be measured to determine predictors who maximize learning outcome.

The Progresso dashboard is, to our knowledge, the first learner analytics dashboard to include optional goal scheduling and planning. This is a notable contribution to the field. The inclusion of goal scheduling aligns with self-regulating theory and holds great potential to increase learning outcomes if implemented satisfactorily. The Progresso dashboard received an acceptable SUS score and was well received, and most critiques were pointing to the learning content provider. Future research can draw inspiration from the Progresso dashboard when designing and developing future customizable learner analytics dashboards.

6.4 Limitations

The limitations of the study are presented in this section. Some factors that may have influenced the results are worth mentioning, and addressing these are important to create a comprehensive understanding of the study's findings.

6.4.1 Experiment

One significant limitation of the experiment phase is that it did not include a control group. The control group could have received the programming tasks but without access to the customizable Progresso dashboard. The lack of a control group greatly limits the opportunity to investigate and differentiate the effects of the Progresso dashboard. Therefore, the results are less reliable than what they could have been.

6.4.2 Data

The team's prior experience in statistics was limited, and no prior experience in regression analysis. The data analysis may have been limited and flawed, and the team may have failed to discover patterns. The team wished that the learning content API tracked more information on the student's exercise interaction.
Because of the implementation of how on-task duration was stored, the team had to delete a substantial proportion of null values in the dataset. This decreased the number of exercise attempts with a valid duration. Moreover, in many cases, the large difference between *visitedAt* and *completedAt* caused so many outliers that the team had to plot the durations on a scatterplot to determine by themselves where a reasonable cutoff point should be. This may have caused unintentional bias introduced in the duration variable. The skill mastery prediction per student is the sum of skill mastery prediction for all participated skills divided by the number of skills participated. As such, students who participate in very few skills may have an artificially high or low average skill mastery

estimation for the whole course. This may have impacted the classification process when determining whether skill mastery is dependent on the numerous independent variables examined. Moreover, the sample size of first-year students was much smaller than second-year students. pyBKT performs better with individual slip and guess parameters, but the sample size did not fulfill the data size requirement. To make the model equal between classes, the guess and slip estimates were parameterized by excise type. This means that the class determines the guess and slip parameter, not the individual student. This implies that it might be challenging to distinguish the proportion of variation in predictions that comes from the student, as opposed to that which comes from the skill content [42].

Prior research projects that have used pyBKT have, in almost all cases, used either the popular ASSISTments or Cognitivetutor datasets, which can offer a granularity much finer than Progresso. Estimating latent knowledge is a complex task, and the team felt like pyBKT was still the most acceptable model because of their familiarity with Python and the simplistic idea BKT is founded on. There may have been other models that would be much more suitable for the data the team collected.

6.4.3 Application

During the onboarding, the students enter a unique ID that the learning content API uses to track the interactions made with the exercises. The team was allocated 100 slots which means that few unique IDs remain. New instantiations would require the supervisor to contact the maintainers for Adapt2 to allocate new slots.

Progresso only includes 184 exercises, where many are easily completed. If the program is meant to be used for an extended period of time, it may not be enough resources. The learning content API gives very little feedback on coding exercises and the submission is not saved after the tab is closed. This was a wanted feature in several of the statements provided by the questionnaire.

During the development phase of Progresso, Next.js 12 was utilized as the framework. However, a significant update was announced and introduced during this phase - Next.js 13. While it is still in beta, this new version is set to replace the 12th version, which ironically contradicts with Progresso's chosen non-functional requirement of modifiability due to the substantial amount of migration required. The situation reveals that the technology behind Progresso requires frequent maintaining to keep up to date.

In addition to the Next.js update, Prisma ORM, underwent a major revision in April 2023. This update drastically reduced the serverless cold start time by a factor of nine, but unfortunately, it occurred after the team's experiment ended.

6.4.4 Questionnaire

One of the limitations of the study was the questionnaire. The fact that the researchers have relatively limited experience in this field may have had an impact on the structure and comprehensiveness of the questionnaire, as well as the quality of the questions asked. A better and more worked-through questionnaire design, with better and more relevant variables, could have led to the results having improved validity and reliability. In addition to this, the sample size of the respondents to the questionnaire is not representative compared to the participants in the experiment seeing that not all of them answered the questionnaire following the experiment phase.

In addition, one could argue that the participants should have answered a pre-questionnaire before the test trial to capture their motivation, opinions, and even prior experience for fixing the prior parameter for BKT. It is hard to predict what data is useful before data generation has been completed. Thus, the team failed to predict that a pre-test could have been useful.

CHAPTER

SEVEN

CONCLUSIONS & FURTHER WORK

The research started with a thorough literature review on the topic of e-learning. Then, Progresso was developed into a fully functional learning analytics dashboard that could provide answers to the research questions. Progresso was tested in an uncontrolled environment with 73 participants, where they could interact and solve Java exercises at their convenience. Finally, the qualitative and quantitative data were analyzed in the Results chapter and discussed in the Discussion chapter.

Low-level concepts of task behavior, such as duration, attempts, and learning path, did not prove to explain the variance in skill mastery of students significantly. A negative strong correlation was found between skill mastery and attempts. The duration had a weak negative correlation with skill mastery in the coding dataset and a very weak positive correlation in the challenges dataset. Neither of them contributed to a large explanation of variance in skill mastery. However, the significance between the means of skill mastery was notable in those who learned in a competitive environment vs. those who did not. A significant difference in the mean between skill mastery and the class was observed, but since the second-level class was more experienced, this finding does not prove much other than that the skill mastery prediction was correct in assuming that second-year students perform better.

Students favor customizable learning dashboards, but the implementation of what and which data ultimately decides the usefulness of the dashboard. Four suggested guidelines were made that should boost student engagement.

Since the scope of Progresso was limited to only a one-month period and a short time in development, the potential to expand the application scope remains as Progresso is open-source on GitHub [1].

[1] https://github.com/mathisfo/bug-free-adventure

7.1 Future Research

Further development of Progresso as an application, could improve feedback mechanisms of the application. Today's implementation uses arbitrary facts such as attempts, duration, and whether tasks are completed or not. A wider range of educational content should be made available so that the learning process is not only dependent on examples, challenges, and coding exercises. The increase in educational content could pave the way for new insights into what predictors maximize learning outcome.

Progresso's five customizable components only represent a small proportion of the extent you can visualize learner analytics. If the learning content is extended, it opens up more metrics to visualize. Optional components do not even have to conform to a dashboard view but can be much larger features such as messaging options between students. A more in-depth exploration of gamification elements can contextualize the relationship between gamification and skill mastery and not just through the leaderboard dimension.

The team acknowledges that certain data collected by Progresso could not be included in the quantitative research analysis due to the time-consuming statistical analysis of latent knowledge estimation. Progresso has saved all configuration changes students have done with their optional components in their dashboards. It could prove useful to check if a specific configuration yields a higher learning outcome. Also, it could be interesting to investigate the relationship between user behavior in toggling data visualization components and their academic performance. This raises the question of whether students tend to disable components that display negative performance during periods of poor academic progress.

Since Progresso stores the time intervals between exercise attempts and thus study sessions, future research could investigate the recency effects of latent knowledge estimation. Since BKT assumes that all observations are in order and proportionally equal, a new investigation could be made to improve the model by utilizing the BKT+Forget model. This could also investigate if different learning paths yield better retention of skill mastery.

Further research could possibly explore leaderboard participation as a classification problem through binominal regression analysis. The investigation could determine what type of learning content encourages the highest probability of participation in the leaderboard. Furthermore, cognitive behaviors, such as the learning path and interaction with the LA dashboard, could be examined. Another interesting aspect involves exploring whether it would yield different results if students freely chose exercises and exercise types, without having been handed tasks with instructions on what exercises to complete first. Further research could explore a learning environment where the students have no restrictions on what type of content is mandatory.

Since the questionnaire's validity is subpar, the conclusions derived from the usefulness of the data visualization components are limited. Future research could provide Progresso with more data visualization components that are more specific in their purpose to determine whether a specific set of selected components predicts skill mastery or if a component that adopts elements from other learning theories increases the learning outcome.

REFERENCES

[1] Shanshan Wan and Zhendong Niu. "An e-learning recommendation approach based on the self-organization of learning resource". en. In: *Knowledge-Based Systems* 160 (Nov. 2018), pp. 71–87. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.06.014. URL: https://www.sciencedirect.com/science/article/pii/S0950705118303204 (visited on 06/11/2023).

[2] Dimah Al-Fraihat et al. "Evaluating E-learning systems success: An empirical study". en. In: *Computers in Human Behavior* 102 (Jan. 2020), pp. 67–86. ISSN: 0747-5632. DOI: 10.1016/j.chb.2019.08.004. URL: https://www.sciencedirect.com/science/article/pii/S0747563219302912 (visited on 05/31/2023).

[3] Yi-Shan Tsai and Dragan Gasevic. "Learning analytics in higher education — challenges and policies: a review of eight learning analytics policies". In: *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*. LAK '17. New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 233–242. ISBN: 978-1-4503-4870-6. DOI: 10.1145/3027385.3027400. URL: https://doi.org/10.1145/3027385.3027400 (visited on 05/31/2023).

[4] Michail N. Giannakos, Patrick Mikalef, and Ilias O. Pappas. "Systematic Literature Review of E-Learning Capabilities to Enhance Organizational Learning". en. In: *Information Systems Frontiers* 24.2 (Apr. 2022), pp. 619–635. ISSN: 1572-9419. DOI: 10.1007/s10796-020-10097-2. URL: https://doi.org/10.1007/s10796-020-10097-2 (visited on 05/31/2023).

[5] Lynne D. Roberts, Joel A. Howell, and Kristen Seaman. "Give Me a Customizable Dashboard: Personalized Learning Analytics Dashboards in Higher Education". en. In: *Technology, Knowledge and Learning* 22.3 (Oct. 2017), pp. 317–333. ISSN: 2211-1670. DOI: 10.1007/s10758-017-9316-1. URL: https://doi.org/10.1007/s10758-017-9316-1 (visited on 06/06/2023).

[6] Clara Schumacher and Dirk Ifenthaler. "Features students really expect from learning analytics". en. In: *Computers in Human Behavior* 78 (Jan. 2018), pp. 397–407. ISSN: 0747-5632. DOI: 10.1016/j.chb.2017.06.030. URL: https://www.sciencedirect.com/science/article/pii/S0747563217303990 (visited on 11/22/2022).

[7] Vladimir Mikić et al. “Personalisation methods in e-learning-A literature review”. en. In: *Computer Applications in Engineering Education* 30.6 (2022). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.22566, pp. 1931–1958. ISSN: 1099-0542. DOI: 10.1002/cae.22566. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22566 (visited on 12/07/2022).

[8] Boban Vesin et al. “Adaptive Assessment and Content Recommendation in Online Programming Courses: On the Use of Elo-rating”. In: *ACM Transactions on Computing Education* 22.3 (June 2022), 33:1–33:27. DOI: 10.1145/3511886. URL: https://doi.org/10.1145/3511886 (visited on 12/07/2022).

[9] Philip I.S. Lei and António José Mendes. “A systematic literature review on knowledge tracing in learning programming”. In: *2021 IEEE Frontiers in Education Conference (FIE)*. ISSN: 2377-634X. Oct. 2021, pp. 1–7. DOI: 10.1109/FIE49875.2021.9637323.

[10] Albert T. Corbett and John R. Anderson. “Knowledge tracing: Modeling the acquisition of procedural knowledge”. en. In: *User Modeling and User-Adapted Interaction* 4.4 (Dec. 1994), pp. 253–278. ISSN: 1573-1391. DOI: 10.1007/BF01099821. URL: https://doi.org/10.1007/BF01099821 (visited on 04/27/2023).

