Master's thesis

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

Jack Minsoo Hurum Syvertsen

Magno: A Platform for Dyslexia Screening

Master's thesis in Computer Science
Supervisor: John Krogstie
June 2022

NTNU
Norwegian University of Science and Technology

Jack Minsoo Hurum Syvertsen

Magno: A Platform for Dyslexia Screening

Master’s thesis in Computer Science
Supervisor: John Krogstie
June 2022

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

Dyslexia is regarded as the most common of all learning disabilities. The ability to detect dyslexia early on is critical for decreasing the negative impacts of dyslexia. There appears to be a deficiency in the magnocellular system in people with dyslexia, this system is responsible for motion detection in visual processing. Three separate tests have been developed to test an individual for motion detection deficiency. This master's thesis implements an administrative platform that integrates the existing tests, with the aim of creating a system which is well accepted by teachers in primary schools. This is achieved through an iterative process, creating and testing designs with teachers, and implementing the platform. The result of this has been a platform with a new fully functional graphical user interface, a server which handles API calls, and a database storage system. The platform also integrates the existing tests, and allows teachers to add students, and test students. The platform with the tests has a System Usability Scale (SUS) score of 92.2, an increase compared to the existing tests. Feedback from teachers and special educators testing the finished platform indicated that it was well accepted, and that there were already cases where it could be useful. This thesis has provided a platform which, with some security improvements can be utilized for testing children in primary schools for dyslexia.

Sammendrag

Dysleksi er ansett som en av de mest vanlige leseforstyrrelsene. Evnen til å oppdage dysleksi tidlig er kritisk for å kunne redusere de negative innvirkningene dysleksi kan ha. Det magnocellulære systemet virker å være svekket hos folk med dysleksi, dette systemet er ansvarlig for visuell bevegelsesoppfatning. Tre separate tester har blitt utviklet for å teste individer for svekket bevegelsesoppfatning. Denne masteroppgaven implementerer en administrativ plattform som integrerer de eksisterende testene, med mål om å lage et system som blir godt mottatt av lærere i barneskolen. Dette er oppnådd gjennom en iterativ prosess med utvikling og testing av design med lærere, og implementasjon av plattformen. Resultatene av dette er en plattform med ett fullt funksjonelt grafisk brukergrensesnitt, en server som håndterer API kall, og en database for lagring av data. Plattformen integrerer også de eksisterende testene, og tillater lærere å legge til og teste studenter. Plattformen med testene har en System Usability Scale (SUS) poengsum på 92.2, en økning sammenlignet med de eksisterende testene. Tilbakemeldinger fra lærere og spesial pedagoger som testet den ferdige plattformen indikerer at den var godt mottatt, og at det allerede fantes tilfeller hvor de tenker at den kunne brukes. Denne masteroppgaven har gitt en ferdig plattform som, med noen sikkerhetsforbedringer kan bli brukt for å teste barn i barneskolen for dysleksi.

Preface

The work presented in this thesis is part of my master's degree in computer science at the Norwegian University of Science and Technology (NTNU). The project is conducted under the Department of Computer Science under the supervision of Professor J. Krogstie.

I would like to thank Professor John Krogstie for his helpful guidance, support and advisory throughout this master's thesis. This thesis would not have been possible without his help.

A specials thanks to my friends and family for being there for me, and for their support which has helped me complete my master's degree here at NTNU.

Jack Syvertsen

June 30, 2022
Trondheim

Table of Contents

List of Figures

List of Tables

1 Introduction

The number of people affected by dyslexia varies according to the type of test used, with recent research showing that it affects between 5-12% of the population [57]. Moreover, recent studies have shown that 49% of dyslexia diagnoses in Norway do not occur until 8th grade or later. This leads to a significant reduction in the effectiveness of support mechanisms, which have been shown to have a greater impact on young children in the first years of primary school [36]. Furthermore, people with late diagnosis often struggle academically as a result of these late diagnoses. In addition to affecting academic performance and well-being, it can also lead to stigmatization and an increased school dropout rate. As such, the negative effects do not just impact the individuals involved, but also society at large [3].

1.1 Motivation

Dyslexia is regarded as common learning difficulty that is the cause of specific difficulties with abilities such as reading and writing which are necessary for learning. It is also a lifelong issue for those affected, and can be the cause of multiple challenges on a daily basis [34]. Despite the fact that dyslexia is regarded as common, a substantial number of both children and adults face issues due to not being diagnosed with the disorder. These people may struggle at work or have academic problems without being able to attribute it to any specific reason. These struggles can have a negative effect on a person's well being and self-esteem [20]. However, if the disorder is diagnosed early, the consequences may not be as severe. A diagnosis shows people the reason for their struggles, and gives schools and other institutions an opportunity to implement targeted measures. These measures can mitigate the effects of dyslexia and often include targeted training and facilitation.

In Norway, screening tests for dyslexia are not always performed as they are not mandatory. Often tests are only conducted when a teacher or parent become aware, and think that it is likely that a child has learning difficulties [37]. There are national reading and writing tests that can give indications that a child is struggling, however these tests do not separate between learning disabilities such as dyslexia and other factors e.g. poor conditions at home. This can often lead to a delayed diagnosis as has been shown in an assessment of dyslexia diagnosis in Norway. In the assessment, 59% of participants responded that diagnosis being set in sixth grade or later was a common occurrence [3]. It is likely that early and targeted screening for dyslexia would help more people receive a correct diagnosis at an earlier stage, and could also lead to targeted measures and facilitation being up to 70% more effective [36].

Magno consists of tests that are based on the magnocellular theory of dyslexia. Results from earlier Magno projects have been shown promise regarding its utility as a screening tool. Projects undertaken in the past have created a dyslexic-friendly prototype with a functional user interface, while assisting in building more evidence for the magnocellular theory [58][24][23].

So far, the application has not been developed for use in schools with children as a supervised testing tool. The goal of this project is to examine what systems and adaptations are necessary to transform Magno into an effective platform solution that teachers can use to screen children for dyslexia in classrooms, and to act as a decision support tool when referring children for additional testing. A key objective of this project is to not only create a application platform that integrates existing tests, but that also allows for easy integration of future tools and tests.

1.2 Report Outline

This paper consists of nine chapters, Chapter 2 describes the research approach, along with research goals and questions for the project. Chapter 3 provides information gathered from the literature review performed during the specialization project, with the main focus of the review being information regarding dyslexia, and theories surrounding it. Chapter 4 described related work that has been done previously on the Magno project, and functional and non-functional requirements for this project. Chapter 5 gives a brief description of the methods and tools that have been used in the project. Chapter 6 describes the design, and how the platform and servers were developed and implemented. Chapter 7 gives further information regarding the usability testing and results from the tests, including the semi-structured interviews. Chapter 8 provides an evaluation of the project and its results. Chapter 9 presents a discussion around the project, in addition to a conclusion and what further work is recommended.

2 Research Approach

The research approach followed during this project will be introduced in this chapter. The research goals are presented in section 2.1, followed by the research questions in section 2.2. Section 2.3 contains a description of the research method.

2.1 Research Goals

The aim of this master's thesis is to implement a new application platform that will allow Magno to function as a dyslexia screening tool that can be used by teachers and special educators in primary schools. This includes the design and development of a user interface that supports login and registering functionality, administration of students, test results and tests. The platform will integrate the existing Magno tests, while ensuring that integration of updated or new tests is both possible and easy to achieve. In addition to this, the new user interface and system will be assessed with regards to usability and specific technology acceptance constructs by utilizing questionnaires and semi-structured interviews with both teachers and special educators.

2.2 Research Questions

The project aims to answer the following research questions.

RQ-1 *How can the existing Magno tests be integrated into an application platform?*

RQ-2 *How should the platform be developed to not restrict framework and technology choices for future tests?*

RQ-2.1 *How should the platform be developed to make integration of new tests simple?*

RQ-3 *How should the new platform be designed and developed for it to be well accepted by teachers and special educators?*

RQ-3.1 *How does the new administrative platform impact the overall usability?*

2.3 Research Method

The research is based on Oates' model on research processes [39]. The following is an overview of the research approach used in this project, which is illustrated in Figure 1, where the red boxes indicate that these research approaches were utilized.

Design and creation will be the key research strategy, creating a new IT artifact for administering students, their results and performing tests in a supervised test setting. The design and creation follows the literature review and requirements derived in the specialization project, and these have formed the basis for the research questions listed above. Analysis will include data from questionnaires sent to participants in the usability testing of the design and system and data acquired from interviews from the monitored usability testing. Furthermore the system itself will by analyzed, with regards to code structure, design and implementation.

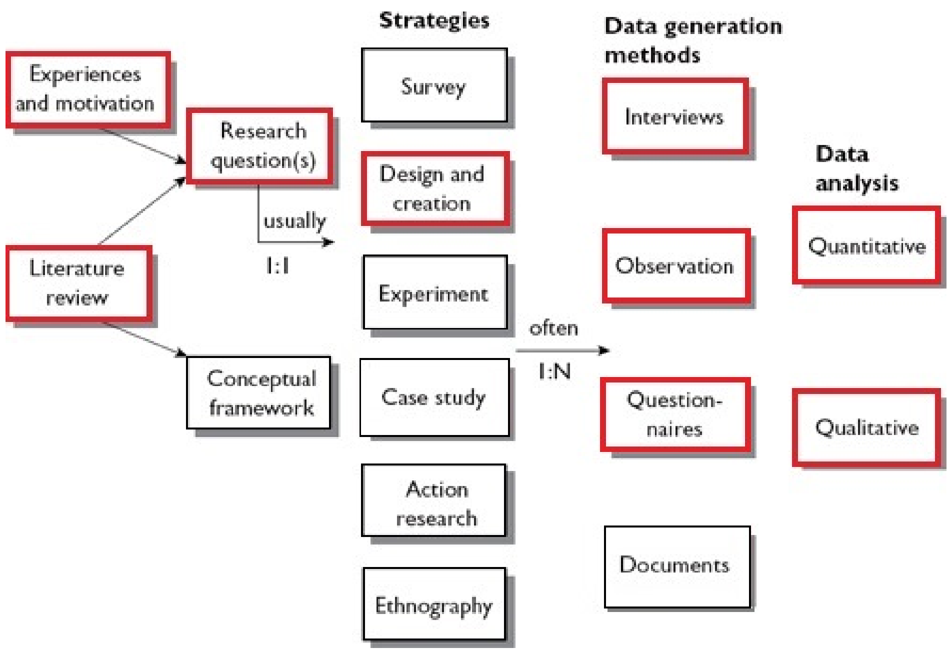


Figure 1: An overview of the research process, based on Oates [39]. The parts of the process utilized in this project are marked with a red border.

2.4 Evaluation

The new system will go through a qualitative analysis both during and at the end of development. Firstly, usability testing of the design will be performed with several teachers and adults. The usability testing of the design with teachers will be monitored, with a semi-structured interview taking place at the end of the test. The remainder of participants performing usability testing of the design will perform this in a self-managed setting, filling in a SUS form at the end of the test. The test of the actual system will be a qualitative test performed only with teachers and special educators. These tests will be guided and monitored, with semi-structured interviews being performed at the end of the test. Furthermore, the teachers and special educators will fill out two forms, one SUS form, and a form based on certain constructs from the Technology Acceptance Model (TAM) [7][15]. The TAM form will consist of four questions aimed at identifying how well the system is being accepted, and how likely it is for teachers and special educators to use the system. It is important to note that a full TAM won't be performed, the constructs used will be utilized in the questionnaire to give an impression of the perceived usefulness and intention to use and in the interview process as a way to receive more qualitative data.

3 Background

It is important to look back over the work that has already been accomplished with Magno, to examine the results, and to determine what more needs to be done to develop a system that can be used by teachers and special educators for dyslexia screening in classrooms. This chapter summarizes the important findings from the literature review that has been performed. In section 3.1 is an explanation of what dyslexia is, how it is diagnosed in Norway today, and why early detection and screening are important. Also discussed in this section are possible causes of dyslexia, which includes the theory upon which Magno is based, namely the magnocellular theory of dyslexia. In closing, the chapter describes usability briefly.

3.1 Dyslexia

3.1.1 Defining Dyslexia

As there are so many different definitions for dyslexia and they continue to be discussed, a single definition is difficult to come up with. Dyslexia manifests differently, which is why finding a single definition is difficult. For instance, one person may only experience one symptom, while another person may experience multiple symptoms.

The largest organization for dyslexics in Norway, *Dysleksi Norge* uses the following definition for dyslexia.

Dyslexia is a specific learning disability that negatively impacts functional reading- and writing skills. It is typically identified due to comprehensive difficulties with word decoding and spelling, in addition to difficulties with other language-related skills. The most common symptoms are difficulties with phonological processing, rapid denomination and phonological short term memory. Some also suffer from difficulties with language-related processing speeds and automation capabilities. [38]

The definition used by *Dysleksi Norge* is based upon operational definitions used by pedagogues and special educators, such as the ones compiled by the International Dyslexia Association (IDA), British Dyslexia Association (BDA) or the ROSE-report. [38]

IDA uses the following definition:

Dyslexia is a specific learning disability that is neurobiological in origin. It is characterized by difficulties with accurate and/or fluent word recognition and by poor spelling and decoding *abilities. These difficulties typically result from a deficit in the phonological component of language that is often unexpected in relation to other cognitive abilities and the provision of effective classroom instruction. Secondary consequences may include problems in reading comprehension and reduced reading experience that can impede the growth of vocabulary and background knowledge.* [4]

In 2009, the ROSE-report outlined the following definition of Dyslexia:

Dyslexia is a learning difficulty that primarily affects the skills involved in accurate and fluent word reading and spelling. Characteristic features of dyslexia are difficulties in phonological awareness, verbal memory, and verbal processing speed. Dyslexia occurs across the range of intellectual abilities. It is best thought of as a continuum, not a distinct category, and there are no clear cut-off points. Co-occurring difficulties may be seen in aspects of language, motor coordination, mental calculation, concentration, and personal organization, but these are not, by themselves, markers of dyslexia. A good indication of the severity and persistence of dyslexic difficulties can be gained by examining how the individual responds or has responded to well-founded intervention. [5]

In 2010, the BDA adopted and extended the ROSE-report's proposed definition, adding the following:

The British Dyslexia Association (BDA) acknowledges the visual and auditory processing difficulties that some individuals with dyslexia can experience and points out that dyslexic readers can show a combination of abilities and difficulties that affect the learning process. Some also have strengths in other areas, such as design, problem-solving, creative skills, interactive skills, and oral skills. [46]

Among all the organizations, the BDA is the only one that recognizes dyslexics' difficulties with visual and auditory processing, and that they might also be actual causes of dyslexia, which is discussed in Section 3.1.3. In spite of the fact that the definitions differ in certain ways, they all agree that dyslexia impairs the ability to read and spell accurately and clearly.

3.1.2 Why Early Detection is Necessary

The consequences of dyslexia can be severe for those affected, as well as for the individual's family and society as a whole. As of now, dyslexia screening tests in Norway are not mandatory, but specific tests may be conducted if the teachers or parents determine that the child has problems with reading and writing. [37] A child's reading and writing abilities are typically determined by mandatory tests which are administered to him or her during the first three years of primary school. While these tests are useful to know that a child is struggling, they cannot give a reason why [38]. According to a recent assessment of dyslexia diagnoses in Norway, current testing procedures are problematic. In the assessment, 59% of special educators responded that setting a diagnosis in sixth grade or later is common [3].