[11] Anirudhan Badrinath, Frederic Wang, and Zachary Pardos. *pyBKT: An Accessible Python Library of Bayesian Knowledge Tracing Models*. arXiv:2105.00385 [cs]. May 2021. DOI: 10.48550/arXiv.2105.00385. URL: http://arxiv.org/abs/2105.00385 (visited on 04/26/2023).

[12] Matthew Montebello. “E-learning paradigms: A model to address known issues”. In: *2017 Computing Conference*. July 2017, pp. 1180–1189. DOI: 10.1109/SAI.2017.8252240.

[13] Wei Yuang et al. “A Review of the Research on the Prediction of Learning Outcomes in the Field of Learning Analytics”. In: *Proceedings of the 5th International Conference on Education and Multimedia Technology*. ICEMT ’21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 154–162. ISBN: 978-1-4503-9022-4. DOI: 10.1145/3481056.3481077. URL: https://dl.acm.org/doi/10.1145/3481056.3481077 (visited on 06/09/2023).

[14] Karrie E. Godwin et al. “The elusive relationship between time on-task and learning: not simply an issue of measurement”. In: *Educational Psychology* 41.4 (Apr. 2021), pp. 502–519. ISSN: 0144-3410. DOI: 10.1080/01443410.2021.1894324. URL: https://doi.org/10.1080/01443410.2021.1894324 (visited on 06/06/2023).

[15] Awaz Naaman Saleem, Narmin Mohammed Noori, and Fezile Ozdamli. “Gamification Applications in E-learning: A Literature Review”. en. In: *Technology, Knowledge and Learning* 27.1 (Mar. 2022), pp. 139–159. ISSN: 2211-1670. DOI: 10.1007/s10758-020-09487-x. URL: https://doi.org/10.1007/s10758-020-09487-x (visited on 05/31/2023).

[16] Shurui Bai et al. "From top to bottom: How positions on different types of leaderboard may affect fully online student learning performance, intrinsic motivation, and course engagement". en. In: *Computers & Education* 173 (Nov. 2021), p. 104297. ISSN: 0360-1315. DOI: 10.1016/j.compedu.2021.104297. URL: https://www.sciencedirect.com/science/article/pii/S0360131521001743 (visited on 06/12/2023).

[17] Chih-Ming Chen. "Intelligent web-based learning system with personalized learning path guidance". en. In: *Computers & Education* 51.2 (Sept. 2008), pp. 787–814. ISSN: 0360-1315. DOI: 10.1016/j.compedu.2007.08.004. URL: https://www.sciencedirect.com/science/article/pii/S0360131507000978 (visited on 06/11/2023).

[18] Prasenjit Basu, Suman Bhattacharya, and Samir Roy. "Online Recommendation of Learning Path for an E-Learner under Virtual University". en. In: *Distributed Computing and Internet Technology*. Ed. by Chittaranjan Hota and Pradip K. Srimani. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 126–136. ISBN: 978-3-642-36071-8. DOI: 10.1007/978-3-642-36071-8_9.

[19] Marina Anikieva. "The choice of customisation strategies in training: An overview of parameters and their systematisation". en. In: *Australasian Journal of Educational Technology* 37.3 (May 2021). Number: 3, pp. 170–186. ISSN: 1449-5554. DOI: 10.14742/ajet.6185. URL: https://ajet.org.au/index.php/AJET/article/view/6185 (visited on 04/19/2023).

[20] Briony J. Oates. *Researching Information Systems and Computing*. en. Google-Books-ID: VyYmkaTtRKcC. SAGE, Nov. 2005. ISBN: 978-1-4462-3544-7.

[21] Boban Vesin, Katerina Mangaroska, and Michail Giannakos. "Learning in smart environments: user-centered design and analytics of an adaptive learning system". en. In: *Smart Learning Environments* 5.1 (Oct. 2018), p. 24. ISSN: 2196-7091. DOI: 10.1186/s40561-018-0071-0. URL: https://doi.org/10.1186/s40561-018-0071-0 (visited on 12/06/2022).

[22] Yeonjeong Park and Il-Hyun Jo. "Development of the Learning Analytics Dashboard to Support Students' Learning Performance". en. In: (2015).

[23] Al Januszewski and Michael Molenda. *Educational Technology: A Definition with Commentary*. en. Google-Books-ID: JO3Yc0UuK74C. Routledge, Jan. 2013. ISBN: 978-1-136-50334-4.

[24] George Siemens and Phil Long. "Penetrating the Fog: Analytics in Learning and Education". en. In: *EDUCAUSE Review* 46.5 (Sept. 2011). Publisher: EDUCAUSE ERIC Number: EJ950794, p. 30. ISSN: 1527-6619. (Visited on 05/26/2023).

[25] Jose Luis Santos et al. "Goal-oriented visualizations of activity tracking: a case study with engineering students". In: *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. LAK '12. New York, NY, USA: Association for Computing Machinery, Apr. 2012, pp. 143–152. ISBN: 978-1-4503-1111-3. DOI:

10.1145/2330601.2330639. URL: https://doi.org/10.1145/2330601.2330639 (visited on 05/29/2023).

[26] Doug Clow. *An overview of learning analytics*. en. ISSN: 1356-2517. 2013. URL: https://www.tandfonline.com/doi/epdf/10.1080/13562517.2013.827653?needAccess=true&role=button (visited on 05/26/2023).

[27] Barry J. Zimmerman Schunk Dale H. "Self-Regulated Learning and Performance: An Introduction and an Overview". In: *Handbook of Self-Regulation of Learning and Performance*. Num Pages: 12. Routledge, 2011. ISBN: 978-0-203-83901-0.

[28] Barry J. Zimmerman. "Becoming a Self-Regulated Learner: An Overview". en. In: *Theory Into Practice* 41.2 (May 2002), pp. 64–70. ISSN: 0040-5841, 1543-0421. DOI: 10.1207/s15430421tip4102_2. URL: http://www.tandfonline.com/doi/abs/10.1207/s15430421tip4102_2 (visited on 06/12/2023).

[29] John Sweller. "Cognitive Load During Problem Solving: Effects on Learning". en. In: *Cognitive Science* 12.2 (1988), pp. 257–285. ISSN: 1551-6709. DOI: 10.1207/s15516709cog1202_4. URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1202_4 (visited on 06/09/2023).

[30] John Sweller. "Cognitive load theory, learning difficulty, and instructional design". en. In: *Learning and Instruction* 4.4 (Jan. 1994). Publisher: Pergamon, pp. 295–312. ISSN: 0959-4752. DOI: 10.1016/0959-4752(94)90003-5. URL: https://www.sciencedirect.com/science/article/pii/0959475294900035 (visited on 06/09/2023).

[31] Wolfgang Schnotz and Christian Kürschner. "A Reconsideration of Cognitive Load Theory". en. In: *Educational Psychology Review* 19.4 (Dec. 2007), pp. 469–508. ISSN: 1573-336X. DOI: 10.1007/s10648-007-9053-4. URL: https://doi.org/10.1007/s10648-007-9053-4 (visited on 06/09/2023).

[32] Seyed Yaghoub Mousavi, Renae Low, and John Sweller. "Reducing cognitive load by mixing auditory and visual presentation modes". In: *Journal of Educational Psychology* 87 (1995). Place: US Publisher: American Psychological Association, pp. 319–334. ISSN: 1939-2176. DOI: 10.1037/0022-0663.87.2.319.

[33] Mariel F. Musso et al. "Identifying Reliable Predictors of Educational Outcomes Through Machine-Learning Predictive Modeling". In: *Frontiers in Education* 5 (2020). ISSN: 2504-284X. URL: https://www.frontiersin.org/articles/10.3389/feduc.2020.00104 (visited on 06/09/2023).

[34] Hayward Andres. "The role of active teaching, academic self-efficacy, and learning behaviors in student performance". In: *Journal of International Education in Business* 13.2 (Jan. 2020). Publisher: Emerald Publishing Limited, pp. 221–238. ISSN: 1836-3261. DOI: 10.1108/JIEB-02-2020-0017. URL: https://doi.org/10.1108/JIEB-02-2020-0017 (visited on 06/09/2023).

[35] Nor Sa'adah Jamaluddin et al. "A Review of Learning Strategies towards Learning Outcome". In: *Int. J. Soc. Sci. Hum. Res. J. Soc. Sci. Hum. Res* 4.12 (2021), pp. 3647–3651.

[36] Victor Deak and Rivosa Santoso. "Learning Strategies and Applications in Learning Achievements". In: *International Journal of Social and Management Studies* 2.4 (2021), pp. 159–167.

[37] Joakim Caspersen and Jens-Christian Smeby. "The relationship among learning outcome measures used in higher education". In: *Quality in Higher Education* 24.2 (May 2018), pp. 117–135. ISSN: 1353-8322. DOI: 10.1080/13538322.2018.1484411. URL: https://doi.org/10.1080/13538322.2018.1484411 (visited on 06/07/2023).

[38] E. Ashby Plant et al. "Why study time does not predict grade point average across college students: Implications of deliberate practice for academic performance". en. In: *Contemporary Educational Psychology* 30.1 (Jan. 2005), pp. 96–116. ISSN: 0361-476X. DOI: 10.1016/j.cedpsych.2004.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0361476X04000384 (visited on 06/09/2023).

[39] MARIEL F Musso. "Understanding the underpinnings of academic performance: The relationship of basic cognitive processes, self-regulation factors and learning strategies with task characteristics in the assessment and prediction of academic performance". In: *Unpublished doctoral dissertation). University of Leuven, Leuven, Belgium* (2016).

[40] Mohammad Khajah, Robert V. Lindsey, and Michael C. Mozer. *How deep is knowledge tracing?* arXiv:1604.02416 [cs]. June 2016. DOI: 10.48550/arXiv.1604.02416. URL: http://arxiv.org/abs/1604.02416 (visited on 04/26/2023).

[41] Yumeng Qiu et al. "Does Time Matter? Modeling the Effect of Time in Bayesian Knowledge Tracing". en. In: ().

[42] Zachary A Pardos and Neil T Heffernan. "KT-IDEM: Introducing item difficulty to the knowledge tracing model". In: *User Modeling, Adaption and Personalization: 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings 19*. Springer, 2011, pp. 243–254.

[43] Zachary A. Pardos and Neil T. Heffernan. "Detecting the Learning Value of Items In a Randomized Problem Set". In: *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*. NLD: IOS Press, July 2009, pp. 499–506. ISBN: 978-1-60750-028-5. (Visited on 04/26/2023).

[44] Zachary A. Pardos and Neil T. Heffernan. *Determining the Significance of Item Order in Randomized Problem Sets*. en. Tech. rep. Publication Title: International Working Group on Educational Data Mining ERIC Number: ED539081. International Working Group on Educational Data Mining, July 2009. URL: https://eric.ed.gov/?id=ED539081 (visited on 04/26/2023).

[45] Zachary A. Pardos and Neil T. Heffernan. "Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing". en. In: *User Modeling, Adaptation, and Personalization*. Ed. by Paul De Bra, Alfred Kobsa, and David Chin. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 255–266. ISBN: 978-3-642-13470-8. DOI: 10.1007/978-3-642-13470-8_24.

[46] Ed de Quincey et al. "Student Centred Design of a Learning Analytics System". In: *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. LAK19. New York, NY, USA: Association for Computing Machinery, Mar. 2019, pp. 353–362. ISBN: 978-1-4503-6256-6. DOI: 10.1145/3303772.3303793. URL: https://doi.org/10.1145/3303772.3303793 (visited on 12/07/2022).

[47] Vincent Cho, T. C. Edwin Cheng, and W. M. Jennifer Lai. "The role of perceived user-interface design in continued usage intention of self-paced e-learning tools". en. In: *Computers & Education* 53.2 (Sept. 2009), pp. 216–227. ISSN: 0360-1315. DOI: 10.1016/j.compedu.2009.01.014. URL: https://www.sciencedirect.com/science/article/pii/S0360131509000207 (visited on 12/05/2022).

[48] Katerina Mangaroska et al. "Challenges and opportunities of multimodal data in human learning: The computer science students' perspective". en. In: *Journal of Computer Assisted Learning* 37.4 (2021), pp. 1030–1047. ISSN: 1365-2729. DOI: 10.1111/jcal.12542. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jcal.12542 (visited on 04/19/2023).

[49] Florence Martin et al. "Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018". en. In: *Educational Technology Research and Development* 68.4 (Aug. 2020), pp. 1903–1929. ISSN: 1556-6501. DOI: 10.1007/s11423-020-09793-2. URL: https://doi.org/10.1007/s11423-020-09793-2 (visited on 04/19/2023).

[50] John Brooke. "SUS: A quick and dirty usability scale". In: *Usability Eval. Ind.* 189 (Nov. 1995).

[51] Shurui Bai, Khe Foon Hew, and Biyun Huang. "Does gamification improve student learning outcome? Evidence from a meta-analysis and synthesis of qualitative data in educational contexts". en. In: *Educational Research Review* 30 (June 2020), p. 100322. ISSN: 1747-938X. DOI: 10.1016/j.edurev.2020.100322. URL: https://www.sciencedirect.com/science/article/pii/S1747938X19302908 (visited on 06/09/2023).

[52] Richard N. Landers and Michael B. Armstrong. "Enhancing instructional outcomes with gamification: An empirical test of the Technology-Enhanced Training Effectiveness Model". en. In: *Computers in Human Behavior* 71 (June 2017), pp. 499–507. ISSN: 0747-5632. DOI: 10.1016/j.chb.2015.07.031. URL: https://www.sciencedirect.com/science/article/pii/S074756321530042X (visited on 04/26/2023).

[53] Jenni Majuri, Jonna Koivisto, and Juho Hamari. "Gamification of Education and Learning: A Review of Empirical Literature." In: May 2018.

[54] Christopher Cheong, Justin Filippou, and France Cheong. "Towards the Gamification of Learning: Investigating Student Perceptions of Game Elements</title>". In: *Journal of Information Systems Education* 25.3 (Aug. 2014), p. 233. URL: https://jise.org/volume25/n3/JISEv25n3p233.html (visited on 12/06/2022).

[55] Michael Sailer and Lisa Homner. "The Gamification of Learning: a Meta-analysis". en. In: *Educational Psychology Review* 32.1 (Mar. 2020), pp. 77–112. ISSN: 1573-336X. DOI: 10.1007/s10648-019-09498-w. URL: https://doi.org/10.1007/s10648-019-09498-w (visited on 12/07/2022).

[56] Richard N. Landers, Kristina N. Bauer, and Rachel C. Callan. "Gamification of task performance with leaderboards: A goal setting experiment". en. In: *Computers in Human Behavior* 71 (June 2017), pp. 508–515. ISSN: 0747-5632. DOI: 10.1016/j.chb.2015.08.008. URL: https://www.sciencedirect.com/science/article/pii/S0747563215300868 (visited on 04/26/2023).

[57] Mauricio Ronny de Almeida Souza et al. "Gamification in Software Engineering Education: An Empirical Study". In: *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*. ISSN: 2377-570X. Nov. 2017, pp. 276–284. DOI: 10.1109/CSEET.2017.51.

[58] Michael B. Armstrong and Richard N. Landers. "An Evaluation of Gamified Training: Using Narrative to Improve Reactions and Learning - Michael B. Armstrong, Richard N. Landers, 2017". In: (). URL: https://journals.sagepub.com/doi/10.1177/1046878117703749 (visited on 04/26/2023).

[59] Bohyun Kim. "Chapter 5. Designing Gamification in the Right Way". en. In: *Library Technology Reports* 51.2 (Mar. 2015). Number: 2, pp. 29–35. ISSN: 0024-2586. URL: https://journals.ala.org/index.php/ltr/article/view/5632 (visited on 11/22/2022).

[60] Christo Dichev and Darina Dicheva. "Gamifying education: what is known, what is believed and what remains uncertain: a critical review". In: *International Journal of Educational Technology in Higher Education* 14.1 (Feb. 2017), p. 9. ISSN: 2365-9440. DOI: 10.1186/s41239-017-0042-5. URL: https://doi.org/10.1186/s41239-017-0042-5 (visited on 04/26/2023).

[61] Ruth Malan et al. "Functional Requirements and Use Cases". en. In: ().

[62] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley Professional, 2003.

[63] *Figma: the collaborative interface design tool*. URL: https://www.figma.com.

[64] *Craft – The Future of Documents*. URL: https://www.craft.do.

[65] *Miro: The Visual Collaboration Platform for Every Team*. URL: https://miro.com.

[66] *GitHub: Let's build from here*. en. URL: https://github.com/ (visited on 05/10/2023).

[67] *ADAPT2 - PAWS Lab*. URL: http://adapt2.sis.pitt.edu/wiki/ADAPT2 (visited on 06/13/2023).

[68] *Supabase*. en. URL: https://supabase.com/ (visited on 05/12/2023).

[69] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. Request for Comments RFC 8259. Num Pages: 16. Internet Engineering Task Force, Dec. 2017. DOI: 10.17487/RFC8259. URL: https://datatracker.ietf.org/doc/rfc8259 (visited on 05/12/2023).

[70] *really-relaxed-json*. en. Nov. 2022. URL: https://www.npmjs.com/package/really-relaxed-json (visited on 05/12/2023).

[71] *create-t3-stack Star History on Github*. URL: https://star-history.com/#t3-oss/create-t3-app&Date (visited on 05/13/2023).

[72] Mohit Thakkar. "Next.js". en. In: *Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications*. Ed. by Mohit Thakkar. Berkeley, CA: Apress, 2020, pp. 93–137. ISBN: 978-1-4842-5869-9. DOI: 10.1007/978-1-4842-5869-9_3. URL: https://doi.org/10.1007/978-1-4842-5869-9_3 (visited on 05/30/2023).

[73] Prisma Inc. Inc. *What is Prisma? (Overview)*. URL: https://www.prisma.io/docs (visited on 05/30/2023).

[74] trpc trpc trpc. *HTTP RPC Specification | tRPC*. en. URL: https://trpc.io/docs/rpc (visited on 03/29/2023).

[75] Colin McDonnell. *TypeScript-first schema validation with static type inference*. en. URL: https://github.com/colinhacks/zod (visited on 05/30/2023).

[76] *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. en. Nov. 2020. URL: https://tailwindcss.com/ (visited on 06/13/2023).

[77] Dan Abrimov. *Passing Data Deeply with Context – React*. en. URL: https://react.dev/learn/passing-data-deeply-with-context (visited on 03/29/2023).

[78] Dan Vanderkam. *Effective TypeScript: 62 Specific Ways to Improve Your TypeScript*. " O'Reilly Media, Inc.", 2019.

[79] Lori Kupczynski et al. "The Impact of Frequency on Achievement in Online Courses: A Study From a South Texas University". In: *Journal of Interactive Online Learning* 10.3 (2011). ISSN: 1541-4914. URL: http://www.ncolr.org/jiol.

[80] Andy Field. *Discovering statistics using IBM SPSS statistics*. sage, 2013.

[81] *UpSet: Visualization of Intersecting Sets*. en. URL: https://vdl.sci.utah.edu/publications/2014_infovis_upset/ (visited on 06/06/2023).

[82] Zachary Pardos, Frederic Wang, and Anirudhan Badrinath. *pyBKT Model Tutorial*. en. URL: https://colab.research.google.com/drive/13abu919edUXbvPV3qeGPpvwnFBExU7Vd#scrollTo=oWRr6FxErB6G (visited on 05/22/2023).

[83] *ASSISTments Data Set 2012-2013*. en. URL: https://www.kaggle.com/datasets/nicolaswattiez/skillbuilder-data-2009-2010 (visited on 05/02/2023).

[84] Karrie Godwin and Anna Fisher. “Selective sustained attention, the visual environment, and learning in kindergarten-age children: Preliminary results of an individual difference study”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 36. Issue: 36. 2014.

[85] Alexander Skulmowski and Kate Man Xu. “Understanding cognitive load in digital and online learning: A new perspective on extraneous cognitive load”. In: *Educational Psychology Review* 34 (2022). Place: Germany Publisher: Springer, pp. 171–196. ISSN: 1573-336X. DOI: 10.1007/s10648-021-09624-7.

[86] Gale M. Sinatra, Benjamin C. Heddy, and Doug Lombardi. “The Challenges of Defining and Measuring Student Engagement in Science”. In: *Educational Psychologist* 50.1 (Jan. 2015), pp. 1–13. ISSN: 0046-1520. DOI: 10.1080/00461520.2014.1002924. URL: https://doi.org/10.1080/00461520.2014.1002924 (visited on 06/07/2023).

[87] Lisa Legault and Michael Inzlicht. “Self-determination, self-regulation, and the brain: Autonomy improves performance by enhancing neuroaffective responsiveness to self-regulation failure.” en. In: *Journal of Personality and Social Psychology* 105.1 (July 2013), pp. 123–138. ISSN: 1939-1315, 0022-3514. DOI: 10.1037/a0030426. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/a0030426 (visited on 06/06/2023).

[88] Elena Karahanna and Detmar W Straub. “The psychological origins of perceived usefulness and ease-of-use”. en. In: *Information & Management* 35.4 (Apr. 1999), pp. 237–250. ISSN: 0378-7206. DOI: 10.1016/S0378-7206(98)00096-2. URL: https://www.sciencedirect.com/science/article/pii/S0378720698000962 (visited on 05/13/2023).

[89] Aaron Bangor and Philip T. Kortum. *Full article: An Empirical Evaluation of the System Usability Scale*. 2008. URL: https://www.tandfonline.com/doi/full/10.1080/10447310802205776 (visited on 06/08/2023).

APPENDICES

A Thesis Description

NTNU

1 of 4

1 av 4

fdsfsfsdfdsffsdsdfsdfsdffdsfsdfsffsdfsdf

Vår dato
Vår referanse

Kommunikasjonsavdelingen

Deres dato
Deres referanse

Online learning platform

Supervisor: Boban Vesin
Suitable for: One or two students

Introduction

ProTuS is a Programming tutoring system designed to provide learners with personalised courses from various domains. It offers a Java programming course with interactive third-party material (https://protus.idi.ntnu.no/, username: testUSN@usn.no, password: test).

The aim of the thesis is to develop a new version of the system, access content from third-party content providers (Java, Python and SQL courses) and create a framework for adding new courses. The system should also provide infrastructure for the collection and display of different learning analytics, logging learners' activities, and providing authentication and security.

Thesis Description

As a first step, the team should review the literature and familiarise themselves with the personalisation in e-learning. Then, the candidate will design and implement the web application that offers online programming courses via an API connection to third-party content providers[1], based on the best practices found and adapted from the literature. Afterwards, the candidate will conduct a user study to empirically test the proposed system and its features. Finally, the candidate will analyse the collected data and write the thesis.

Requirements

The ideal candidates will have a background in software design, solid programming skills and an interest in hands-on development and experimentation.

Programming skills: React, Firebase DB, Experience API https://xapi.com/ (optional),

Expected Project Work Packages

1. WP: Short literature study on personalisation in e-learning.
2. WP: Iteratively develop and test the system.
3. WP: Conduct a usability study of the system and finalise the development.
4. WP: Conduct a user study to test the effectiveness of the system.