The consequences of setting a diagnosis at such a late stage are significant. In the 1st and 2nd grades of primary school, the effectiveness of targeted measures and facilitation is at around 80%, whereas in the 5th or later grades, the effectiveness drops to about 10-15% [36]. Furthermore, children that have not been diagnosed before the fourth grade no longer qualify for individualized support, which further compounds this problem. Consequently, it becomes even harder to determine if a student has dyslexia or is struggling due to other reasons. By the time the child reaches this grade of primary school, good reading and writing skills are required otherwise the syllabus may become more challenging than intended.

In a study, Livingston et al. [27] investigated possible consequences and effects caused by dyslexia and how they affected individuals, families and society as a whole. The study looked at close to 100 articles published between 1980 and 2018 on the subject matter. Livingston et al. concluded that dyslexia could have both primary and secondary consequences, and that these could be categorized accordingly as shown in Figure 2.

The primary consequence of dyslexia is a lower level of performance both at work and in academics, as well as differences noticed by peers, teachers and family and an implied or real stigma caused by the perception of being different. The lower performance and stigma may also lead to feelings of inadequacy and a decreased sense of self-worth, which affect self-esteem, motivation, emotional well-being and social relationships, all of which are considered secondary consequences. If not diagnosed early, children are particularly at risk for motivation loss due to not knowing why they are performing poorly.

The study concluded that early detection and remediation could reduce or negate the consequences of dyslexia. Furthermore, it concluded that early detection and treatment is crucial to enabling positive outcomes in the areas of social, emotional, economic, and academic development for those affected.

3.1.3 What Causes Dyslexia?

According to recent studies, dyslexia is regarded as the most common of all learning disabilities, affecting between 5 and 12 percent of the population with estimates differing depending

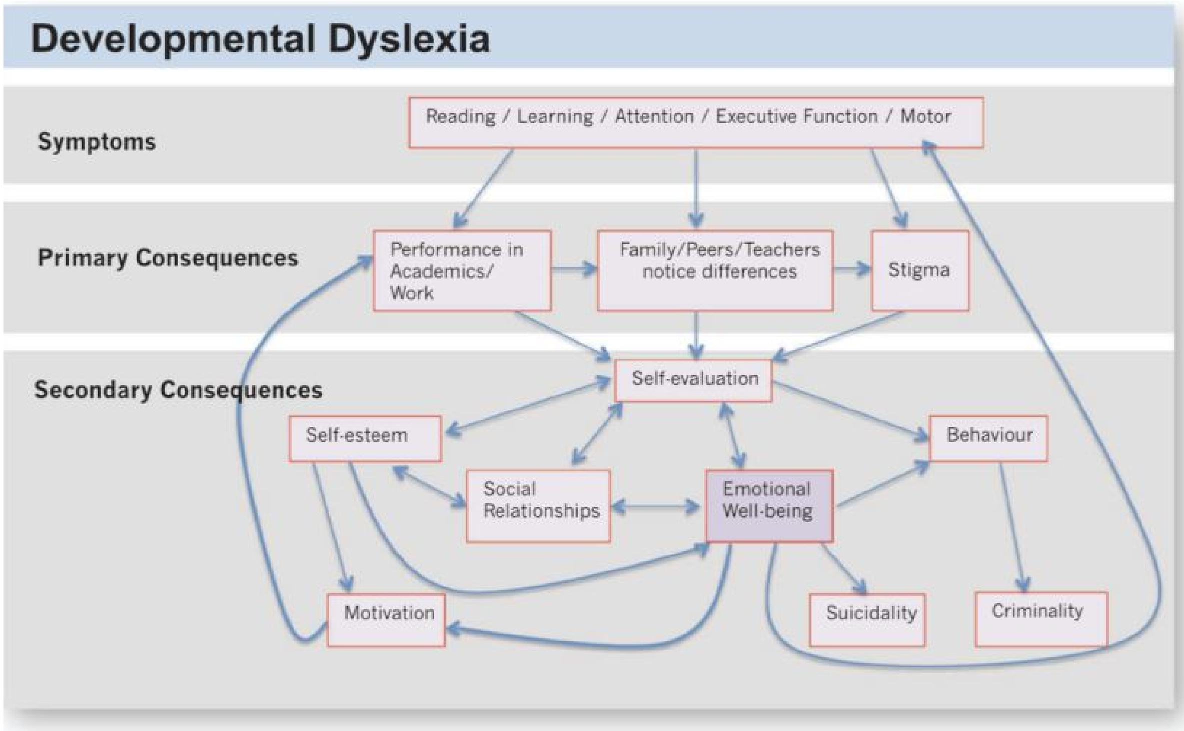


Figure 2: Primary and secondary consequences of dyslexia

on the test used[57]. Despite of this, the exact cause or causes of dyslexia have not been agreed upon by researchers. Nevertheless, what is evident is that reading is a complex skill that requires multiple cognitive processes; and in dyslexics, some or all of these processes are weakened. As a result, theories about dyslexia often focus on cognitive processes associated with reading, such as visual processing, rapid naming, verbal short-term memory, and phonological and orthographic coding [48].

The ability to read does not only require one to be able to visually identify letters and letter orders. One must also be able to comprehend the phonological structure of words. In order to do this, one needs the ability to break words down into phonemes, converting letters into sounds [51]. One of the more widely accepted theories of dyslexia concerns phonological deficits, which is an impairment of the phonological processes in a person [48].

The Phonological Deficit Theory

The phonological deficit theory asserts that dyslexics' difficulties with reading stem from a diminished ability to separate words into phonemes. More specifically the theory argues that dyslexic individuals of all ages display phonological processing problems, and that dyslexia should be considered a core phonological deficit. [50] Numerous tests have already been developed to evaluate phonological processing, and one of the most common consists of participants reading fictitious words such as 'gletike'. This test measures an individual's ability to turn letters into sounds, and since the word has been made up, meaning or context are no longer relevant to the outcome [53].

Yet some critics have objected to the theory because it does not provide information as to why children score poorly on these tests. Furthermore, the tests do not distinguish between dyslexics and poor readers, since all children struggling with reading have problems with phonemes. Moreover, many dyslexics do not seem to have phonological issues, which does not fit into the theory [41]. The critics contend that phonological processing difficulties are not enough to distinguish dyslexics from people who are struggling due to other factors, such as poor parenting or inadequate education [53].

The Parvo- and Magnocellular Systems

The ability to analyze text visually is required before one is able to develop phonological processing abilities. A key component of this process is the ability to identify each letter separately, and then to sequence the letters correctly. These abilities are dependent upon the parvo- and magnocellular systems [53]. Researchers have studied primates' visual processing and found that the parvo- and magnocellular pathways are responsible for the transfer of information between eyes and brain. Even though the visual processing in humans and primates are not perfectly correlated, the researchers assume that there are multiple similarities. According to these studies, the parvocellular (P) neurons are used to process detailed form, and colors, whereas the magnocellular (M) neurons are responsible for detecting motion and other rapid changes in visual perception [9].

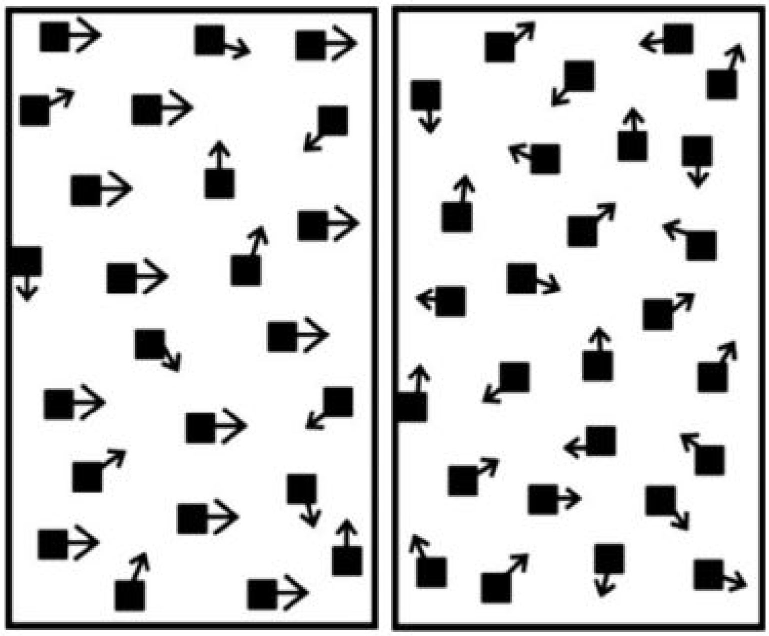
A deficiency in the M neurons found in the retina is thought to contribute to timing issues. It is believed that these impairments interfere with a person's ability to read efficiently and they are related to the link between M- and P pathways. The timing issue either impairs a person's ability to discriminate the direction of change in visual stimuli, resulting in a problem with phonemic awareness, or impairs their ability to discriminate the direction of movement of moving patterns. The direction-selectivity network is primarily controlled by neurons of the M class, because these neurons have large axons and dendritic arbors, making them sensitive to motion. However, this has a downside in that the M neurons are unable to distinguish fine details, such as the edges of the letters in a word. P-neurons are responsible for performing this function, since their axons and dendritic arbors are small and they're particularly sensitive to fine details. So, an impaired M system might result in a person having trouble with timing, since the brain misinterprets where there is activity, and where the attention and the P-neurons should be focused [26]. With dyslexics, letter sequencing is often slow and more likely to contain errors, which has contributed to forming the basis for the magnocellular theory that hypothesizes dyslexia is caused by an impairment in the magnocellular pathway [51].

The Magnocellular Theory

According to the magnocellular theory, dyslexia is associated with impairments in visual processing. The impairment of the brain's visual system and the difficulties dyslexics have with vision may be related to the impairment of the brain's visual system, according to a study by Stein and Walsh [54]. This theory does not disregard the possibility that a phonological deficit might also occur, despite focusing on impairments of the visual system. Moreover, the magnocellular theory also encompasses all known dyslexic impairments, which include phonological, auditory, visual [52], as well as motor and tactile problems [42].

A growing body of research and evidence supports the magnocellular theory. An examination of the brains of five dyslexics conducted post mortem through micro anatomic examination and MRI revealed irregularities in the magnocellular- (M) system. In contrast, no irregularities were observed in the non-dyslexic control group [28]. A comparison of the M-layers of the lateral geniculate nucleus (LGN) in dyslexics with that in the non-dyslexics group revealed that the M-layers were 30% smaller and more disorganized in the brains of dyslexics. Furthermore, fMRI studies of dyslexics were used to find activity in the visual motion area, for which the M-system is mainly responsible. The researchers found that the dyslexics were less responsive to motion in comparison to those without dyslexia [10].

Through testing the sensitivity of the middle temporal visual motion area (V5/MT) in dyslexics, further evidence has been gathered that supports the theory of an impaired M-system in dyslexics. Motion detection is handled by this system, which is located in the cortical dorsal pathway, which consists primarily of M-neuron cells. In other words, 90% of visual information relayed to the V5/M5 system is handled by the M system [51]. A person's ability to detect coherent motion can be tested in order to assess the sensitivity of the V5/MT system. Random dot kinematicogram tests can be used to test this. These consist of a multitude of moving dots, some of which move directly right and left in a pattern, while others move randomly. In order to measure a person's sensitivity to motion, the amount of cohesion that is required to detect it is tested, that is, how many dots must move together coherently before the motion can be detected. Those with dyslexia have been found to be less sensitive to this occurrence, as they require a much greater number of dots to move coherently for the movement to be identified as coherent [51]. An example of a random kinematogram test can be seen in Figure 3.



(a) Some dots are moving coherently

(b) All dots are moving randomly

Figure 3: An example of a random dot kinematogram test

In a second study by Lawton, direction discrimination training (DDT) was utilized to improve magnocellular function. Over the course of three months, dyslexic children received DDT in a pattern designed to maximize activation of magnocellular neurons. Based on the study's findings, the magnocellular theory is well supported, since targeting the magnocellular system led to an 11 fold increase in reading rates, in addition to improvements in spelling, word identification, and comprehension by 1-3 grade levels, with the improvements persisting over time [26]. Figure 4 illustrates how patterns are used for training direction discrimination, here the subject must determine if the pattern is moving from left to right or from right to left when it cycles.

Figure 4: Background patterns of differing difficulty with a “fish shaped” test pattern. The most difficult is when the background cycles at the same frequency as the "fish shaped" test pattern.

3.2 Quality Attributes

3.2.1 Modifiability

Modifiability is a key quality attribute in the development of a platform that allows new dyslexia screening tests and tools to be integrated in a simple manner. This attribute has been standardized in the ISO 25010 standard, which places it under the maintainability characteristic. The ISO standard states that modifiability is the degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality [21]. In addition, coding, design, documentation, and verification of changes are cited in the standard as factors that influence modifiability. Furthermore, both modularity and analysability influence the modifiability of a system.

- Modularity - degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components [21]
- Analysability - degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified [21]

3.2.2 Usability

As one of the goals of this project is to develop a platform that should be well accepted by teachers and special educators, it is important to have knowledge of usability. ISO 9241-11 standardized usability, stating that usability can be measured through several factors, which are the solution's effectiveness, efficiency and user satisfaction. Moreover, the standard states that usability is dependent on context. Here the context refers to the task being performed, and both the experience and the background of the user performing the task, and the environment where the task is performed [22][6].

4 Related Work

The chapter provides information about Magno from its earliest days, as well as information about the subsequent iterations. It includes insights and findings from earlier iterations of the project that have been considered as important for future work.

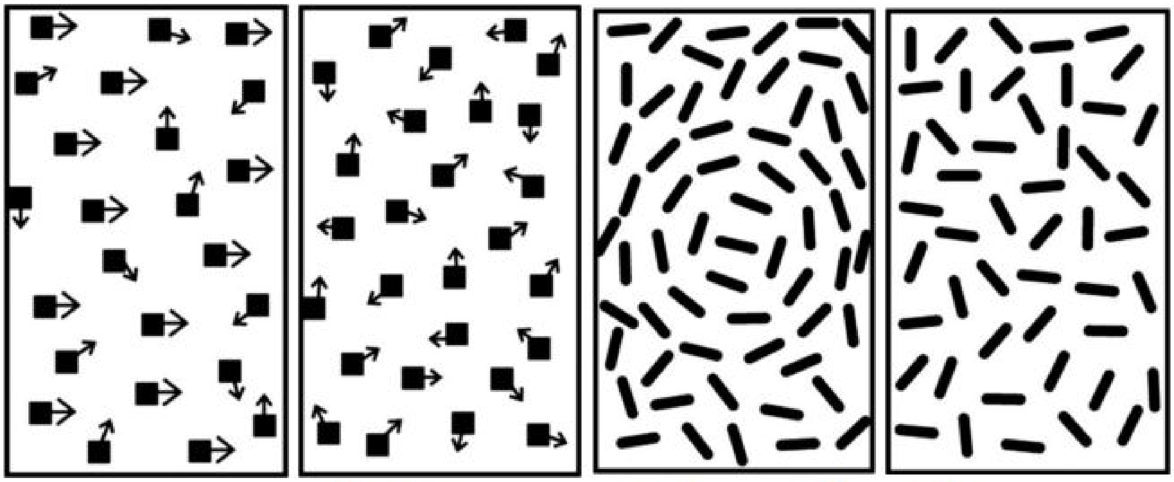
4.1 Magno: An App for Early Detection of Dyslexia

4.1.1 Iteration One: Functionality

In the beginning, Magno was developed as a reimplementation of an early 2000s prototype, *Form* [18], developed by Hansen, Stein, Ordre, Winter, and Talcott. The purpose of Form was to assess the visual processing abilities of individuals and to check whether they suffered from visual impairments. Several tests are included in the program in order to assess visual processing abilities. Because *Form* was created in the early 2000s, and it was designed for MS-DOS, which had its final release on September 14, 2000 [31], it was determined that modernizing it to work with current computers and tablets was a necessity and desirable. Master student Wold re-implemented Form as part of the Magno project in 2015-2016. This new program, developed by Wold, was a Java application that includes random-dot kinematogram tests, as well as two forms tests using static global patterns.. These tests are illustrated in Figure 5.