[1] http://adapt2.sis.pitt.edu/aggregate2/GetContentLevels?grp=NorwayFall2020&sid=TEST&cid=352&mod=all

Postadresse	Org.nr. 974 767 880	Besøksadresse	Telefon	
7491 Trondheim	E-post:	Hovedbygningen	+ 47 73 59 55 40	
	info@adm.ntnu.no	Høgskoleringen 1	**Telefaks**	
	http://www.ntnu.no/adm/info	Gløshaugen	+ 47 73 59 54 37	Tlf: + 47 lkjlljljkljklkjljklj

5. WP: Writing the thesis.

Thesis grading scheme

Grade	Description of the evaluation criteria
A	The candidate demonstrates excellent judgement and a high degree of independent thinking. **Significantly exceeded expectations** with original contribution.
B	The candidate demonstrates sound judgement and a very good degree of independent thinking. A very good performance, **the candidate has exceeded expectations.**
C	A good performance in most areas. The candidate demonstrates a reasonable degree of judgement and independent thinking in the most important areas, **the expectations are met but not surpassed.**
D	A satisfactory performance, but with significant shortcomings. The candidate demonstrates a limited degree of judgement and independent thinking.
E	A performance that meets the minimum criteria, but no more. The candidate demonstrates a very limited degree of judgement and independent thinking.
F	A performance that does not meet the minimum academic criteria. The candidate demonstrates an absence of both judgement and independent thinking.

Requirements

The different stakeholders will have different requirements. The following list of stakeholders and their specific interests are suggested:

- A teacher is interested in different learning analytics collected and visualized by the system.
- A student should have an overview and access to learning content from third-party content providers. Optionally, students can receive reports about their interactions and progress.
- Admin should be able to add links to new content providers.

The functional requirements are as follows (ordered by the priority for implementation):

- Same web user interface for all main categories of users (stakeholders): teacher, student and admin.
- Authentication of users (log-in and registration for students) with access control (visibility of specific parts of UI to different categories of users)
- For students:
  - overview and access to available (third-party) courses and learning content via APIs
  - learning analytics about their progress, results, and visited/solved content.
  - customization of the course options, UI
- For teachers:
  - an overview of learning content, courses and learning analytics (progress and results of students)
  - customization of the courses and teaching process
- For admin: adding new APIs information and administration of other users (overview and modification).
- Communication options (exchange of messages among different users)
- Personalisation options:
  - Recommendation of the courses and content
  - Adaptation of the content
  - Automatic evaluation and grading

The non-functional requirements:

- To enable rich integration with other learning platforms and tools, the app must use LTI (Learning Tools Interoperability) protocols
  - https://www.overtsoftware.com/all-about-learning-tools-interoperability/
  - https://www.imsglobal.org/activity/learning-tools-interoperability
- Data collection and tracking using LTI protocols

Learning content is accessed via APIs (example for Java course):
http://adapt2.sis.pitt.edu/aggregate2/GetContentLevels?grp=NorwayFall2020&sid=TEST&cid=352&mod=all

Report about the interaction of the learners:

http://adapt2.sis.pitt.edu/aggregate2/GetContentLevels?usr=demo&grp=ADL&mod=user&sid=TEST&cid=352&lastActivityId=while_loops.j_digits&res=-1

Third-party content:
https://canvas.instructure.com/courses/2062633

B Github repository link

The source code of Progresso is included in the Github repository linked below.

- `https://github.com/mathisfo/bug-free-adventure`

C Information Letter to Participants

Invitation to taking part in the research project Online Learning Platform.

Purpose of the project

You are invited to participate in a research project where the main purpose is to track and investigate students' use of an online learning platform.

Which institution is responsible for the research project?

The Department of Computer Science at NTNU is responsible for the project.

Why are you being asked to participate?

You are being asked to participate in this study since you are taking Object-Oriented Programming at USN.

What does participation involve for you?

Participating in this study involves the use of an online learning platform over the course of 6-8 weeks. After the test period you might be sent an online questionnaire. The questionnaire is tied to questions related to your experience of the dashboard. The questionnaire will be fulfilled by a third-party provider your university has a data processing agreement with.

Participation is voluntary

By participating in this research project you consent to that we temporarily store your name and USN email. Participation is voluntary. If you would like to withdraw please inform Boban Vesin. Sensitive information collected thus far will be anonymized or deleted at your request.

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

We will only use your personal data for the purpose specified here and we will process your personal data in accordance with data protection legislation (GDPR).

In the final report the participants will be completely anonymous.

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

The planned end date of the project is 20.06.23. Your name and USN email will be deleted afterwards. Your data collected during the test phase will not be uniquely identifiable afterwards.

Your rights

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

- acces the personal data that is bein 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,
- 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?

During the test phase we need your USN email to reach you if any thing goes wrong with the dashboard.

Where can I fint out more?

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

- The Department of Computer Science via Boban Vesin (boban.vesin@ntnu.no).
- The students: Johanne Tronstad (johatr@stud.ntnu.no) / Mathias Fossum (mathisfo@stud.ntnu.no)

Yours sincererely,

Boban Vesin (project supervisor) Johanne Tronstad, Mathias Fossum (students)

D Tasks given to Participants

Task 1

Due date 20th of March - 23.59

Dashboard
Complete your assignments here
https://www.progressoapp.no/

Introduction

This exercise is mainly to let you try out the dashboard and get an idea on how to navigate around. First, you have to log in to the dashboard by using your GitHub account. If you don't have a GitHub account yet you can create one here https://github.com/join. After signing up to the dashboard you'll receive a mail on the email you signed up with on Github with instructions on how to start task 1.1.

1.1

Complete the onboarding process. Select the data visualization components you would like in your dashboard based on your personal preference.

1.1.1 Optional

If you selected the Exercise Planner component during the onboarding process in 1.1 you can make yourself a plan on when and which exercises you want to do.

1.2

Complete at least 10 example exercises from the any of the topics.

Complete at least 5 challenges from any of the topics.

1.3

Complete at least 3 coding exercises from any of the topics. It is up to you which 3 you pick, so maybe go with something you know you need to improve upon.

Task 2

Due date 27th of March - 23.59

Dashboard
Complete your assignments here
https://www.progressoapp.no/

2.1

Select a topic that you know you need to improve upon.

Complete at least 3 examples, at least 3 challenges and at least 2 coding exercises on the topic you have selected.

Task 3

Due date 3th of April - 23.59

Dashboard
Complete your assignments here
https://www.progressoapp.no/

3.1

For the last task you must do at least five new coding exercises across any of the topics. This is the last task, and should prep you for your upcoming exam. If you have decided to participate in the leaderboard, maybe now is the time to achieve your spot!

E Questionnaire

Progresso Questionnaire

* Obligatorisk

Personalia

Your personalia will only be used to categorize the answers.

1

USN email *

2

Gender *

- Female
- Male
- Non-binary
- Prefer not to say

3

Age *

Tallet må være mellom 18 og 50

4

Attended course *

- OBJ2000
- OBJ2100

System Usability Scale

The System Usability Scale (SUS) is a standardized metric used to measure usability perception of computer interfaces. SUS consists of 10 questions, and the response system for each question is a 5-point Likert agreement scale.

5

*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I think that I would like to use Progresso frequently.	○	○	○	○	○
I found Progresso unnecessarily complex.	○	○	○	○	○
I thought Progresso was easy to use.	○	○	○	○	○
I think that I would need assistance to be able to use Progresso.	○	○	○	○	○
I found the various functions in Progresso were well	○	○	○	○	○

integrated.

I thought there was too much inconsistency in Progresso.	○	○	○	○	○
I would imagine that most people would learn to use Progresso very quickly.	○	○	○	○	○
I found Progresso very cumbersome/awkward to use.	○	○	○	○	○
I felt very confident using Progresso.	○	○	○	○	○
I needed to learn a lot of things before I could get going with Progresso.	○	○	○	○	○

Engagement and Motivation

Select your opinion on the statements below about the dashboard

6

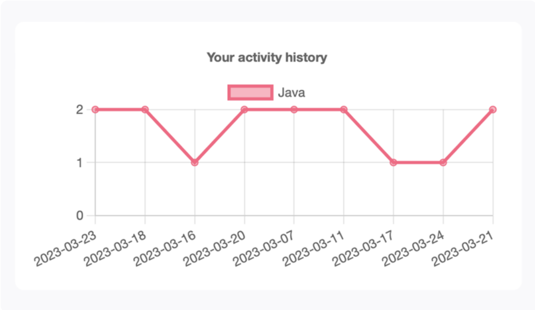
Select your opinion on the statements below *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I liked that I could customize my dashboard	◯	◯	◯	◯	◯
The customizable dashboard made feel motivated	◯	◯	◯	◯	◯
I found it easy to get an overview of my progression	◯	◯	◯	◯	◯

Exercise Graph component

7

Did you select the Exercise Graph? *



- Yes
- No
- Initially yes, finally no
- Initially no, finally yes

8

Select your opinion on the Exercise Graph statements *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I frequently used the exercise graph	○	○	○	○	○
The exercise graph made me feel more engaged	○	○	○	○	○
The exercise graph gave me a better overview of my progression	○	○	○	○	○
I found it easy to extract relevant information from the Exercise Graph	○	○	○	○	○
The exercise graph made me feel motivated	○	○	○	○	○

9

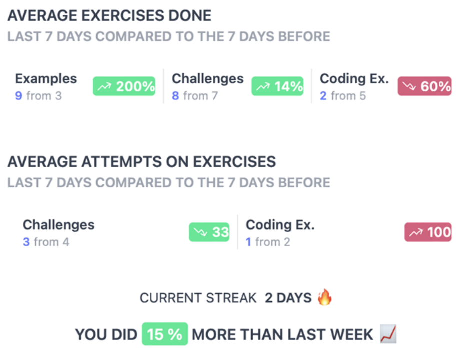
Rate the Exercise Graph component based on usefulness *

☆ ☆ ☆ ☆ ☆

Stats component

10

Did you select Stats? *



- ◯ Yes
- ◯ No
- ◯ Initially yes, finally no
- ◯ Initially no, finally yes

11

Select your opinion on the Stats statements *

	Stongly disagree	Disagree	Neutral	Agree	Strongly agree
I frequently used the stats component	○	○	○	○	○
The stats made me feel more engaged	○	○	○	○	○
The stats gave me a better overview of my progression	○	○	○	○	○
I found it easy to extract relevant information from the stats component	○	○	○	○	○
The stats component made me feel motivated	○	○	○	○	○

12

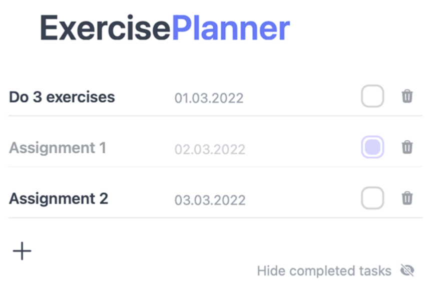
Rate the Stats component based on usefulness *

☆ ☆ ☆ ☆ ☆

Exercise Planner component

13

Did you select ExercisePlanner? *



- ◯ Yes
- ◯ No
- ◯ Initially yes, finally no
- ◯ Initially no, finally yes

14

Select your opinion on the ExercisePlanner statements *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I frequently used the exercise planner	○	○	○	○	○
The exercise planner made me feel motivated	○	○	○	○	○
The exercise planner made me feel more engaged	○	○	○	○	○
The exercise planner gave me a better overview of my progression	○	○	○	○	○
I found it easy to extract relevant information from the exercise planner	○	○	○	○	○

15

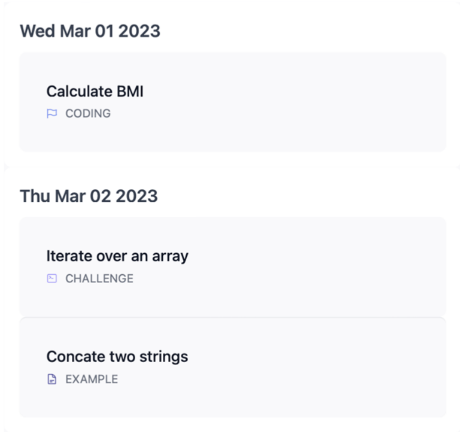
Rate the Exercise Planner component based on usefulness *

☆ ☆ ☆ ☆ ☆

Activity History component

16

Did you select Activity History? *



- Yes
- No
- Initially yes, finally no
- Initially no, finally yes

17

Select your opinion on the Activity History statements *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I frequently used activity history	◯	◯	◯	◯	◯
The activity history made me feel more engaged	◯	◯	◯	◯	◯
The activity history gave me a better overview of my progression	◯	◯	◯	◯	◯
I found it easy to extract relevant information from the exercise planner	◯	◯	◯	◯	◯
The activity history made me feel motivated	◯	◯	◯	◯	◯

18

Rate the Activity History component based on usefulness *

☆ ☆ ☆ ☆ ☆

Leaderboard component

19

Did you participate in the leaderboard? *

LEADERBOARD

JAVA

RANKING	NAME	SCORE
1	User A	182
2	User B	163
3	User C	150
4	User D	144
5	User E	134
6	User F	123
7	User G	120
8	User H	65
9	User I	57
10	User J	56
23	Me	4

- Yes
- No
- Initially yes, finally no
- Initially no, finally yes

20

Select your opinion on the Leaderboard statements *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I joined the leaderboard because of social recognition	◯	◯	◯	◯	◯
I joined the leaderboard to see how my classmates were doing	◯	◯	◯	◯	◯
I joined the leaderboard to compete with others	◯	◯	◯	◯	◯
The leaderboard made me feel motivated	◯	◯	◯	◯	◯
The leaderboard made me feel more engaged	◯	◯	◯	◯	◯
The leaderboard gave me a better overview of my progression	◯	◯	◯	◯	◯

21

Rate the Leaderboard component based on usefulness *

☆ ☆ ☆ ☆ ☆

Order

22

Rank which order you prefered when selecting your exercises *

- Exercise type (example, challenge, coding exercise)
- The exercise's topic (ie. "Variables and Operations")
- If the exercise was recommended by the system
- An exercise I needed to improve upon
- Exercises I found easiest

23

Order the type of exercise you selected from the most down to the least *

- Challenge
- Example
- Coding exercise

Design

24

Select your opinion on the statements below *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The design and colors made me feel happy	○	○	○	○	○
The design and colors made me feel motivated	○	○	○	○	○
The onboarding process was easy to complete	○	○	○	○	○

Other

25

Other comments about Progresso

E.1 Questionnaire Feedback

ID ↑	Navn	Svar
1	anonymous	I think it would be nice to be able to get tips/answer in the coding exercises, after you fail the exercise some amount of times.
2	anonymous	I find it non-productive towards learning; it's too much text (most people would like a video explanation instead) and it's in English. Most people have a different mother tongue making it more difficult for them to understand the tasks. Also, there are bugs where one must submit the same answer several times before it accepts it as correct, and most of my progress was lost and I had to start over again.
3	anonymous	It was cumbersome that the excersises were in a different tab than the application. It would have been nice if everything would be on the same page
4	anonymous	I felt like the optional components were too similar to each other.
5	anonymous	I had a persistent problem where upon submitting code I would be told that I had to reauthenticate and reload the page. Not a big problem, as I was able to copy my code and paste it back in after reloading but still a bit annoying.
6	anonymous	There could be more or larger coding exercises. Some topics only had 1 or 0
7	anonymous	Leaderboard was not tracking my progress like it should. I had done everything, but leaderboard was way behind.
8	anonymous	Great system that I will use even after finishing my degree, to practice my Java developing skills. Only thing I have to complain about is the loading time on some of the coding exercises. Made me impatient and resultet in me skipping the exercises with long loading time. I would also wish for more hint in the coding exercises, maybe in the same way as the coding challanges, to increase my motivation when I am dead stuck on a task. Frustrating to be stuck and not have a clue about what to do in order to complete an exercise. But overall, great system that definetly motivates!
9	anonymous	I wish there were more tasks on writing an entire program. starting from a small program to a bigger one. That is a program from a-z because we get trained on coding exercises, but in addition to this I would appreciate entire programs with feedback from the system like the coding exercises.
10	anonymous	It was an easy to use interface, and i felt like it had all the right components to make it pleasant for the user.
11	anonymous	I would like to be able to see my answer to the questions especially the coding, because that would be helpful in other assignments. I did not like the fact that as soon as I exited the question my answer was gone
12	anonymous	The "CONTINUE WHERE YOU LEFT OF" box, made me think I had some unfinished task beacuse of the yellow colour.
13	anonymous	Really nice program
14	anonymous	First of all .. my age is above 50 :(and the compiler .. shity output does it exists better, yes https://www.hackerrank.com/domains/java
15	anonymous	I wish it saved my code/soultion for later. Would be nice to be able to go back and see how I solved it. It would also be helpful to get some kind of feedback when you don't get it right in the coding exercises, maybe some tips or something? I did enjoy using it!
16	anonymous	the coding sheet took time to understand

F Data models

The data models received from the adapt2 API. These models are an aggregation from the object structure received from the API. The original object structure did not fit the application's needs

F.1 activitySchema

Table F.1: Activity schema

Key	Type	Description
`relatedmodule`	string	
`activityName`	string	
`activityId`	string	
`url`	string	
`visited`	boolean	
`attempts`	number	
`successRate`	number	
`t`	number	
`aSeq`	string	
`sequencing`	number	
`type`	enum	`type.EXAMPLE`, `type.CODING`, or `type.CHALLENGE`

F.2 moduleProgressSchema

Table F.2: moduleProgressSchema

Key	Type	Description
`examples`	number	The number of completed examples for the module.
`challenges`	number	The number of completed challenges for the module.
`coding`	number	The number of completed coding exercises for the module.

F.3 activityAnalyticsSchema

Key	Type	Description
`examples`	Array<activitySchema>	Array of activities of type.EXAMPLE.
`challenges`	Array<activitySchema>	Array of activites of type.CHALLENGE
`coding`	Array<activitySchema>	Array of activities of type.CODING

F.4 moduleAnalyticsSchema

Key	**Type**	**Description**
`name`	string	
`description`	string	
`progress`	moduleProgressSchema	
`sequencing`	Object	{example, challenge, coding}

F.5 learnerActivitySchema

Key	**Type**	**Description**
`learner`	object	Contains id and name.
`moduleAnalytics`	Array<moduleAnalyticsSchema>	Analytics array for each module.
`activityAnalytics`	Array<activityAnalyticsSchema>	Course activities analytics.

G Implementation

G.1 Feedback 2nd Iteration

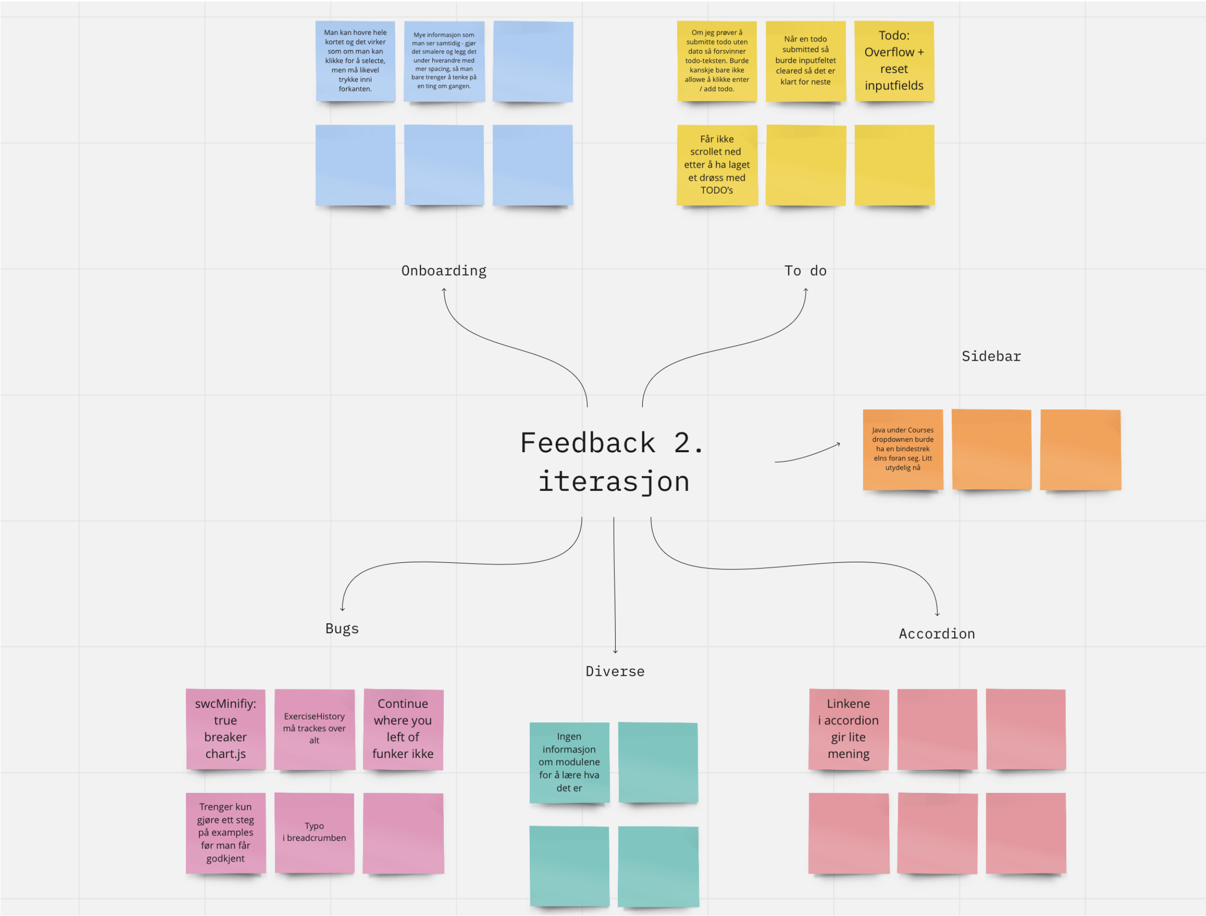
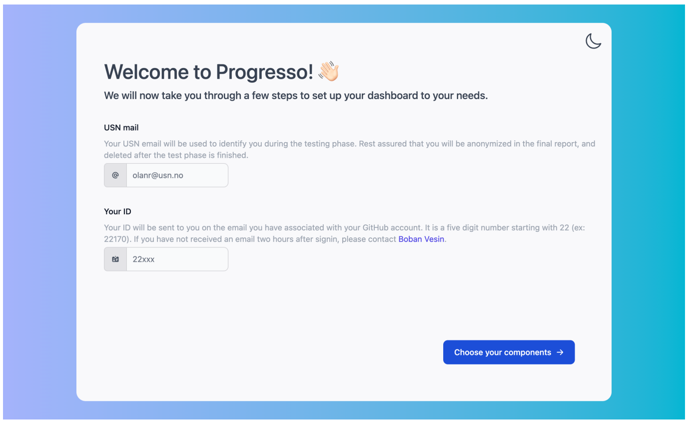
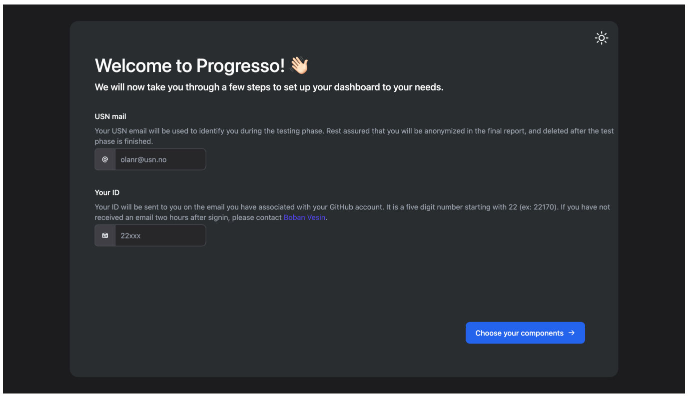