In an effort to create the tests using modern technology whilst preserving the functionality of the old tests, Wold interviewed those responsible for the development of *Forms*. Java and libGDX were used to develop the program, with libGDX being a development framework that's ideal for creating games. The new version of the program included a main menu, a motion test, two form tests, and a settings screen. Figure 6 shows the main menu of the program.

Motion Test
The motion test has two squares containing 300 randomly placed dots. Each dot has a radius of 1 pixel, and is placed at least 1 pixel away from the other dots. Each level includes two squares, with a randomly selected square containing dots moving in a coherent pattern. The dots which are moving coherently move horizontally at equal intervals from left to right, reversing direction every 0,572 seconds. Randomly moving dots change direction after 0,572 seconds, or whenever they collide with another dot. Default settings have all dots moving at



(a) Some dots are moving coherently

(b) All dots are moving randomly

(c) Pattern visible in line segments

(d) No pattern in line segments

Figure 5: Random dot kinematogram tests shown in (a) and (b) and form tests with static global patterns shown in (c) and (d)

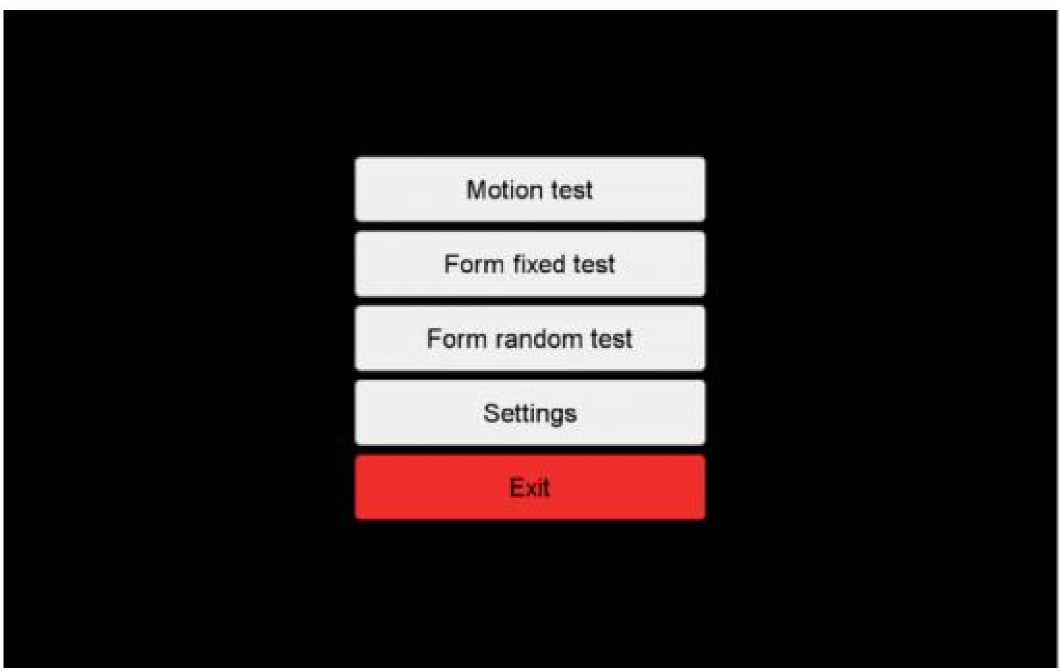
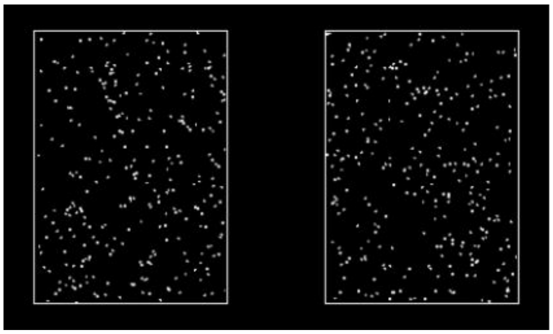


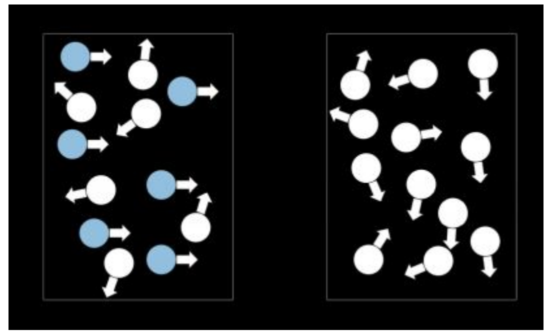
Figure 6: Magno's main menu in the first iteration

a rate of 50 pixels per second.

During the test, two squares are shown simultaneously for five seconds. After this time has passed, the dots are removed, and the person is asked to choose which square contained coherent moving dots. Depending on which square a person selects, the levels of coherence is changed and the dots' locations and movement patterns are changed. Whether a person chose the right square will determine the level of coherency. In the case where the person is right, the coherency decreases, and in the case where the person is wrong, the coherency increases, making each level either harder or easier. An illustration of a test with 50% coherently moving dots is shown in Figure 7.



(a) Screen capture of the motion test



(b) Motion test illustrating 50% coherency

Figure 7: (a) shows a screen capture of the re-implemented motion test and (b) illustrates a motion test with 50% coherently moving dots

The Form Tests

There are also two forms tests in the program; a fixed and a random form. A static global pattern can be found in this set of tests, which have a fixed target in the form fixed test, and a random target in the form random test. The tests consist of 600 lines evenly distributed throughout a square. Each line segment has a thickness of 1 pixel and a length of 0,4 degrees of the viewing angle.

In the same way as the motion test, the person being tested receives two squares for every level taken. In the test, the lines of one of these squares are selected at random to form circles, and a fixed point will be set at the center of the square in the form fixed test, and the circles will always center around it. For the form random test, the center of the circles within the square will be chosen at random. In these tests, the individual has to identify the correct square by selecting the square that contains circles within a given period of time, which has been set to four seconds. As soon as the four seconds have passed, the lines vanish, and the person selects the square that they believe is correct. In the same way as the motion test, the difficulty increases or decreases depending on whether the square selected was correct. Choosing the correct square will result in a decrease in the percentage of line segments forming circles, while choosing the incorrect square will result in an increase in the percentage. Figure 8 shows the fixed form test at 100% coherency, and Figure 9 shows the test at 50% coherency. The form random test at 100% and 50% coherency can be seen in

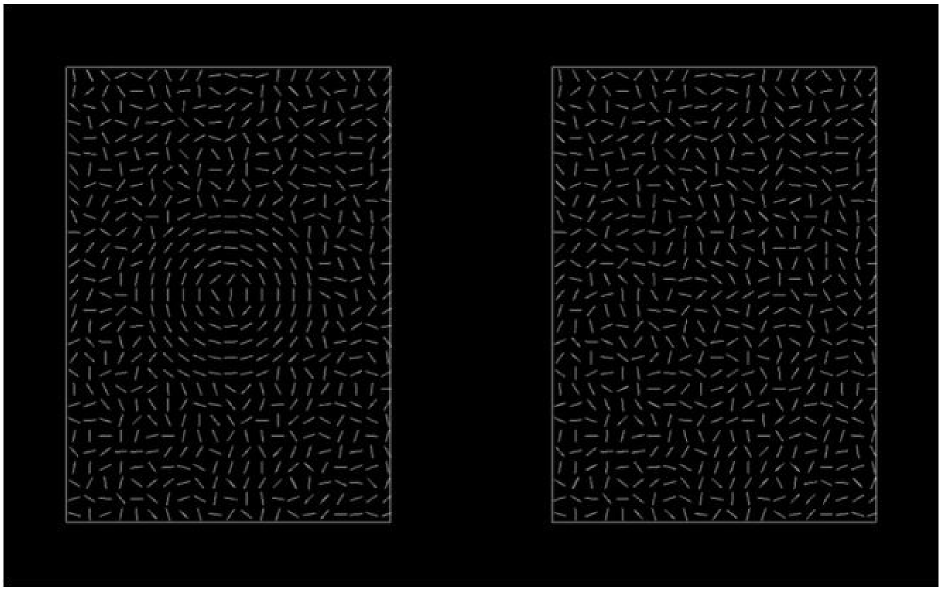


Figure 8: The form fixed test at 100% coherency with the left square containing line segments forming a circle around the center of the square.

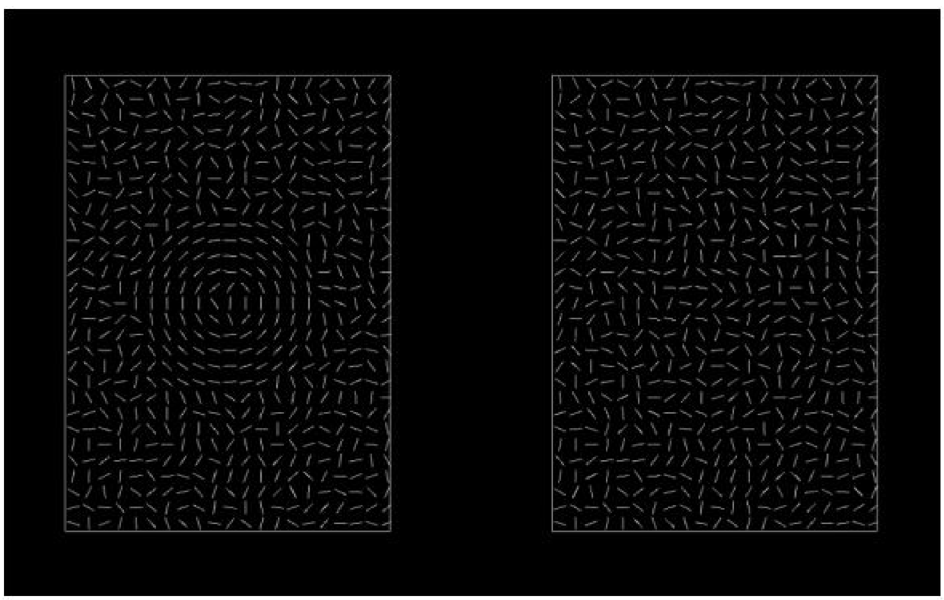


Figure 9: The form fixed test at 50% coherency with the left square containing line segments forming a circle around the center of the square.

Figure 10 and Figure 11.

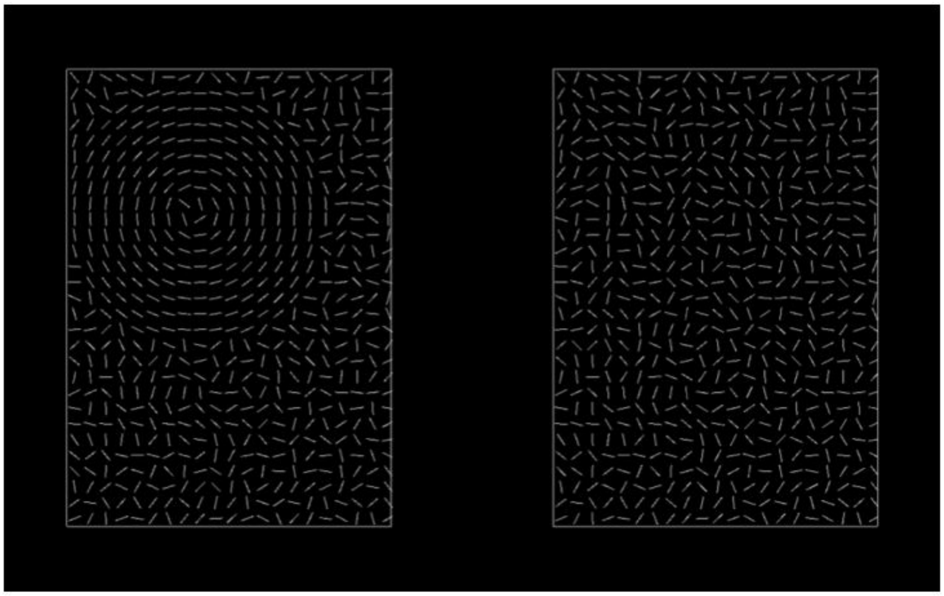


Figure 10: The form random test at 100% coherency with the left square containing line segments forming a circle around a randomly selected point

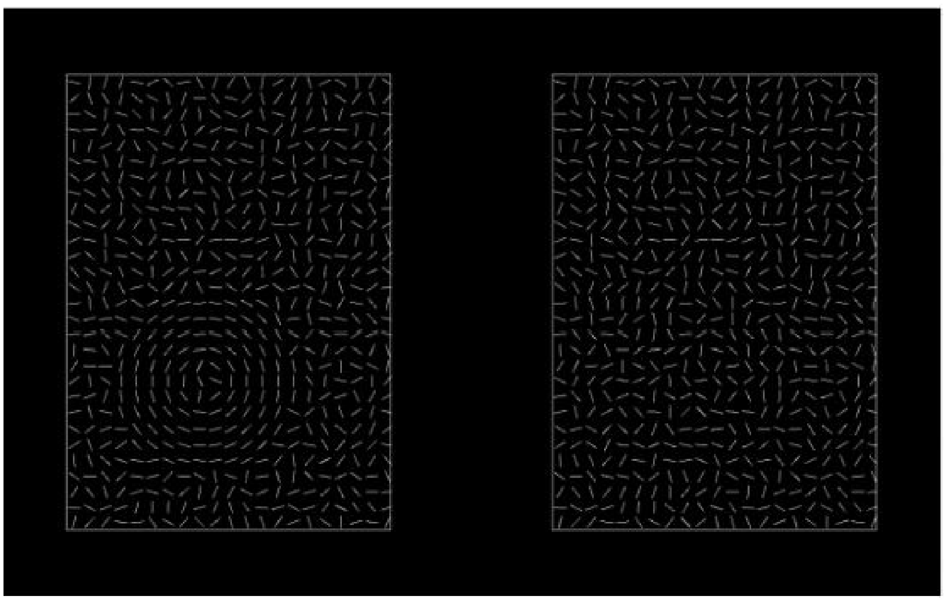


Figure 11: The form random test at 50% coherency with the left square containing line segments forming a circle around a randomly selected point

4.1.2 Iteration Two: Improving Usability

In their second phase of Magno development, the two master students Johansen and Kirkerød [24] concentrated on improving Magno's usability after the first iterations focus on functionality. Their goal was to design and implement a user interface that was easy to use and dyslexic friendly. They performed a number of iterations, creating new solutions and testing them with users. The final design by Johansen and Kirkerød included a navigation bar so users could easily find each section of the program. Additionally, a tutorial to help users manage their own tests, a way to enter a user's age for statistical analysis, and a test results screen that provided the results of the test were added.

Dyslexics have indicated that they have trouble reading due to visual issues, including overlapping letters, letter rotation, and switching order of letters [43]. Dyslexics would also face this problem when using Magno, so Johansen and Kirkerød made modifications to not only improve the usability but also make it more accessible for dyslexics in particular. In addition to color overlays, the reading area background was changed, justified text alignments were avoided, double spacing after end of paragraphs was used, and black text was not displayed on a white background. Screenshots of the new menu, and screens can be seen in Figure 12.