Figure G.1: Feedback from the 2nd iteration user test

H Final User Interface

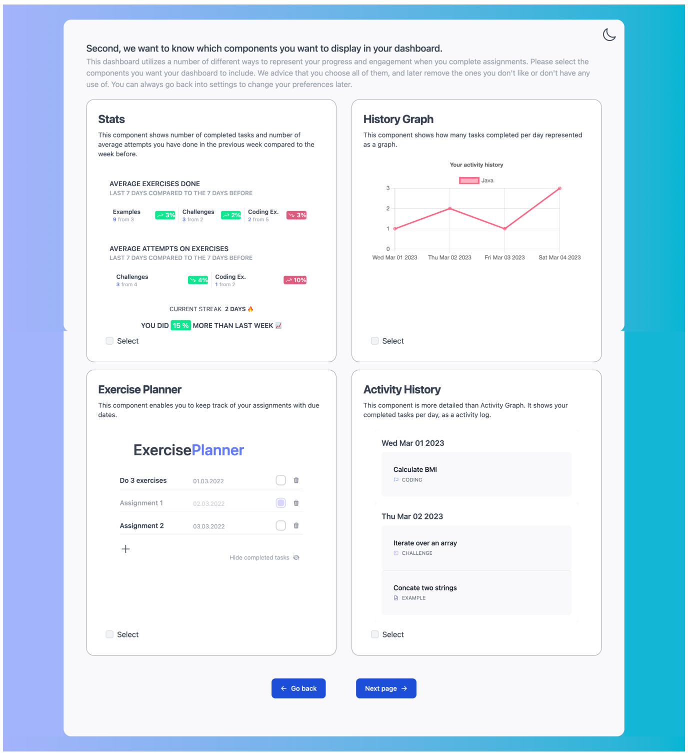


(a) Light mode

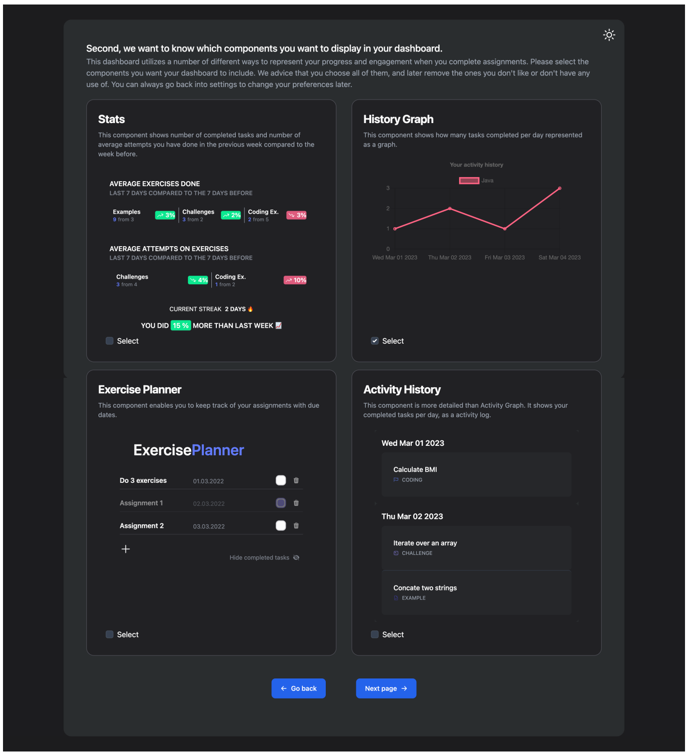


(b) Dark mode

Figure H.1: Fullsize screenshot of first on-boarding page

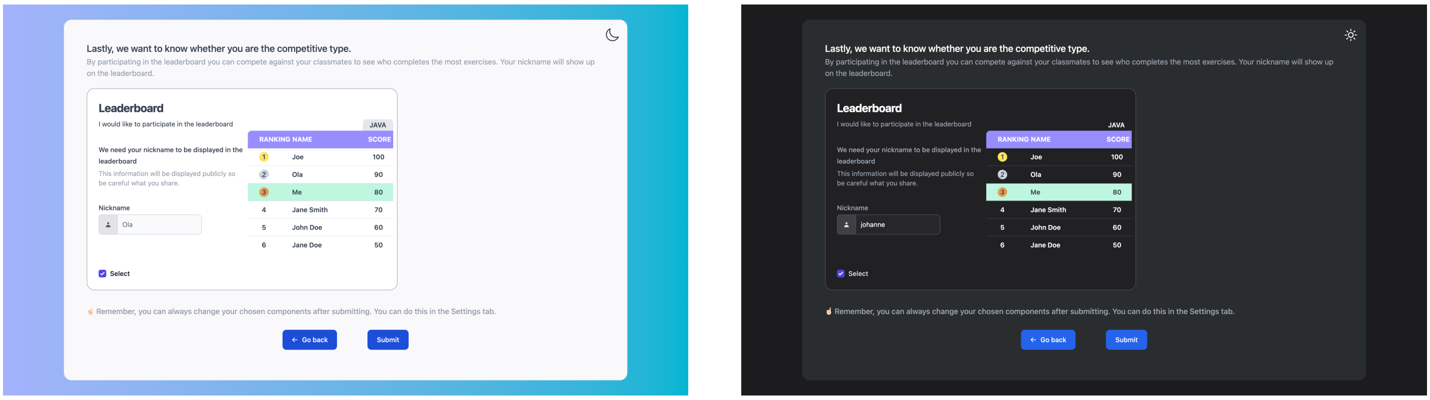


(a) Light mode



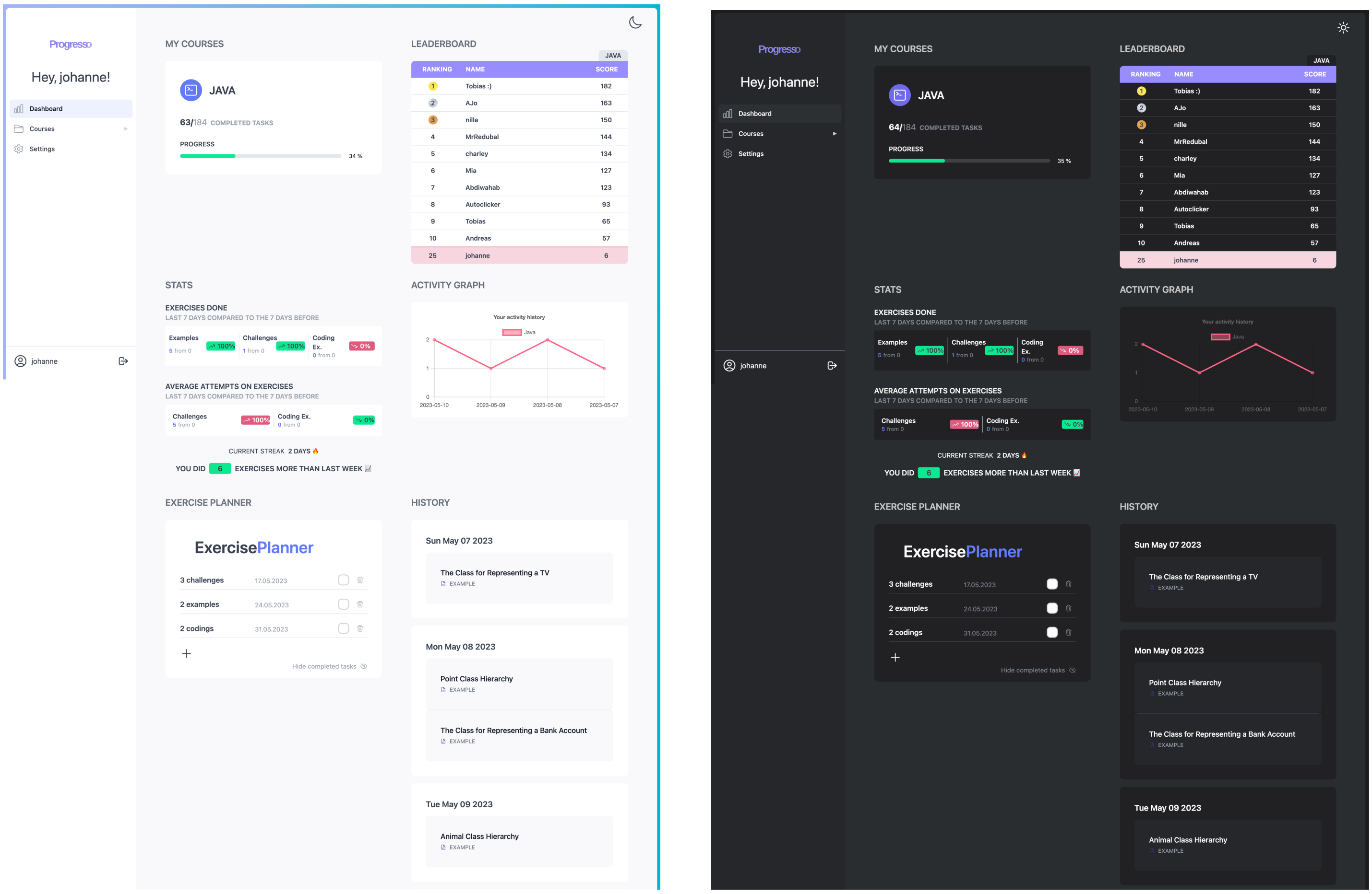
(b) Dark mode

Figure H.2: Fullsize screenshot of second on-boarding page



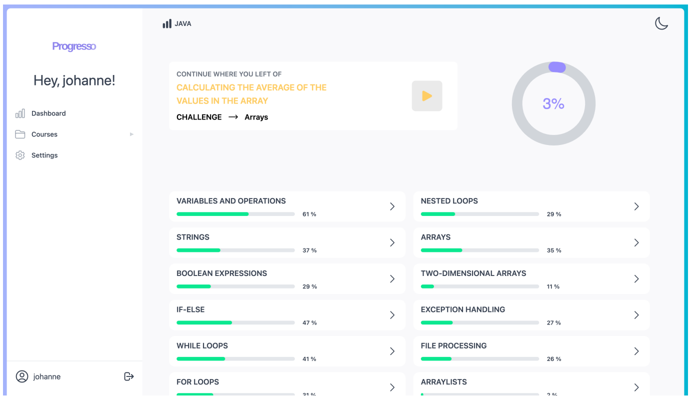
(a) Light mode (b) Dark mode

Figure H.3: Fullsize screenshot of third on-boarding page

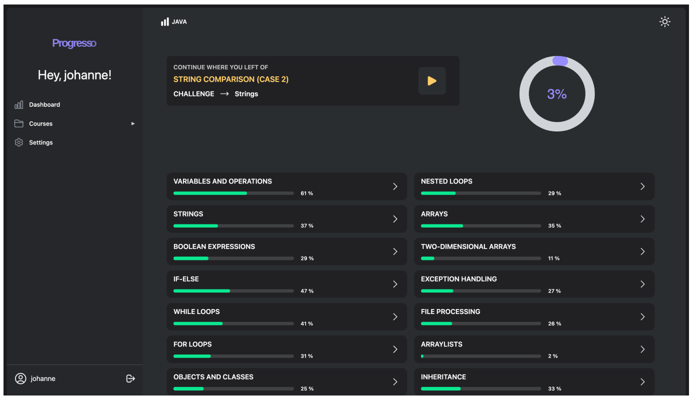


(a) Light mode (b) Dark mode

Figure H.4: Fullsize screenshot of dashboard

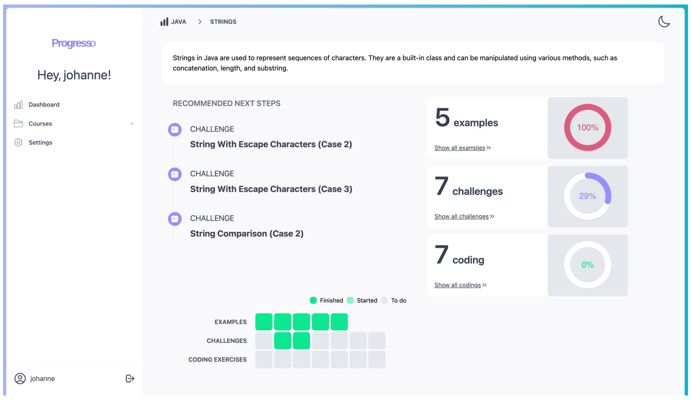


(a) Light mode

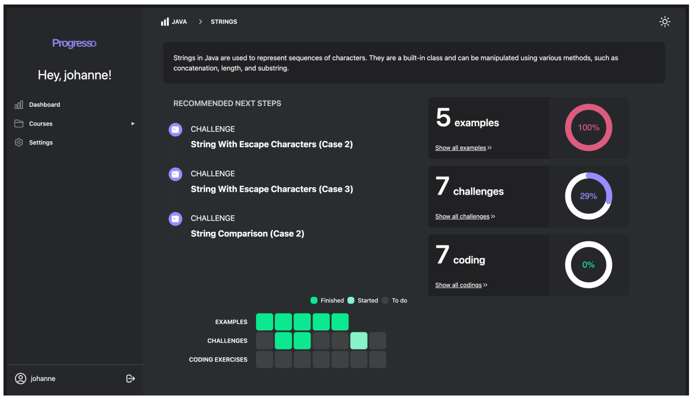


(b) Dark mode

Figure H.5: Fullsize screenshot of course page

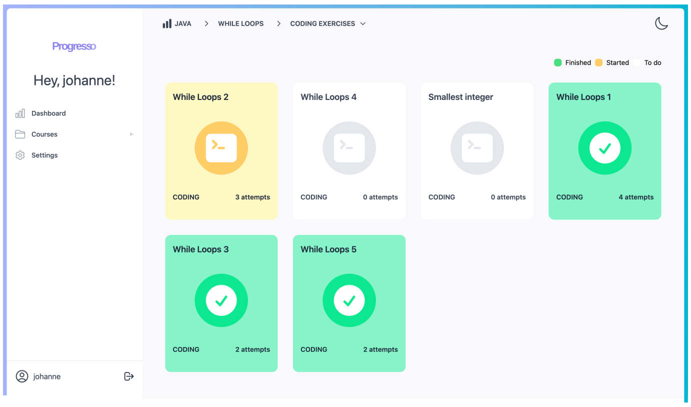


(a) Light mode

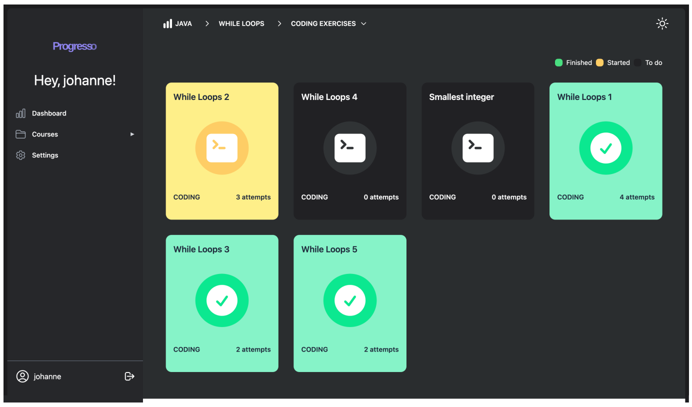


(b) Dark mode

Figure H.6: Fullsize screenshot of module page

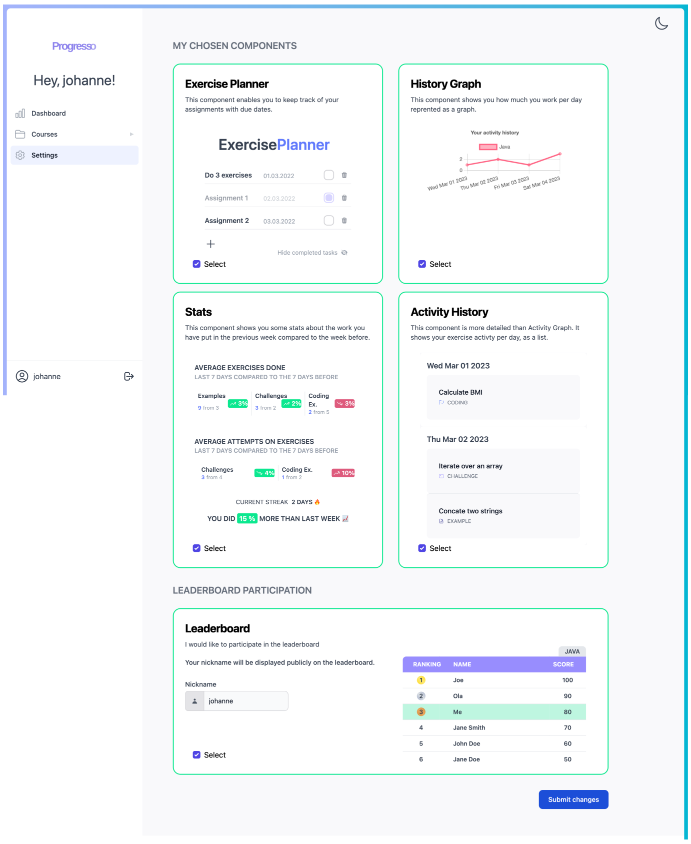


(a) Light mode

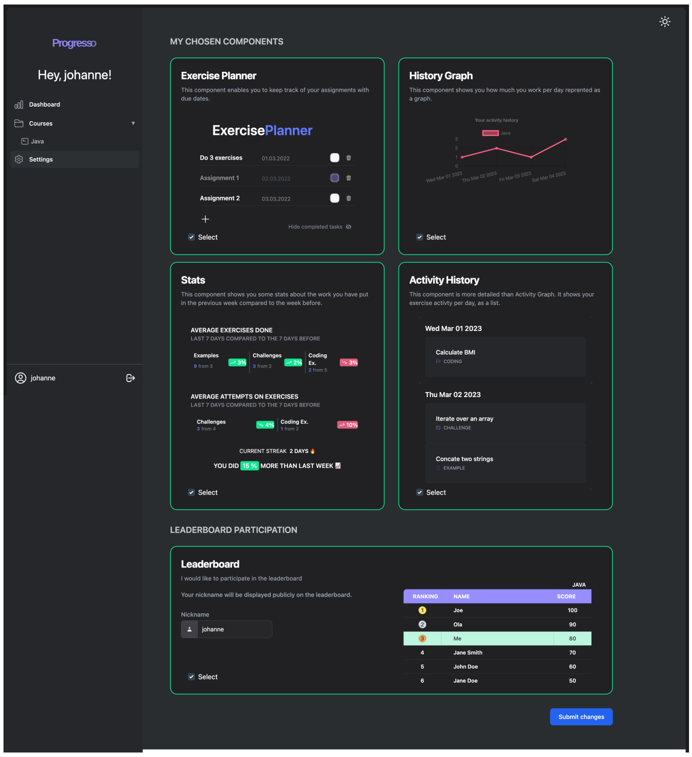


(b) Dark mode

Figure H.7: Fullsize screenshot of exercise page

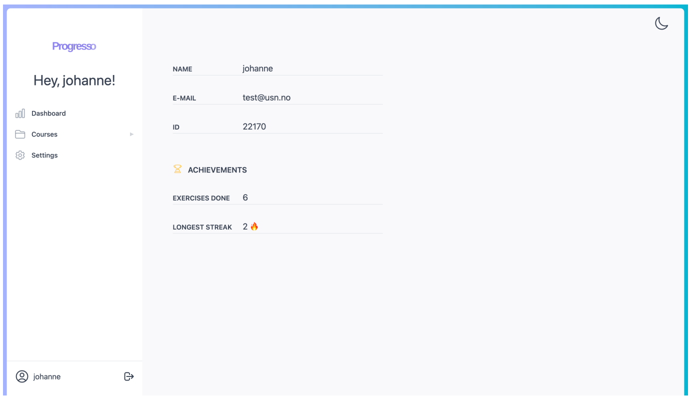


(a) Light mode

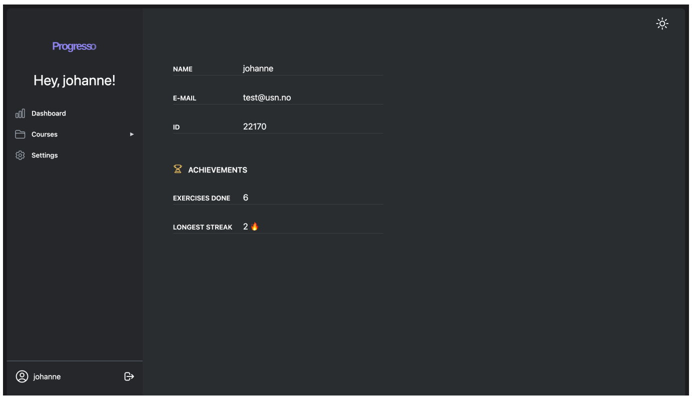


(b) Dark mode

Figure H.8: Fullsize screenshot of settings page



(a) Light mode



(b) Dark mode

Figure H.9: Fullsize screenshot of profile page

I Challenges dataset

user_id	avg_attempts	Leaderboard	avg_duration	avg_success_rate	AVERAGE of state_predictions	class	timesConfigChanged	EXERCISEHISTORY	HISTORYGRAPI	STATS	TODO
norway22101	4.33	TRUE	59.05	0.336	0.280	OBJ2000	1	1	1	1	1
norway22102	2.92	FALSE	237.18	0.532	0.390	OBJ2000	1	1	1	1	1
norway22104	1.63	FALSE	268.81	0.795	0.750	OBJ2100	1	0	0	0	0
norway22105	3.36	TRUE	105.53	0.374	0.180	OBJ2000	1	0	0	0	0
norway22108	2.38	TRUE	90.88	0.743	0.920	OBJ2000	3	0	0	1	0
norway22109	3.75	FALSE	76.16	0.274	0.280	OBJ2000	1	1	0	1	1
norway22112	4.50	FALSE	255.27	0.225	0.080	OBJ2000	3	0	0	0	0
norway22114	2.44	FALSE	182.38	0.641	0.674	OBJ2000	1	1	1	1	1
norway22116	5.25	TRUE	74.88	0.225	0.460	OBJ2000	2	1	1	1	1
norway22117	2.55	TRUE	119.27	0.585	0.667	OBJ2000	1	1	1	0	1
norway22118	3.00	TRUE	44.42	0.559	0.608	OBJ2100	10	0	0	0	0
norway22119	2.50	TRUE	257.90	0.705	0.805	OBJ2000	1	1	0	1	1
norway22121	2.75	FALSE	239.39	0.519	0.413	OBJ2100	5	0	0	1	0
norway22125	4.81	FALSE	51.36	0.274	0.328	OBJ2100	1	0	0	1	0
norway22127	1.40	FALSE	133.89	0.859	0.765	OBJ2100	1	1	0	0	1
norway22128	3.00	FALSE	113.36	0.465	0.385	OBJ2100	3	0	0	0	0
norway22132	2.80	FALSE	164.93	0.382	0.060	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22133	2.76	FALSE	72.76	0.491	0.260	OBJ2100	1	0	0	0	0
norway22137	2.83	FALSE	296.92	0.498	0.443	OBJ2100	1	1	1	1	1
norway22138	4.17	TRUE	191.54	0.273	0.180	OBJ2100	1	0	0	0	0
norway22139	4.70	FALSE	48.51	0.286	0.185	OBJ2100	8	1	1	1	0
norway22148	3.42	FALSE	86.46	0.353	0.435	OBJ2000	1	1	1	1	0
norway22149	3.15	FALSE	103.27	0.415	0.310	OBJ2100	1	0	0	0	0
norway22151	4.27	FALSE	28.23	0.330	0.350	OBJ2100	15	1	1	1	1
norway22153	4.17	TRUE	46.27	0.363	0.511	OBJ2000	2	1	1	1	0
norway22155	2.21	FALSE	68.09	0.673	0.603	OBJ2000	1	1	0	1	1
norway22156	2.20	TRUE	136.56	0.599	0.535	OBJ2100	10	0	0	0	0
norway22159	1.90	FALSE	92.49	0.691	0.620	OBJ2100	1	1	0	0	0