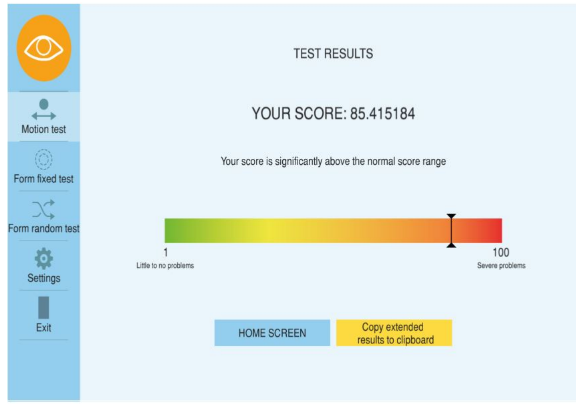
4.2 Magno Web App for deCODE's Application Platform

During 2020-2021 Fredrik Jenssen, a master student, developed the next version of Magno [23]. Jenssen planned once again to reimplement Magno's functionality, this time as a web application that would be integrated with the application platform called Svipgerð.is developed by DeCode. The company deCode genetics is headquartered in Iceland and specializes both in analyzing the human genome and in understanding it [8]. The company has access and is able to distribute the program to about 15 000 people who suffer from reading disorders. As such, integrating Magno with deCode's application platform would make it possible for the system to be tested with and information to be gathered from a much larger population than previously possible.

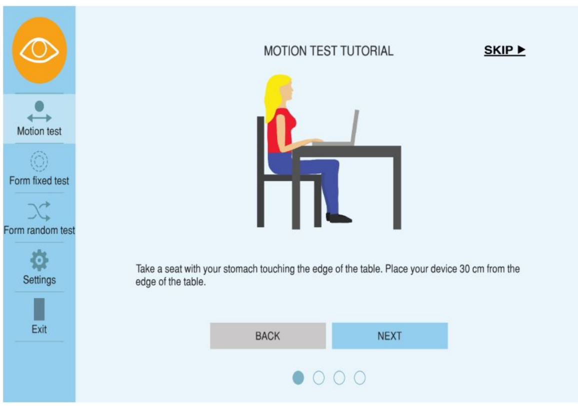
As Magno was a Java program, it was required to convert it to a web application using programming languages and frameworks designed for web applications. The software used by Svipgerð.is runs on an AngularJS and Bootstrap front-end, an Express.js server for API calls, and an Oracle database for data storage. The architecture of Svipgerð.is is visible in Figure 13. Jenssen's re-implementation mainly used TypeScript and PixiJS, and as requested by deCode was implemented as independent applications. This would allow for each test to be taken separately, and simplify integration with Svipgerð.is. The proposed architecture is visible in Figure 14 which includes the motion, form fixed and form random test.

Some aspects of the design of Johansen and Kirkerød's implementation changed when the tests were developed as independent applications. As there was no longer a need for naviga-

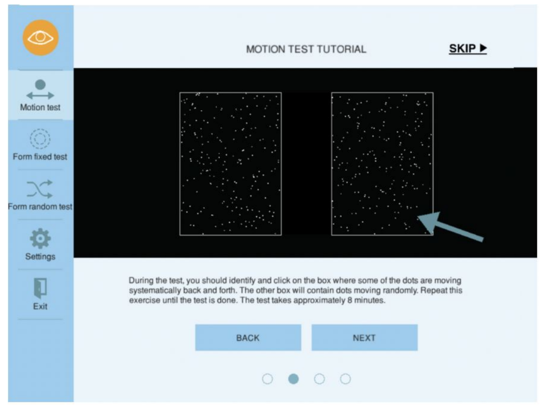
(a) Home screen with navigation bar to the left



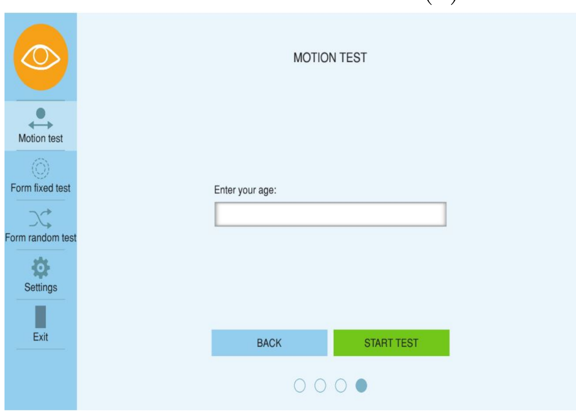
(b) Test results screen



(c) First screen of the tutorial



(d) Second screen of the tutorial



(e) Enter age screen

Figure 12: The updated user interface for Magno, designed to improve usability and to accommodate dyslexics

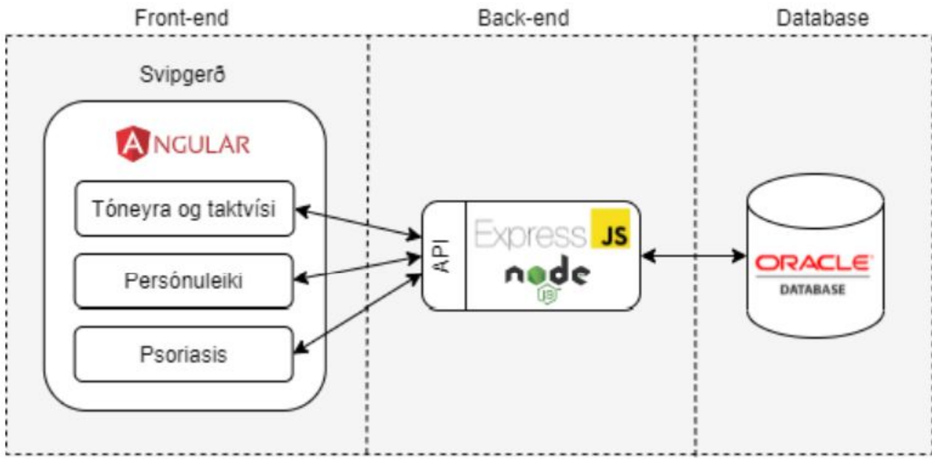


Figure 13: Architecture for Svipgerð.is

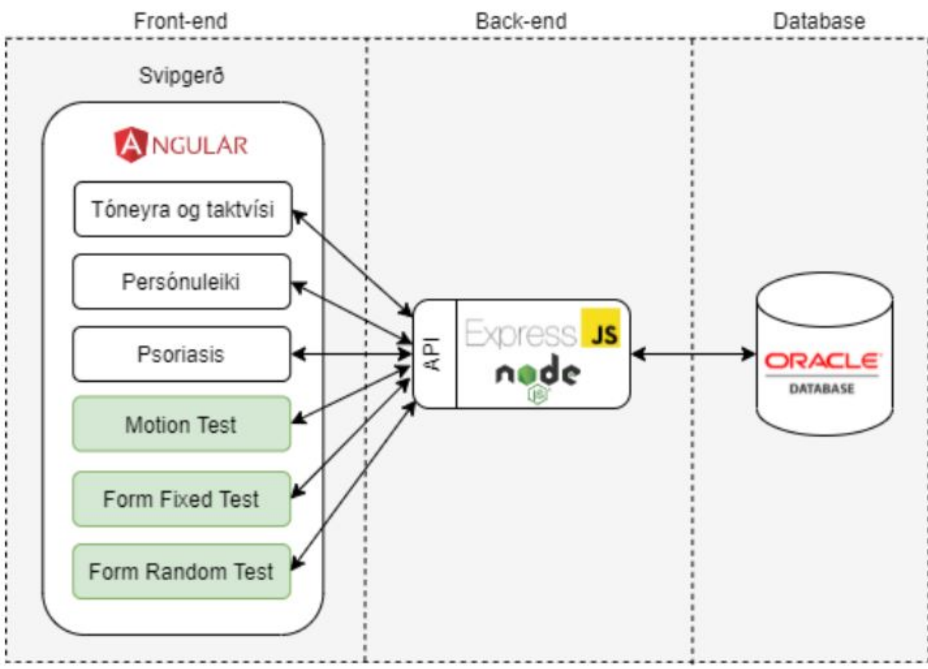
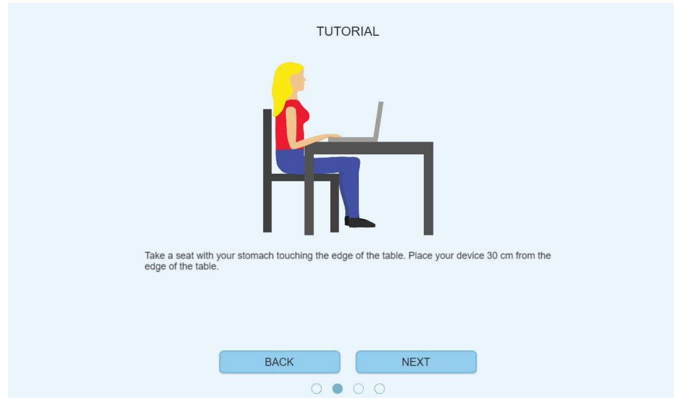


Figure 14: Svipgerð.is with the motion, form fixed and form random tests included as new modules

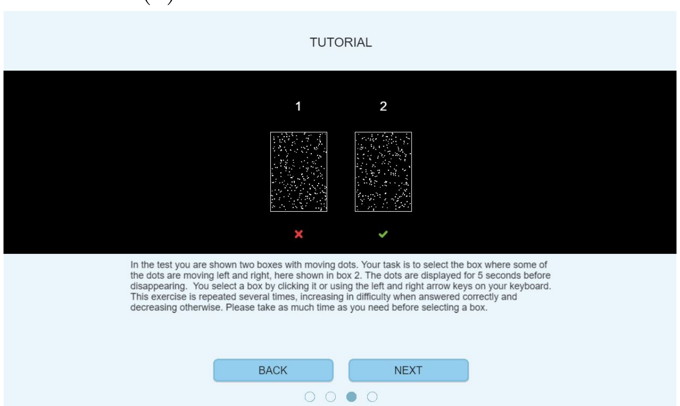
tion between tests, the left-hand navigation bar on the screen was removed. As a result, the settings option was also removed. The original design has also been revised in other minor ways, including removing the gender and age inputs, since Svipgerð.is handled those. The new design is visible in Figure 15.

Jenssen considered many of the functions that are common in web applications out of scope, since Svipgerð.is was already handling them. This includes authorization and authentication mechanisms, and age-group-specific scores, as well as handling multiple tests per

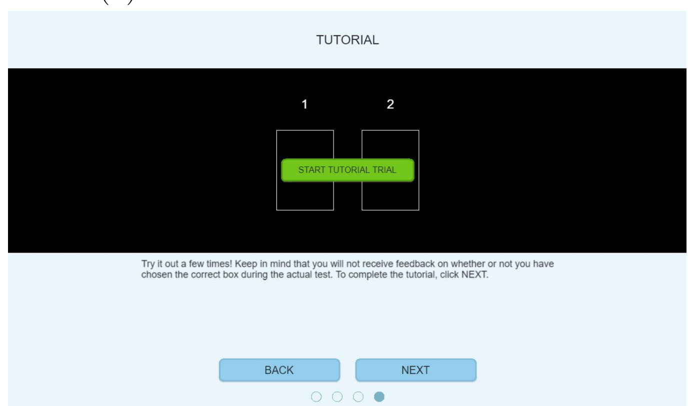
(a) Introduction screen



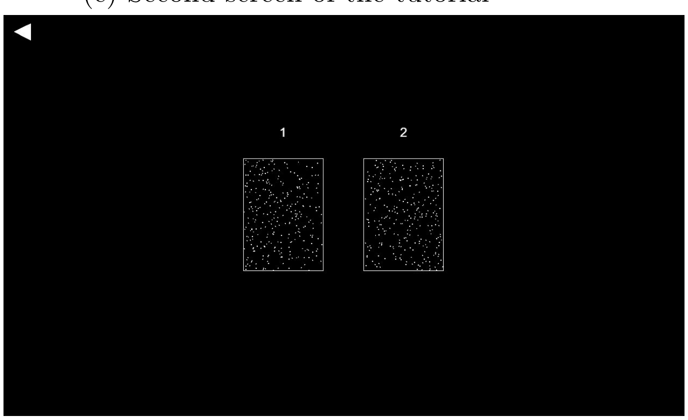
(b) First screen of the tutorial



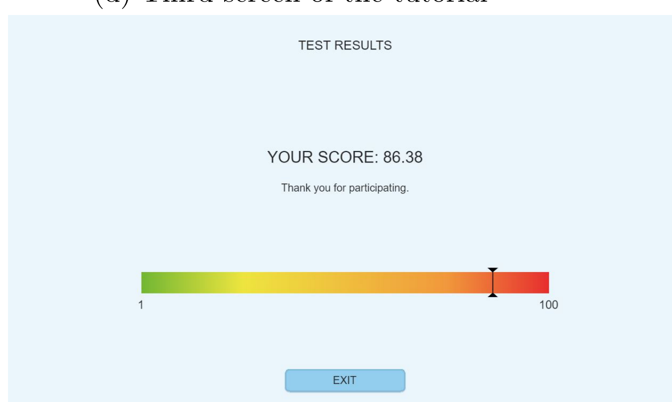
(c) Second screen of the tutorial



(d) Third screen of the tutorial



(e) Motion test



(f) Results screen

Figure 15: The updated user interface for the Magno Motion Test developed as an independent web application

user.Additionally, a server and database were not required for handling API calls or storing data. Moreover, Jenssen proposed that more usability testing should be conducted specifically on tablets, due to the low number of tests performed on these devices.

4.3 Screening Young Children

The specialization project of Pernille Klevstuen was carried out in the spring of 2021, and her objective was to assess what modifications were necessary to make Magno a platform for conducting dyslexia screenings of children [25]. By reviewing literature and interviewing an obstetrician and a special educator, she gathered several insights into what changes needed to be made to Magno in order to make it suitable for young children. Klevstuen's project identified several areas for improvement, most importantly the fact that the testing currently is designed for self-management. TIt was determined that the introduction to the test should be modified so that guidance could be given verbally to young children who might not be able to read yet. In addition, the results screen would likely be confusing to a child, and the results should be communicated by the supervisor if deemed appropriate, so as not to damage the self-esteem of a child. Feedback on the application also indicated that the application should not be used with children younger than 10-12 years, as it does not provide the motivation needed to motivate young children to finish the test.

4.4 Requirements

Requirements were specified using insights gained from the specialization project, and from previous Magno iterations. A lot of the functionality that is required from the new platform were specified as out of scope in the previous iteration of Magno perform by Jenssen 4.2, as these were already implemented in deCode's platform. More specifically they pertain to authentication, authorization, multiple test-runs per user, storing data in a database, and the creation and maintenance of an API for communication with the database. Furthermore, additional requirements were derived during interviews with teachers and special educators performed during the specialization project. These derived requirements were more focused on what was aspects were important for teachers and special educators to use the system, such as the test results giving immediate textual information regarding the risk. The requirements for this iteration can be seen in Table 3

Table 1: Functional and non-functional requirements for the platform and server

Id	Descriptions	Priority
FR1	It should be possible to register an account with an email address and a password	High
FR2	It should be possible to log in to a registered account by providing the correct email/password combination	High
FR3	The platform should allow users to register which school they work at	High
FR4	The platform should be responsive, and elements should adapt to fit the current screen size and resolution	High
FR5	The platform should be able to start motion and form tests and receive test results from these	High
FR6	It should be possible to add students to the platform	High
FR8	Users should be able to view all the students that have been added to the platform, and attend the user's school	High
FR9	Student test results should be viewable in a table or list	High
FR12	The platform should be able to display detailed results for each student	High
FR14	The database should store data necessary for the users	High
FR15	The platform should be able to communicate with the server API	High
FR17	The back-end API should be able to store user information in the database	High
FR18	The back-end API should be able to store student information in the database	High
FR21	The server should be able to serve both the platform and the motion and form tests	High
FR7	It should be possible to remove students from the platform	Medium
FR10	The table or list showing test results should be sortable	Medium
FR11	The table or list showing test results should be searchable	Medium
FR13	The platform should be able to give text based information regarding a student's test results	Medium
FR14	It should be possible to both use and navigate the platform without any prior instructions	Medium
FR19	The platform should have Feide integration for authentication and authorization	Low
FR20	The platform should fetch user and student information from Feide	Low
NFR1	The platform should have a minimum SUS score of 80	High
NFR2	The server and platform should be loosely coupled to the tests	High
NFR3	Developers should have access to documentation which is thorough and precise.	Medium

4.4.1 Out of Scope

It is important to not only describe what is important and required of the platform at this stage, but also to describe what has been deemed to be of less importance, and subsequently will not be a part of this iteration. Firstly, security has not been highly prioritized during this iteration, except for certain elements such as authentication, parameterized SQL queries, and secure salting and hashing of passwords. Other security elements were deemed to be too time consuming due to the amount of work required for implementation, penetration testing and verification, and due to these functions not having any impact on usability or technology acceptance during user testing.

Furthermore, any large changes to the motion and form tests themselves were regarded as being out of scope for this project, with the exception of small changes to text to clarify where results could be shown and interpreted and other small changes necessary to integrate them with the server and platform.

5 Methods, Frameworks, Tools and Technology

This chapter includes information regarding the frameworks, tools, technologies and method used for development. Furthermore, it includes specific information regarding how the usability tests were conducted.

5.1 Scrum and Agile Methodology

The development methodology for this project will be based upon certain elements from Scrum. Scrum is an agile process framework that is often used within software development and management of project [49]. As Scrum is based around larger teams consisting of five to ten members, only certain parts of the framework were used. These were an iterative approach to development, and active involvement of stakeholders in the process. The framework is meant to break down work into smaller pieces, which can be achieved in iterations that have a set amount of time, these are called Sprints. Normally, sprints last around two to four weeks, and following this the project will be divided into two sprints. The goal of the first sprint is to create an entire design for the new application platform, and performing usability testing of the design. The objective of the second sprint is to fully develop the new application platform. The design of the platform will be based upon the original design, implementing suggested changes from stakeholders, and revising areas that did not function well. Finally, a second usability test will be performed on the application platform , this will be compared not only to the usability of the design, but also to the usability of the individual tests that were developed by Fredrik Jenssen [23]. More information regarding Fredrik Jenssen's work can be found in section 4.2.

5.2 Usability Testing

To ensure that the new application platform is well received by its intended users, it is important that the usability has been tested. There is always a risk of usability being degraded when a system is expanded. With more options and features there is a risk that users no longer feel sure or confident when using the system. Therefore, is important to test and compare the usability with the results from the previous Magno motion and form tests, to ensure that there is no significant degradation in usability.

5.2.1 Remote Testing

There are two common approaches when performing usability testing remotely, namely remote asynchronous usability testing (RAUT) and remote synchronous usability testing (RSUT). The two approaches live up to their names, as such RSUT is performed synchronously, or in other words in real-time with the tester and evaluator being separated over some distance physically. With RAUT being performed asynchronously, the tester and evaluator are separated not only in space, but also in time. Previous studies have shown that RSUT can be found to be virtually equivalent to a physical test setting in a lab [2]. RSUT can be performed using a live-video chat service such as Teams, Zoom or Skype and can otherwise be conducted as one would in a think-aloud lab setting, and screen sharing. Furthermore, all participants in the usability testing of the platform were given the option to perform the test at their workplace, home or remotely, with all participants preferring the remote option.

One drawback of RAUT is that it is less effective in identifying usability issues when compared with a lab based setting [1]. However, it is also far less time consuming to perform as the evaluator does not have to be present. This allows the usability tests to be conducted with far more participants, leading to a much larger sample size of results. In addition, given enough participants, it is very likely that all usability issues would be identified. Given that there are a limited amount of relevant participants (e.g. teachers or special educators) available for testing during this project, and that there are time constraints for development, a combination of methods will be used. The design was tested with a select few participants in a physical setting to discover any large errors in the design that would prevent participants from being able to perform the tests themselves remotely. The remainder of participants performed remote asynchronous tests and filled out usability questionnaires afterwards. The participants performing the usability test early in the physical setting did not fill out questionnaires, as there could be changes to the design that could impact the usability score. However, they were asked to think aloud during the test, and asked for feedback after the test was concluded.

The platform was tested using RSUT only. The participants were given the option of performing the test at their workplace, home or remotely, with all participants preferring the remote option. The remote testing was performed using Zoom and screen sharing with voice recordings. Furthermore, the usability testing of the platform did not include any participants that had already tested the design, to prevent familiarity with the design impacting the results. Finally at the end of the RSUT tests, the participants filled out two questionnaires anonymously, one which measures usability, and another which measures technology acceptance.

5.2.2 SUS Form

A form that is often used to measure usability is the System Usability Scale (SUS) form. John Brooke created the SUS form in the 80s as a way to quickly test the perceived usability of a computer system. The SUS form has since become very common in usability testing [6]. Further observations and analysis performed by Jeff Sauro in 2011 concluded that SUS is a great tool for measuring the usability of software products. His analysis showed that SUS results are both reliable and valid, with the form detecting differences at smaller sample sizes compared to other questionnaires [47].

The SUS form consists of ten questions regarding usability. These are answered with a score ranging from 1 to 5, with 1 corresponding to strong disagreement, and 5 to strong agreement. Each question is either positively or negatively worded with odd numbered questions being positively worded, and even numbered questions being negatively worded. This structure is designed to avoid biases in the response, such as agreeing to all items in the questionnaire.

The questionnaire which will be used during this project has been translated to Norwegian by Dag Svanæs, a professor at NTNU specializing in the fields of human-computer interaction and interaction design. This is due to all participants of the usability tests being Norwegian and that text in the system will be worded in Norwegian. The complete form is as follows:

1. Jeg kunne tenke meg å bruke dette systemet ofte.
2. Jeg synes systemet var unødvendig komplisert.
3. Jeg synes systemet var lett å bruke.
4. Jeg tror jeg vil måtte trenge hjelp fra en person med teknisk kunnskap for å kunne bruke dette systemet.
5. Jeg syntes at de forskjellige delene av systemet hang godt sammen.
6. Jeg syntes det var for mye inkonsistens i systemet. (Det virket “ulogisk”)
7. Jeg vil anta at folk flest kan lære seg dette systemet veldig raskt.
8. Jeg synes systemet var veldig vanskelig å bruke.
9. Jeg følte meg sikker da jeg brukte systemet.

10. Jeg trenger å lære meg mye før jeg kan komme i gang med å bruke dette systemet på egen hånd.

The standard SUS form worded in english can be found in Appendix A

5.2.3 Technology Acceptance Form

The technology acceptance form is based on questions from the Technology Acceptance Model (TAM). It is suggested in the model that a number of factors influence a person's decision about how and when to make use of a new technology. The original model suggested that the following two factors heavily influenced a person's decision [7].

- Perceived usefulness (PU), defined as *the degree to which a person believes that using a particular system would enhance their job performance*
- Perceived ease-of-use (PEOU), defined as *the degree to which a person believes that using a particular system would be free from effort*

TAM has been part of multiple studies and expanded upon several times, and the questions that will be used in this project are based on Shang Gao's study and extension of TAM towards mobile services. Gao attempted to optimize the model towards mobile services. This process also included the removal of questions that could be perceived as ambiguous or redundant [15]. The final form used in this project consisted of four questions, one measuring a person's Intention to Use (IU) and three questions measuring Perceived Usefulness (PU). The questions used were decided on based on feedback from the thesis supervisor, John Krogstie. Furthermore, the questions were translated into Norwegian, so as to match both the language in the SUS form, and the system.

1. Gitt at jeg har tilgang til systemet, ser jeg for meg at jeg ville brukt det.
2. Systemet ville vært nyttig for meg i min jobb.
3. Systemet ville gjort det enklere å følge opp barn med lese- og skrivevansker.
4. Jeg ville kunne spare tid ved å bruke systemet.

5.2.4 Participant recruitment

Participants for the usability testing were recruited using a combination of purposive and snowball sampling. The initial participants were drawn from my professional network, who then recommended other participants with a similar background, e.g. teachers or special educators. Participants were contacted through SMS text messages and e-mail which explained why they were being contacted and who recommended them for participation. After agreeing to participate, a time and/or place or method for performing the usability test was agreed upon and scheduled. In total 20 individuals were contacted, with two participants agreeing to a physical test of the design, nine agreeing to remote asynchronous testing of the design, and an additional nine performing the remote synchronous test of the platform.

5.3 Frameworks, Tools and Technology

The following sections provides introductions to frameworks, tools and technologies that have been used as part of this project.

5.3.1 React

React is a front-end framework that makes the process of realizing user interfaces easier. It is declarative making code more predictable, easier to understand and simplifies debugging. In addition, it is component-based, allowing for the creation of separate encapsulated components that manage their own state, which can then be composed into complex UIs [44][45]. React was chosen not only to ensure the efficiency and quality of the finished system, but also due to its popularity, with a recent survey placing it as the most-used web framework among developers [56]. The popularity increases the likelihood that developers working on the project in the future are familiar with the framework.

5.3.2 Material UI

Material-UI is a comprehensive library of components that features an implementation of Google's Material [33]. Material is an adaptable system of guidelines, components, and tools created by Google to support the best practices of user interface design [29]. As all the teachers and special educators in the local region use Google's tool for work, it makes sense to incorporate the design system Google uses to take advantage of the users familiarity with it. In addition, it allows for the use of a comprehensive collection of prebuilt components that are production ready.

5.3.3 Typescript

TypeScript is a programming language that can be transpiled into JavaScript and, as such it is also compatible with React. With its object-oriented programming capabilities, it provides JavaScript with inheritance, classes, strong static typing, and pre-compilation error detection [55].

5.3.4 Node.js

Node.js is a JavaScript runtime environment used for server-side and web applications. It uses the Google V8 engine to execute JavaScript code, allowing JavaScript programs to run on servers. Additionally, Node.js comes with the Node Package Manager (NPM), which keeps track of dependencies and module versions. NPM contains a registry of JavaScript packages for fixing a variety of problems, and it can automatically install them by running an easy to use NPM install command [35].

5.3.5 Express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web applications. It offers a myriad of HTTP utility methods and middleware, allowing quick and easy creation of robust APIs. The framework itself is very lightweight, so a large part of its benefit and potential comes from third party libraries and features, which it can easily access using Node.js and NPM [11] as mentioned in section 5.3.4.

5.3.6 Azure

Microsoft Azure is a public cloud computing platform. The platform offers many different services, including cloud computing, analytics, storage, and networking. These services enable developers and application operators to build new applications or scale existing ones on the public cloud. Additionally, it is commonly used to host databases in the cloud, where it offers serverless relational databases and non-relational databases [30].

5.3.7 Azure Cosmos Database

Azure Cosmos DB is a database system developed by Microsoft, it was developed to address problems their developers were facing with large internet-scale applications. It was officially launched in 2017, and supported capabilities such as global distribution, and horizontal scaling with low latency and high throughput. Furthermore it supports several different data models, and APIs for accessing data, giving developers the freedom to work with data in the form that best fits it [40].

5.3.8 Git

Git is a distributed version control system that tracks code changes and facilitates software collaboration. The git workflow facilitates the development process by allowing developers to work on different features on different branches while maintaining a functional master branch. In this project, it will be used to aid in maintaining an organized code structure.

5.3.9 Github

GitHub is a website that offers cloud-based Git repository hosting. As a result of using a GitHub repository, software development can be collaboratively done more easily since all developers have access to the entire code base and its history. Other features of GitHub include access control, feature requests, bug tracking, task management, continuous integration, and web hosting [16].

5.3.10 Heroku

Heroku is a container-based cloud Platform as a Service (PaaS). It is used to deploy, manage, and scale modern apps. Heroku also offers additional services including metrics, a dashboard for easily scaling deployments, and database hosting. The service supports a multitude of different programming languages, and allows easy deployment by using simple git commands [19].

5.3.11 Visual Studio Code

One of the most popular integrated development environments (IDE), Visual Studio Code provides syntax highlighting for hundreds of languages, debugging, code completion, refactoring, and integration with Git, as well as community-made extensions. Additionally, it is known for its simplicity, speed, and high degree of customizability [32].

5.3.12 Figma

Figma is a web-based app for editing graphics and designing user interfaces. It can be used for all kinds of graphic design work such as wireframing websites, designing mobile app interfaces, and prototyping designs. Furthermore, it works directly in the browser, allowing projects to be accessed from any computer without having to install software, and allows real-time collaboration between designers [14].

5.3.13 Feide

Feide is the Norwegian national solution for secure login functionality and data sharing within the education and research domains. Feide gives students, researchers and educators safe and correct access to a variety of different digital services with a single username and password. Furthermore, it enables secure and simple sharing of data for administrations, while ensuring that user privacy is protected [12]. During the initial design and planning phases, Feide was planned to be implemented to provide authentication, however, due to stringent demands and the possibility of a long application process for access, it ultimately ended up not being used.

float

6 Design and Implementation

This chapter details the design, development and implementation process and provides an overview of the different views in the platform.

6.1 Software Architecture

The architecture of the platform was inspired by the architecture found in the Svipgerð.is platform as described in subsection 4.2. This made sense seeing as the existing Magno web applications were already designed and created specifically to work with this architecture. The following sections give further details regarding the specifics of the software architecture used for the front-end and back-end.

6.1.1 Client-Server

The entire application follows a client-server architecture, this means that the code is split into a client and a server. Users interact with the graphical user interface provided by the client. The client acts as a consumer and does not do any form of calculation or treatment of data. User interaction leads to client requests being sent to the server which acts as a producer. To communicate these requests to the server, representational state transfer application programming interfaces (REST APIs) are used. When the server receives the requests it can produce data based on the parameters sent from the client.

6.1.2 Front-end Architecture

The front-end architecture follows the Model-View-Controller (MVC) and observer patterns. The model is split into separate data-model stores with distinct responsibilities which handle all data that is either shared between components, or needs to be non-volatile. The remaining data is handled internally by components, these only handle data which is not required by any other components. To detect changes to the data-models, the components follow an observer pattern, allowing them to easily detect changes, and update the view if necessary. Following these patterns significantly reduced the number of callback functions and properties being passed between components, thus reducing the overall complexity of the system.

6.2 Back-end Architecture

6.2.1 REST-API

The REST design pattern provides a simple and easy to use interface between the client and the server. When the client wants some operation done it simply sends a request through the API. There are five mandatory constraints that the API satisfies which makes it RESTful [13].

When a client sends a request it provides a resource identifier. Following this the server responds with enough information to allow the client to modify its resource. This is commonly information about students, allowing the client to display the students and their results. All requests sent to the server provide enough information to perform the requested action, and the server responds with all the information necessary for the client to handle it. These put together means that the requests look the same on all possible systems, and that the interface is uniform.

Following this, are the remaining constraints.

- Client–server: The application is divided into a client and server.
- Stateless: The server does not maintain a state of the user, nor does it remember what requests are sent or what resources have been consumed.
- Cacheable: Upon sending data to the client, the server informs the client whether or not it can be cached.
- Layered system: The client doesn't care how the server is constructed, and the requests are never affected by the state or layers of the server.