norway22160	2.38	FALSE	85.89	0.504	0.330	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22165	2.09	TRUE	185.75	0.666	0.747	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22175	3.45	FALSE	93.32	0.416	0.350	OBJ2000	1	0	0	0	0
norway22179	4.25	FALSE	39.36	0.361	0.333	OBJ2000	1	0	0	0	1
norway22182	2.00	TRUE	81.50	0.755	0.501	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22113	3.90	TRUE	23.56	0.377	0.390	OBJ2100	1	0	0	0	1
norway22134	1.21	TRUE	140.65	0.893	0.670	OBJ2100	1	1	1	1	1
norway22135	2.97	TRUE	70.89	0.523	0.541	OBJ2000	3	0	0	1	1
norway22136	1.75	FALSE	167.16	0.816	0.860	OBJ2100	1	0	0	0	1
norway22150	2.48	TRUE	144.00	0.634	0.569	OBJ2100	1	0	1	1	0
norway22158	3.50	FALSE	158.13	0.408	0.187	OBJ2100	1	0	0	0	0
norway22161	3.58	FALSE	295.79	0.383	0.230	OBJ2100	1	0	0	1	0
norway22166	2.06	FALSE	142.30	0.662	0.576	OBJ2000	1	1	0	1	1
norway22168	1.55	FALSE	115.35	0.757	0.600	OBJ2000	3	0	0	0	1
norway22107	1.53	FALSE	96.26	0.817	0.917	OBJ2100	1	0	0	0	0
norway22144	3.27	TRUE	63.25	0.522	0.503	OBJ2100	2	1	1	1	1
norway22145	1.87	TRUE	58.05	0.696	0.789	OBJ2100	1	1	1	0	1
norway22147	2.56	FALSE	104.42	0.502	0.373	OBJ2100	1	1	0	1	1
norway22154	2.44	FALSE	184.24	0.573	0.555	OBJ2100	1	0	0	0	0
norway22110	5.00	FALSE	333.51	0.420	0.280	OBJ2100	2	0	0	1	0
norway22130	2.09	TRUE	136.12	0.683	0.642	OBJ2100	2	0	0	0	0
norway22167	3.67	FALSE	129.47	0.394	0.357	OBJ2100	3	1	0	1	1
norway22126	2.23	FALSE	130.00	0.586	0.487	OBJ2100	1	1	1	1	1
norway22157	2.45	TRUE	66.65	0.609	0.627	OBJ2000	2	0	0	1	1
norway22172	1.38	FALSE	249.51	0.813	0.540	OBJ2100	1	1	1	1	1
norway22181	4.11	FALSE	68.62	0.447	0.375	OBJ2100	1	1	0	0	1
norway22120	2.50	FALSE	87.69	0.561	0.450	OBJ2100	5	1	0	1	0
norway22140	4.28	FALSE	37.85	0.313	0.322	OBJ2100	1	1	1	1	0
norway22176	4.14	TRUE	33.69	0.357	0.180	OBJ2100	1	0	1	1	1
norway22115	3.99	FALSE	30.29	0.365	0.392	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22141	1.93	TRUE	183.78	0.695	0.436	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22142	1.40	FALSE	155.91	0.800	0.815	OBJ2100	3	0	0	0	0
norway22162	3.13	FALSE	66.20	0.358	0.203	OBJ2000	1	0	0	1	0
norway22178	4.02	FALSE	19.29	0.346	0.459	OBJ2100	1	0	0	0	0
norway22103	3.25	FALSE	186.92	0.484	0.500	OBJ2100	1	0	0	1	1
norway22152	3.20	TRUE	132.89	0.502	0.548	OBJ2100	1	0	0	0	0
norway22173	3.50	FALSE	59.39	0.409	0.452	OBJ2100	1	0	0	1	1
norway22131	4.33	TRUE	81.67	0.357	0.640	OBJ2100	1	1	1	1	1

J Coding dataset

user_id	avg_attempts	Leaderboard	avg_duration	avg success_rate	AVERAGE of state_predictions	class	timesConfigChar	EXERCISEHIST(	HISTORYGRAPH	STATS	TODO
norway22101	3.07	TRUE	214.21	0.336	0.257	OBJ2000	1	1	1	1	1
norway22102	2.00	FALSE	#DIV/0!	0.000	0.060	OBJ2000	1	1	1	1	1
norway22104	3.25	FALSE	382.91	0.398	0.185	OBJ2000	1	0	0	0	0
norway22105	2.56	FALSE	131.53	0.467	0.325	OBJ2000	1	0	0	0	0
norway22108	5.42	TRUE	135.48	0.236	0.122	OBJ2000	3	0	0	1	0
norway22109	2.89	FALSE	72.41	0.370	0.369	OBJ2000	1	1	0	1	1
norway22112	4.00	TRUE	#DIV/0!	0.000	0.030	OBJ2000	3	0	0	0	0
norway22114	2.40	FALSE	268.04	0.566	0.449	OBJ2000	1	1	1	1	1
norway22116	3.56	TRUE	68.45	0.496	0.600	OBJ2000	2	1	1	1	1
norway22117	3.56	FALSE	629.08	0.406	0.256	OBJ2000	1	1	1	0	1
norway22118	2.67	TRUE	136.17	0.644	0.535	OBJ2000	10	0	0	0	0
norway22119	4.60	FALSE	198.50	0.360	0.477	OBJ2000	1	1	0	1	1
norway22121	3.55	FALSE	114.65	0.140	0.472	OBJ2000	5	0	0	1	0