6.2.2 Modules

The back-end is split up into separate modules with distinct responsibilities. These are the router, the API, and the controllers. The router handles the initial request, and displays the correct web application serving either the application platform, or one of the three tests available in the system. The API handles the remaining requests, receiving them, and directing them to the correct controller, which in turn handles the actual request and response. The back-end consists of several modules, as seen in Figure 16.

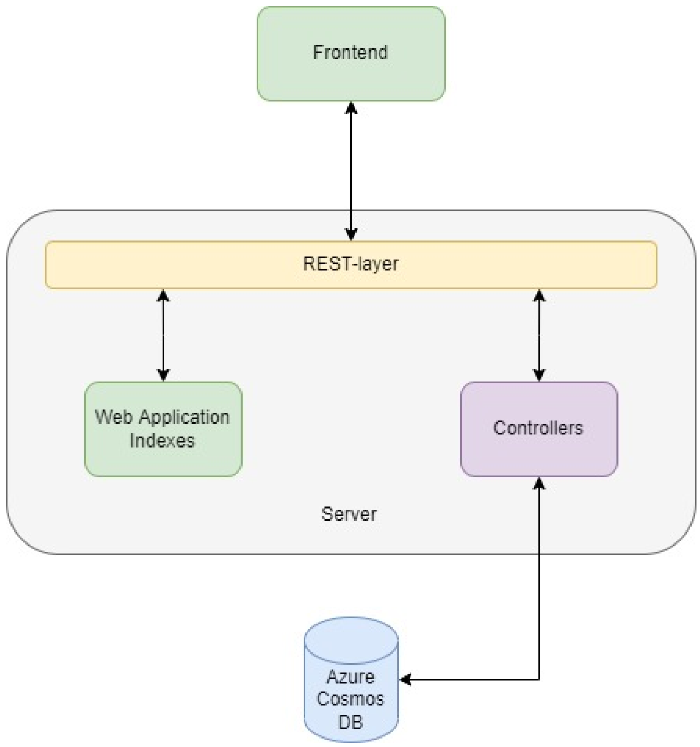


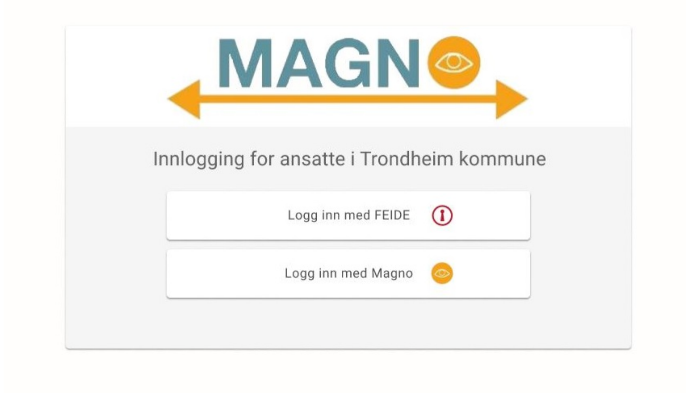
Figure 16: The figure shows the communication between the front-end and the back-end.

6.3 First Iteration: Design

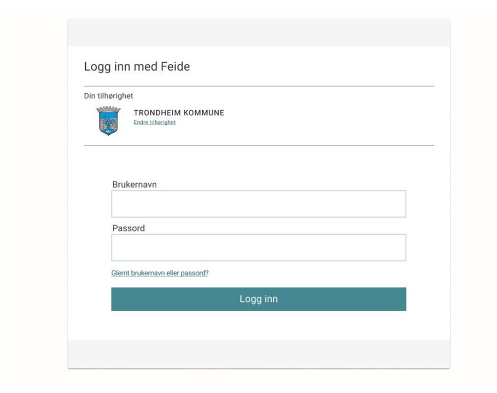
The aim of the first iteration was to create a design for the application platform. This would include designs for all the necessary screens, which included login-, home-, student overview-, tests-, and single student screens. The design was created using Figma, and was based on feedback received from teachers and special educators in the specialization project.

6.3.1 Design Overview

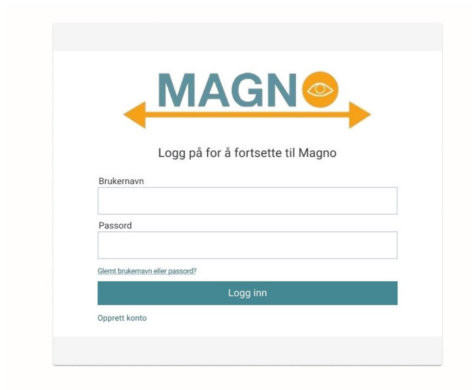
This section shows the screens in the design, they will be shown in order, from the first screen to the last one. The design did not include the actual tests, as changes to these were considered to be out of scope for this project. The overall design has kept many of the colors found in the Magno logo, and adopted many of Google's own design guidelines, and drew inspiration from the Google user dashboard [17]. The screenshots have been taken on a 27" WQHD monitor and cropped to fit this document, so the contents do not necessarily represent how the screens look when using Figma for testing.



(a) Login choice screen



(b) Feide login option



(c) Magno login option

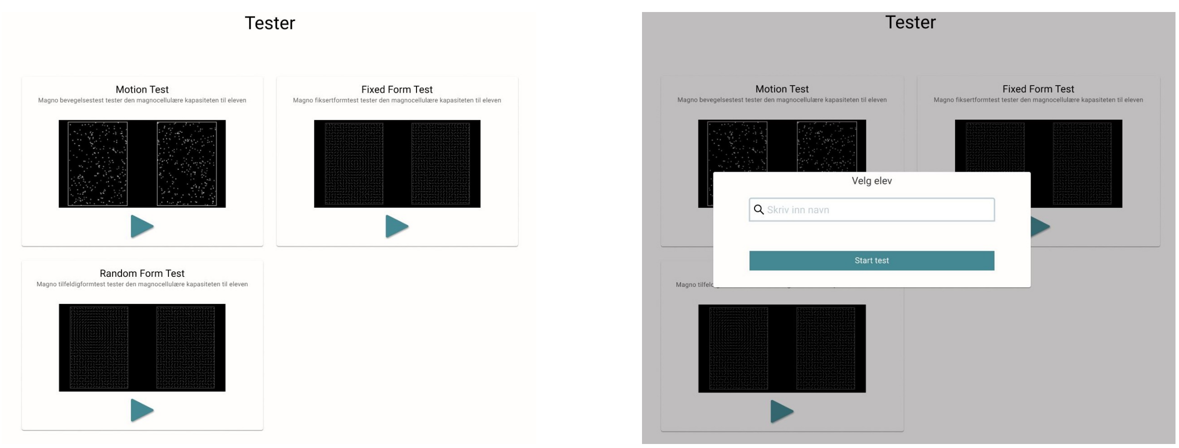
Figure 17: Design: login options screens in the design

Figure 17 shows the login screens. Subfigure (a) shows the first screen visible in the design, giving users an option to login with Feide, or to login with Magno. In the test of the design, users did not need to enter any username or register an account to gain further access to the system. The figure also shows cropped screenshots, and are missing details regarding the possibility of going back, an uncropped screenshot is visible in Figure 18 where both the sidebar and browser navigation bar is visible.

Figure 18 shows the home screen, which is the first screen shown after a successful login. It has the Magno logo on top, followed by 3 cards, giving users quick access to the main features. In addition there is a navigation bar on the left hand side, with "Hjem" being active. The user now has the possibility to either press "Start en test", which navigates users to the test screen, or press "Elevoversikt" which navigates the user to the student overview.

Figure 18: Design: uncropped version of the home screen

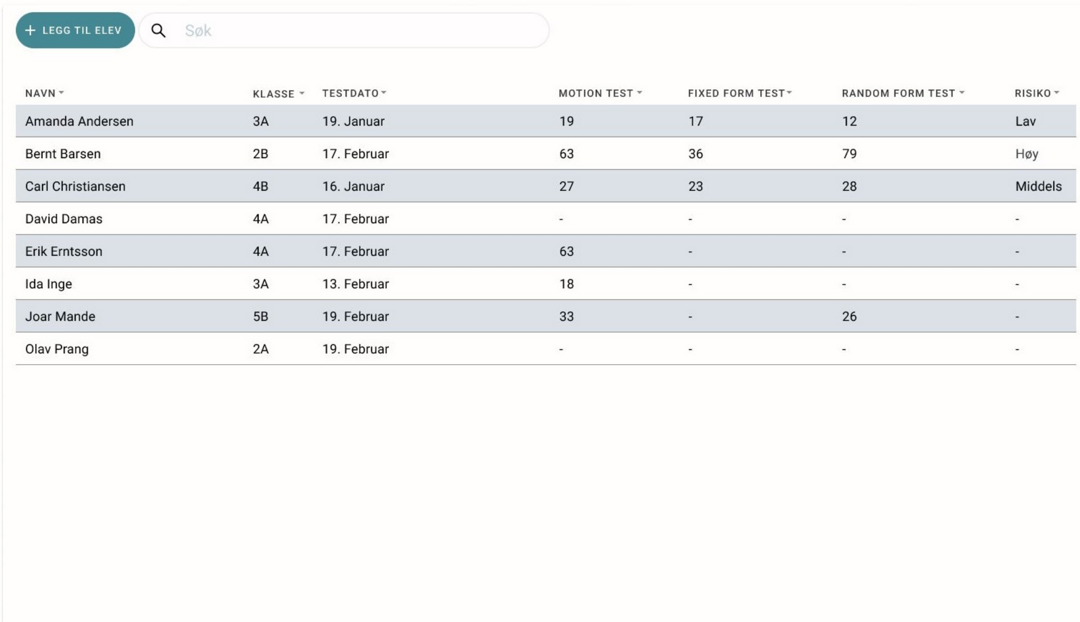
In addition, the user can press "Siste resultater" which will navigate to the student overview, but sort the student list by the latest test results. Or, the user can skip all of the mentioned options, and just navigate using the navigation bar on the left hand side.



(a) Tests screen (b) Tests screen after pressing a test

Figure 19: Design: tests screen with interactions

The tests screen with interactions is shown in Figure 19. The navigation bar and the browser bar as shown in Figure 18 are not visible due to cropping, this also skewers the perspective of the dialog box, which is actually centered in the middle of the screen. Pressing either one of the tests brings up the same interaction, opening up a dialog box as shown in (b). Pressing the input field "Skriv inn navn" brings up a drop down list of students in alphabetical order, and allows the user to either select from the list, or use the input field as a search box, filtering the drop down list based on the input. The actual tests were not included in the Figma design, as changes to these were considered to be out of scope for this project.



NAVN	KLASSE	TESTDATO	MOTION TEST	FIXED FORM TEST	RANDOM FORM TEST	RISIKO
Amanda Andersen	3A	19. Januar	19	17	12	Lav
Bernt Barsen	2B	17. Februar	63	36	79	Høy
Carl Christiansen	4B	16. Januar	27	23	28	Middels
David Damas	4A	17. Februar	-	-	-	-
Erik Erntsson	4A	17. Februar	63	-	-	-
Ida Inge	3A	13. Februar	18	-	-	-
Joar Mande	5B	19. Februar	33	-	26	-
Olav Prang	2A	19. Februar	-	-	-	-

Figure 20: Design: student overview screen

The student overview screen is visible in Figure 20. The screen mainly consists of a table of students, showing their names, grade, last test date, risk and so on. Pressing "LEGG TIL ELEV" brings up a dialog box allowing users to add a student. Furthermore users could navigate to specific students' screens by clicking their respective rows in the table. For the purpose of testing the design, only three student screens were designed, one for each risk category, shown as "Lav", "Middels" and "Høy", which correspond to "Low", "Medium" and "High".

Figure 21 shows a student screen, here with "Høy" risk. It is in many ways similar to other risk categories, with the text differing slightly for low risk students, and the color of the warning icon being coloured either light teal, or yellow, using the same colors as the Magno logo. Furthermore the screen has many different cards, one showing the risk with a textual explanation and recommendation, three cards showing the results for the different tests, and lastly three cards for initiating tests directly from the student's screen. In addition, there are three small buttons in the top right corner, one for navigating back, one for options, allowing changing the grade and/or name of the student, and lastly one for deleting the student from

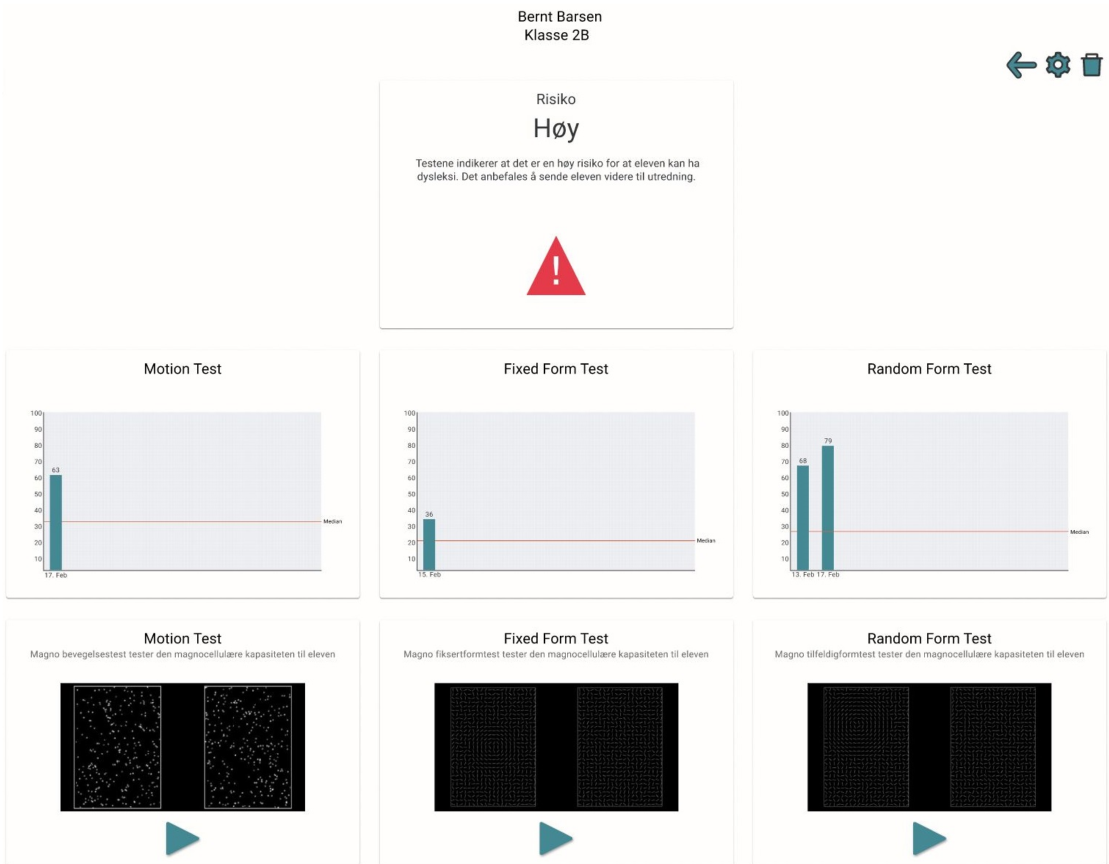


Figure 21: Design: selected student screen

the system.

6.4 Second Iteration: Implementation

The main goal of the second iteration was to implement the application platform, with design changes based on feedback from usability testing of the design. In addition to developing the front-end however, the back-end system and database also had to be developed. The system would have to be able to support multiple different web applications using different technologies and frameworks. This section will provide an overview of the implementation, with descriptions of notable design changes, and a description of how the backend was implemented to support the different web applications.

6.4.1 Notable design changes

One of the most notable design changes, was the removal of the Feide login option. With this removed, there was no need to have a login-choice screen, as such the landing page has been set to the Magno login screen as shown in Figure 22. Furthermore, in the selected student screen the three tests have been removed, this was due to feedback from testing of the design which indicated that it was confusing. In addition, an information page has been added to give users information regarding how the tests work. Finally, the risk text in the student overview screen has been color coded according to the risk.

6.4.2 Front-end Application Overview

This section shows the screens of the front-end system, they will be shown in order, from the first screen to the last one. The screenshots have been taken on a 27" WQHD monitor and cropped to fit this document, so the contents do not necessarily represent how the systems looks in use. Figure 23 shows an image that has not been cropped, and the entire user interface with navigation options is visible.

Figure 22 (a) shows the first landing page of the system. This is the login screen, allowing a user with a registered account to login by entering their email, password and clicking login. If they do not have an account, they can switch to register an account by clicking "Opprett konto" which navigates to the register screen which is shown in Figure 22 (b). A user registers an account by entering their email address, desired password and selecting the school that they work on. The school input field works as a search box, and also has a dropdown-list which is filtered based on the input.

(a) Platform: login screen

(b) Platform: register account screen

Figure 22: Login and register screens of the implemented platform

Figure 23: Platform: uncropped home screen

Upon a successful login, users are navigated to the home screen which is shown in Figure 23. The figure is uncropped to show the browser bar and the user interface navigation bar on the left hand side. This navigation bar is visible in all other parts of the application platform. The home screen allows easy navigation with textual information. "Start en test" navigates to the tests screen, "Elevoversikt" navigates to the student overview, and finally "Siste resultater" navigates to the student overview, with the table of students sorted by last test date. There are some minor changes from the design, mainly the addition of a welcome message to the user, indicating that they have logged in, and the icons have been changed due to not being available in the material ui package.

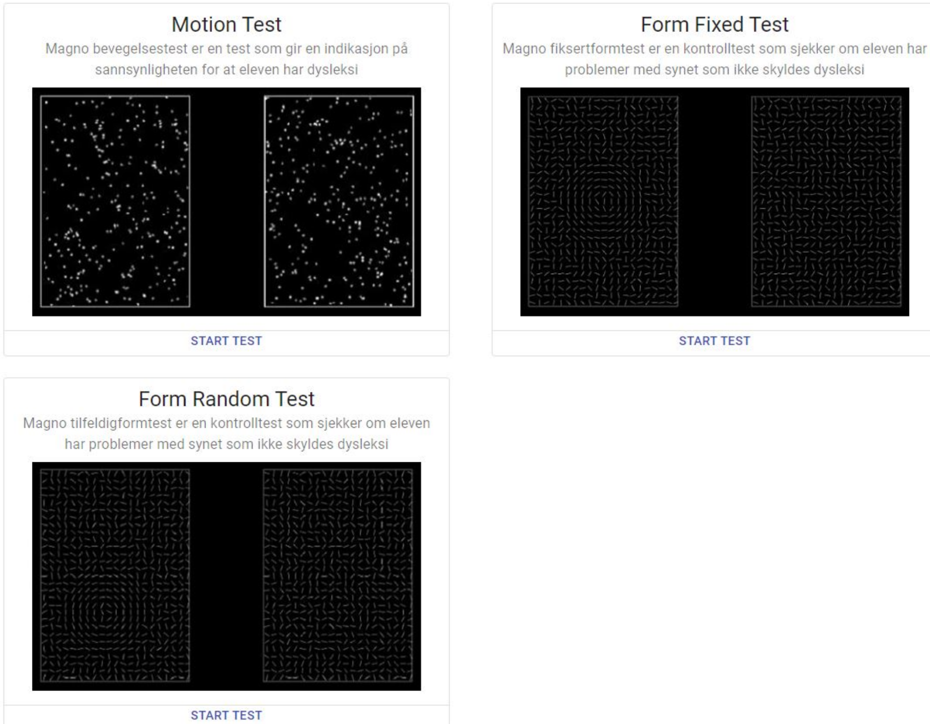


Figure 24: Platform: tests screen

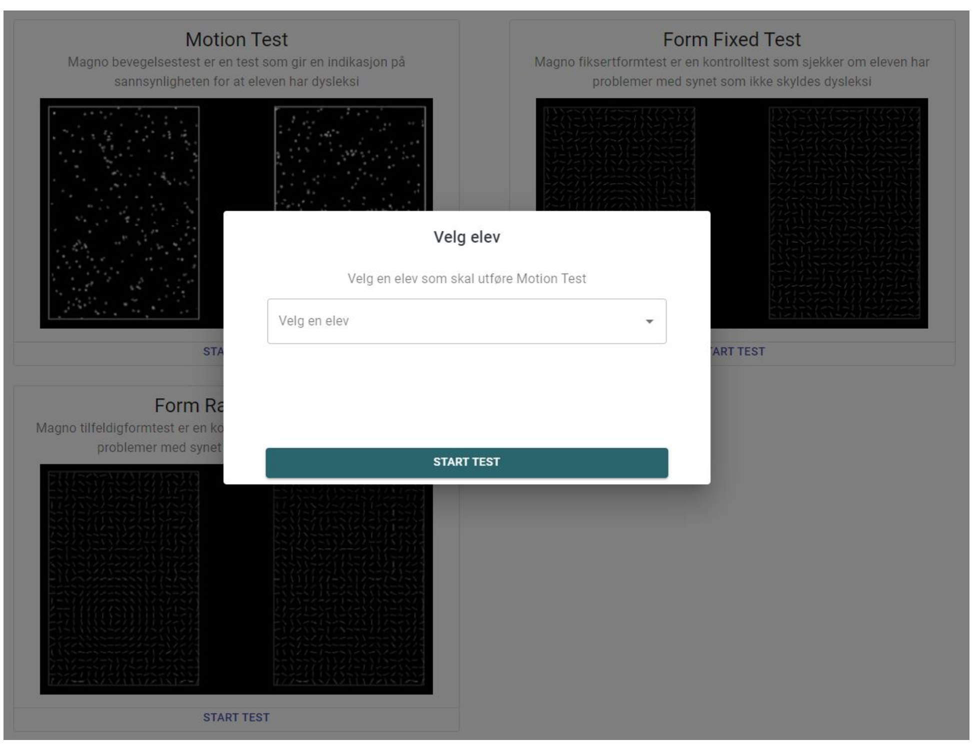
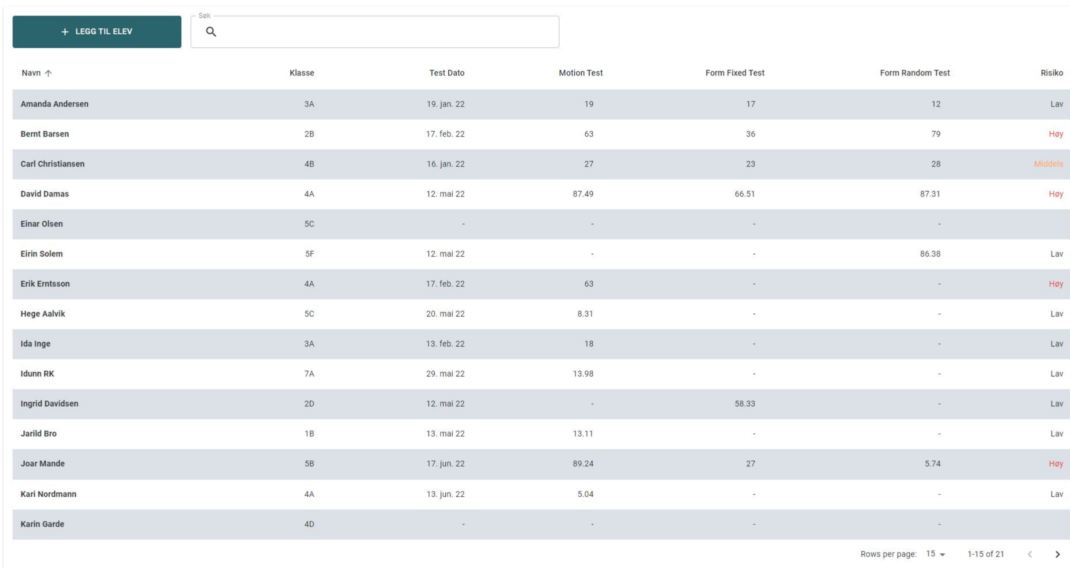


Figure 25: Platform: tests screen with dialogbox

The tests screen is shown in Figure 24 and Figure 25 shows the tests screen after clicking the motion test. In this screen there were several small changes compared to the original design. A hover effect was added for all tests, these inform the user that you get an option to select a student before the test starts. Furthermore the texts for each test has been changed to inform users more accurately regarding what each test is used for, specifically that the form tests are used as controls for the motion test. The large "Play" icon was also removed, and replaced with a text that says "START TEST", this made the design of the cards fit more in line with the overall design. Upon clicking a test, a dialog box opens up where the user can select a student, this is similar to the original design, however more text has been added which was meant to inform users about which test they had clicked.



Navn	Klasse	Test Dato	Motion Test	Form Fixed Test	Form Random Test	Risiko
Amanda Andersen	3A	19. jan. 22	19	17	12	Lav
Bernt Barsen	2B	17. feb. 22	63	36	79	Høy
Carl Christiansen	4B	16. jan. 22	27	23	28	Middels
David Damas	4A	12. mai 22	87.49	66.51	87.31	Høy
Einar Olsen	5C	-	-	-	-	
Eirin Solem	5F	12. mai 22	-	-	86.38	Lav
Erik Erntsson	4A	17. feb. 22	63	-	-	Høy
Hege Aalvik	5C	20. mai 22	8.31	-	-	Lav
Ida Inge	3A	13. feb. 22	18	-	-	Lav
Idunn RK	7A	29. mai 22	13.98	-	-	Lav
Ingrid Davidsen	2D	12. mai 22	-	58.33	-	Lav
Jarild Bro	1B	13. mai 22	13.11	-	-	Lav
Joar Mande	5B	17. jun. 22	89.24	27	5.74	Høy
Kari Nordmann	4A	13. jun. 22	5.04	-	-	Lav
Karin Garde	4D	-	-	-	-	

Figure 26: Platform: student overview screen

Figure 26 and Figure 27 shows the student overview. This screen is nearly identical to the original design, with the only notable changes being the color coded risk texts, and that the search field and "LEGG TIL ELEV" button being square, instead of rounded. This was changed so that they were more in line with the overall design of the system.

There are some larger changes to the selected student's screen shown in Figure 28. The most apparent change is the removal of the option to start the tests directly from this screen, this was changed based on feedback from the usability testing, where some testers were confused regarding what the test action cards would do, and initially thought they would show a replay of the test the student had performed. The remaining minor changes are the removal of the configure, back and delete buttons in the upper right corner as these were found to be confusing during testing.

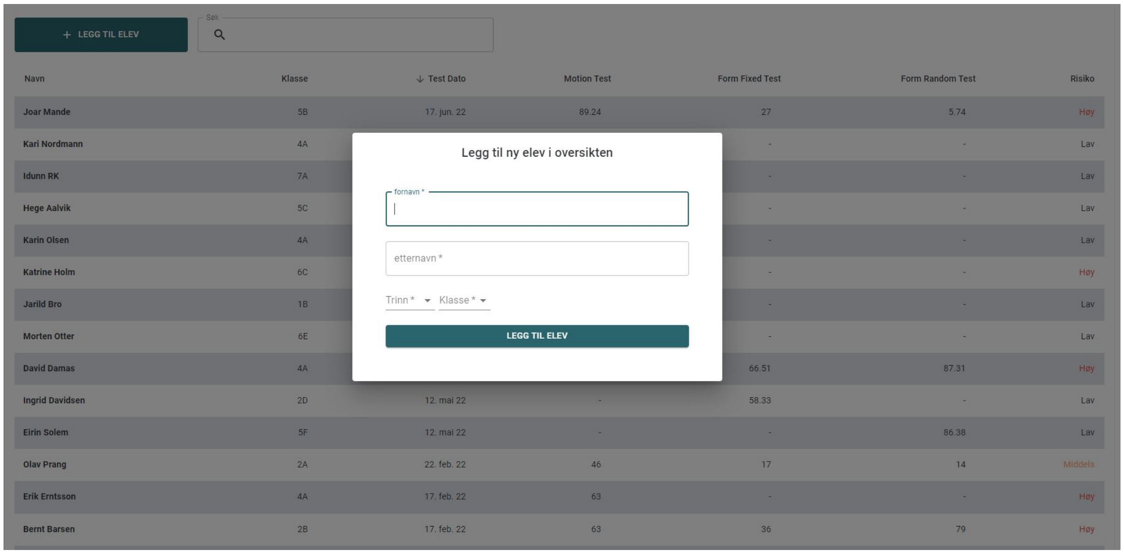


Figure 27: Platform: student overview screen with dialogbox

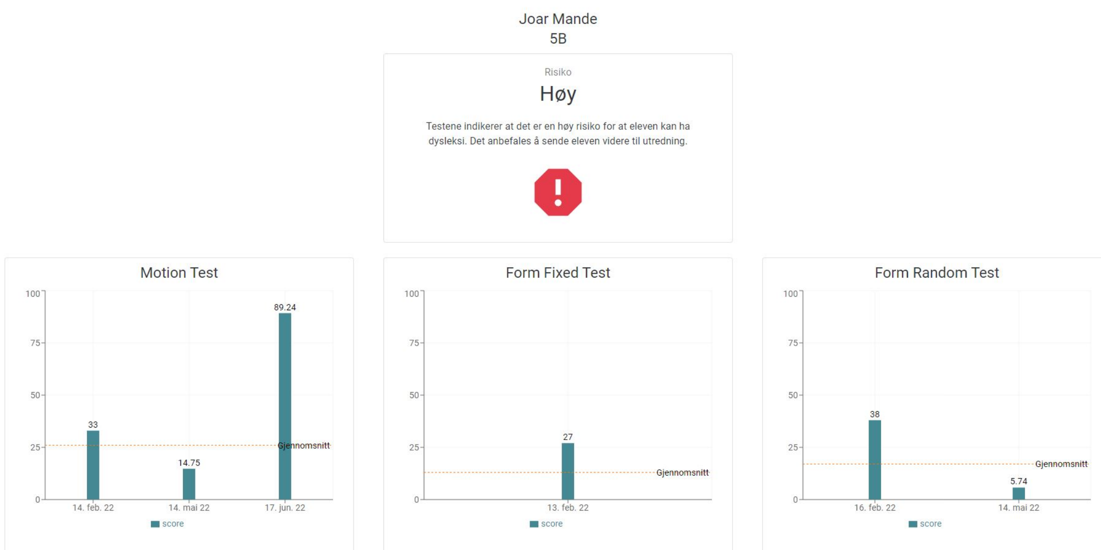


Figure 28: Platform: selected student screen

The final screen is the information screen shown in Figure 29. This screen was added based on feedback and a discussion with thesis supervisor Krogstie. The intention of the screen is to give users a quick introduction to Magno, and inform them how the tests work, and what the different results are likely to indicate.

Figure 29: Platform: information screen

6.4.3 Hosting

There were two different providers used for the database and hosting the server. Azure was used for the database and utilization of Azure allowed the database to be always online, highly scaleable, and with easy solutions for re-creating the database structure, in addition to being free for low demands. Looking into the metrics of the database, indicate that the max normalized resource consumption reached 4% with two users interacting with the system at the same time as shown in Figure 30, based on these numbers, the database should support at least 50 users within the free limits. This could be further improved upon by further optimizing caching for the more resource demanding requests.