norway22125	1.60	FALSE	311.97	0.689	0.418	OBJ2000	1	0	0	1	0
norway22127	2.43	TRUE	409.11	0.655	0.495	OBJ2000	1	1	0	0	1
norway22128	2.21	TRUE	465.98	0.624	0.393	OBJ2000	3	0	0	0	0
norway22132	1.50	TRUE	134.44	0.208	0.010	OBJ2000	#N/A	#N/A	#N/A	#N/A	#N/A
norway22133	2.20	FALSE	434.66	0.596	0.629	OBJ2000	1	0	0	0	0
norway22137	5.75	FALSE	1061.51	0.280	0.110	OBJ2000	1	1	1	1	1
norway22138	3.43	FALSE	546.53	0.230	0.030	OBJ2000	1	0	0	0	0
norway22139	2.59	FALSE	144.82	0.609	0.600	OBJ2000	8	1	1	1	0
norway22148	1.38	FALSE	89.39	0.906	0.767	OBJ2000	1	1	1	1	0
norway22149	2.08	FALSE	113.64	0.625	0.563	OBJ2100	1	0	0	0	0
norway22151	1.67	TRUE	67.03	0.653	0.698	OBJ2100	15	1	1	1	1
norway22153	2.43	TRUE	194.26	0.641	0.850	OBJ2100	2	1	1	1	0
norway22155	3.31	FALSE	569.14	0.548	0.097	OBJ2100	1	1	0	1	1
norway22156	1.72	FALSE	166.58	0.749	0.786	OBJ2100	10	0	0	0	0
norway22159	3.53	TRUE	241.21	0.392	0.364	OBJ2100	1	1	0	0	0
norway22160	2.61	TRUE	146.60	0.429	0.274	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22165	3.54	FALSE	443.34	0.472	0.405	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22175	3.20	FALSE	99.33	0.500	0.240	OBJ2100	1	0	0	0	0
norway22179	2.50	TRUE	156.84	0.606	0.571	OBJ2100	1	0	0	0	1
norway22182	3.64	FALSE	765.19	0.533	0.534	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22113	3.39	FALSE	203.58	0.531	0.457	OBJ2100	1	0	0	0	1
norway22134	1.74	TRUE	114.75	0.861	0.877	OBJ2100	1	1	1	1	1
norway22135	2.76	FALSE	232.51	0.677	0.518	OBJ2100	3	0	0	1	1
norway22136	2.67	TRUE	269.77	0.716	0.682	OBJ2100	1	0	0	0	1
norway22150	3.75	TRUE	623.11	0.557	0.337	OBJ2100	1	0	1	1	0
norway22158	4.14	FALSE	558.26	0.461	0.311	OBJ2100	1	0	0	0	0
norway22161	6.20	FALSE	684.71	0.226	0.350	OBJ2100	1	0	0	1	0
norway22166	3.17	FALSE	225.82	0.413	0.448	OBJ2100	1	1	0	1	1
norway22168	3.55	TRUE	738.62	0.578	0.660	OBJ2100	3	0	0	0	1
norway22107	2.73	FALSE	429.18	0.581	0.370	OBJ2100	1	0	0	0	0
norway22144	3.08	TRUE	340.10	0.527	0.323	OBJ2100	2	1	1	1	1
norway22145	2.41	FALSE	125.94	0.670	0.623	OBJ2100	1	1	1	0	1
norway22147	2.73	FALSE	1191.34	0.661	0.586	OBJ2100	1	1	0	1	1
norway22154	3.42	FALSE	242.14	0.525	0.376	OBJ2100	1	0	0	0	0
norway22110	5.25	TRUE	1447.01	0.315	0.369	OBJ2100	2	0	0	1	0
norway22130	2.43	TRUE	162.43	0.722	0.704	OBJ2100	2	0	0	0	0
norway22167	3.13	FALSE	399.75	0.657	0.613	OBJ2100	3	1	0	1	1
norway22126	1.51	FALSE	264.21	0.903	0.905	OBJ2100	1	1	1	1	1
norway22157	2.41	FALSE	437.01	0.645	0.682	OBJ2100	2	0	0	1	1
norway22172	5.20	FALSE	1686.13	0.208	0.426	OBJ2100	1	1	1	1	1
norway22181	3.00	FALSE	225.27	0.484	0.466	OBJ2100	1	1	0	0	1
norway22120	2.30	TRUE	163.92	0.724	0.873	OBJ2100	5	1	0	1	0
norway22140	1.58	FALSE	69.80	0.835	0.930	OBJ2100	1	1	1	1	0
norway22176	5.25	FALSE	264.71	0.525	0.180	OBJ2100	1	0	1	1	1
norway22115	3.80	FALSE	57.60	0.724	0.604	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22141	1.31	FALSE	190.09	0.893	0.968	OBJ2100	#N/A	#N/A	#N/A	#N/A	#N/A
norway22142	3.33	FALSE	522.21	0.303	0.343	OBJ2100	3	0	0	0	0
norway22162	1.70	FALSE	123.29	0.800	0.880	OBJ2100	1	0	0	1	0
norway22178	1.41	TRUE	141.78	0.878	0.784	OBJ2100	1	0	0	0	0
norway22103	1.50	FALSE	73.31	0.861	0.818	OBJ2100	1	0	0	1	1
norway22152	1.57	FALSE	216.83	0.761	0.834	OBJ2100	1	0	0	0	0
norway22173	1.15	FALSE	39.52	0.959	0.977	OBJ2100	1	0	0	1	1
norway22131	1.00	TRUE	88.68	1.000	0.472	OBJ2100	1	1	1	1	1

NTNU

Norwegian University of
Science and Technology