Heroku was used for hosting the server. Heroku uses containers that they have named dynos and offer up to 1000 hours of dyno hours per month for free, which is more than adequate for testing the platform and server. Furthermore, it allows the platform and server to remain available and operational after this project has been concluded. The maximum amount of concurrent users for the free tier option has not been tested with this server, as it highly depends on the amount of stress each user puts on the dyno. However, each request sent should have a minimal impact on the server, as no websocket connections are used, and each request and response consists of small HTTP messages.

Both of the hosting options are however highly scaleable, and would likely be able to support thousands of concurrent users at low costs.

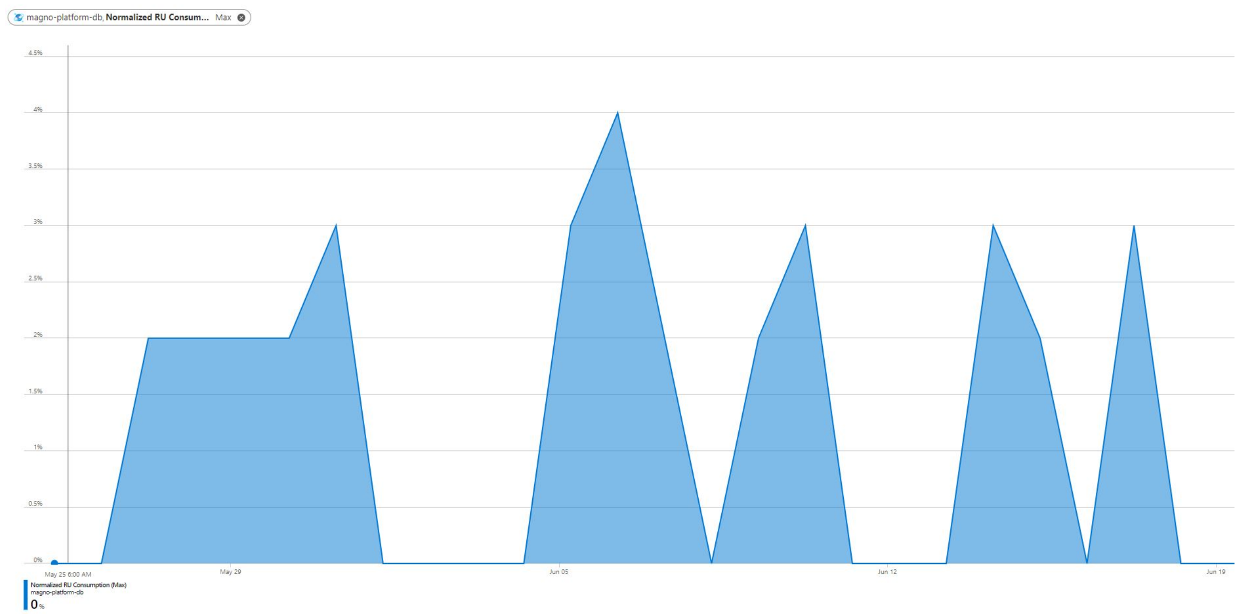


Figure 30: Database Normalized Resource Unit Consumption

6.5 Class Diagram

This section gives an overview of classes and components in the platform, Figure 31 shows a UML class diagram of the front-end system. The white colored boxes indicate that these classes/components have a support function, and are not directly visible to users. The green colored boxes indicate that these are screens/views, and are not re-usable components. The blue colored boxes indicate that these are components that can or are being re-used multiple times. Finally, the yellow colored box indicate the connection with the back-end, with the class being responsible for handling all communication.

The back-end system is much more minimalistic in its implementation. As briefly mentioned in subsubsection 6.2.2, the back-end system consists of three main parts, the router, the API and the controllers. In total, the backend consists of four modules. The initial entry point is *server.js* which acts as the router. This module imports necessary dependencies, configures the port and routes for different web applications, and starts listening to the configured port.

In total, the module has four different endpoints, one for the platform, and one for each of the three separate tests.

The second module is *platform-api.js*, this module receives requests that are passed to it by the router. If the request matches an API endpoint, the API will forward the request to one of the controllers. There are two controllers, *account-controller.js* and *student-controller.js* which handles requests. They have distinct responsibilities, with the account-controller handling anything user-account related, such as registering a new account or authenticating a user. The student-controller handles anything student related, such as retrieving all students at a specific school, adding a new student, or saving test results.

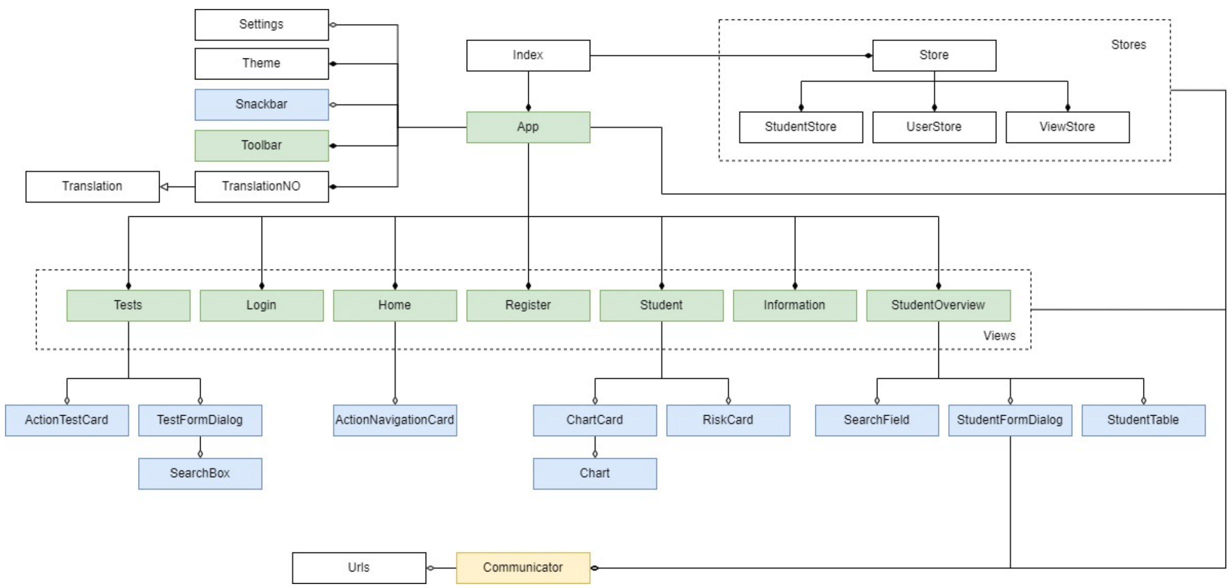


Figure 31: UML Class Diagram of the front-end platform

6.5.1 Adding New Tests

The platform and server have been designed to make integration of new tests straightforward. There are a few necessary items needed to integrate a new test, first and foremost, the tests' HTML, JavaScript and CSS files need to be placed in the server's public folder, and a route has to be added to the router in server.js, as seen in Figure 32. Furthermore, the test has to implement a function to post results to the server, with the post method needing the student id, test type and test score.

In addition, a test card has to be added to Tests.tsx in the front-end platform, these cards reuse the ActionTestCard.tsx component and only needs text to fill out the name of the test, a short description, an image of the test and the link to the test which is set when adding a route in Server.js. Finally, a ChartCard for the test has to be added to Student.tsx, which is also a reusable component. This component requires a text header, the risk score for the student, and a risk average value (e.g. a median line that depicts if the student has scored well or poorly.). The risk average value for each test is set in Settings.tsx. Below is a numbered list showing the steps necessary to integrate a new test.

1. Add HTML, JavaScript and CSS files to the servers public folder
2. Add Route to server.js
3. Add new test card in Tests.tsx
4. Add new chart card in Student.tsx
5. Add an average value to Settings.tsx (e.g. add <"test_name": 33> to the riskAverages dictionary found in Settings.tsx)
6. Push server changes to Heroku to deploy.

```
const motion_dir = `${__dirname}/public/tests/motion/`;
const form_fixed_dir = `${__dirname}/public/tests/form_fixed/`;
const form_random_dir = `${__dirname}/public/tests/form_random/`;

app.use(express.static('public'))
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cors());
app.use(cookieParser());
app.use('/', platform_api);

const port = process.env.PORT || 5000

app.get('/motion', (req, res) => {
  res.sendFile(motion_dir + "index.html", {dotfiles: "allow"});
})

app.get('/form-fixed', (req, res) => {
  res.sendFile(form_fixed_dir + "index.html", {dotfiles: "allow"});
})

app.get('/form-random', (req, res) => {
  res.sendFile(form_random_dir + "index.html", {dotfiles: "allow"});
})
```

Figure 32: Snippet of Server.js code for routes

7 Usability Testing

7.1 Usability Testing of the Design

The usability testing of the design was first performed in a physical setting with one primary school teacher, and one student who was studying to become a teacher, and who had practical field experience from placements at junior high schools. These tests were performed mainly to uncover any blatant design errors in the design, before remote asynchronous usability tests were performed. No such large issues were discovered during testing. The remote asynchronous tests were performed by nine people, with most of them having a background in pedagogy. They were first given a short description of the project and the purpose of the usability test. Afterwards they were given instructions on how navigation in Figma works. Finally, they received the link to the design, and to the questionnaire, and were informed that all answers were anonymous, and that no personal information would be recorded.

7.1.1 Usability Testing Results

SUS scores based on usability testing of the design varied greatly, with a a large number of testers giving top marks, and a few testers giving it scores below 80. Overall the design received an average SUS score of 91.67, with the median SUS score being 100, with the minimum SUS value calculated being as low as 70 as shown in the boxplot in Figure 33. Average scores of each respective SUS question is shown in Figure 34, with the questions in the questionnaire accepting a score range between 1-5.

7.1.2 Feedback and Observations

Feedback and observations from the design test was primarily received through the physical test setting. Although more feedback from the remote tests could have been gathered if an extra free-text question was included which specifically requested feedback. However, some of the remote testers gave free-text written feedback directly through email and/or SMS.

The observations and feedback received during testing uncovered a few areas where the design could be improved. Primarily, these related to the student overview screen, and the student screen. In the student overview, feedback indicated that colored risk text would aid in visibility and prioritization for teachers. In the student screen, feedback indicated there was confusion regarding the option to start tests directly from this page, with some testers

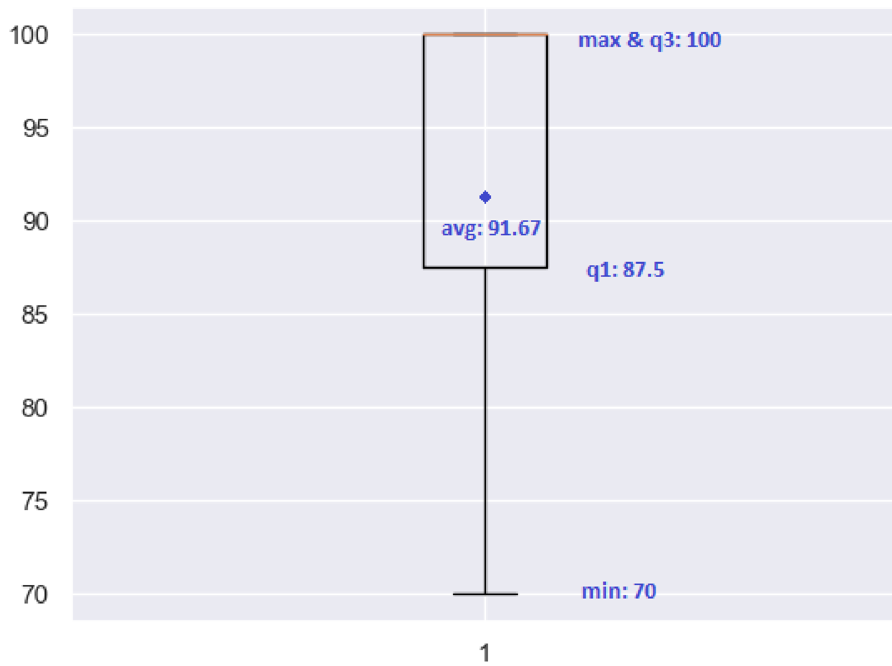


Figure 33: Boxplot of SUS scores based on usability testing of the design

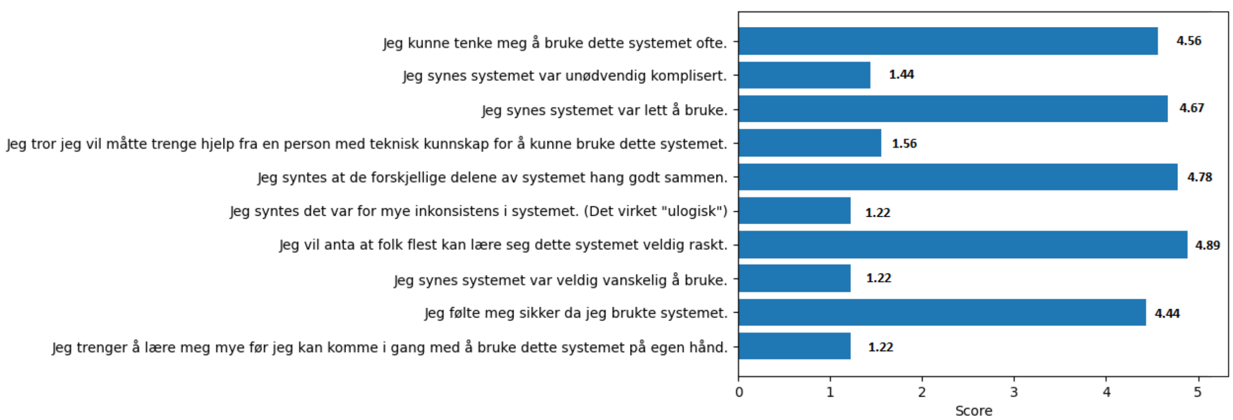


Figure 34: Average score for SUS questions based on usability testing of the design

believing that this feature would actually show them a recording or replay of the test the student had performed. In addition through observation, the buttons in the screen for going back, options for the student and deleting the student had poor placement, as they were never used for going back, and it took time for the participants to observe and find these functions.

Table 2: Proposed changes

Id	Descriptions
C1	Remove the start test cards from the student screen to reduce confusion
C2	Change icons and/or placement of the configure and delete buttons from the student screen to improve visibility
C3	Remove the back button from the student screen to reduce clutter in the user interface
C4	Add colors based on the risk level in the student table in the student overview screen, to make high risk cases more visible

7.1.3 Proposed changes

Several changes have been proposed based on the feedback and observations performed in the testing of the design. These are shown in Table 2.

7.2 Testing of the System

The system tests did not only include answers regarding the usability of the platform, but also regarding technology acceptance of the platform. As mentioned in subsubsection 5.2.1 the usability tests were performed using remote synchronous testing. The tests were all performed using Zoom with screen sharing. Furthermore all the tests were voice recorded, with a semi-structured interview performed at the end. In addition, all the participants answered two questionnaires anonymously at the end of the test, one regarding usability, and one regarding technology acceptance. In total there were nine participants, two of which worked as special educators at a primary school, with the remainder either working as teachers or studying to become teachers and having experience from placements at schools.

7.2.1 Usability Testing Results

The SUS scores from the testing of the platform were slightly lower compared to tests of the design, with the median SUS score being 90.0, with an average SUS score of 92.22 as shown in the boxplot in Figure 35. The average scores of each respective SUS question is shown in Figure 36. The questions in the SUS questionnaire accepted a range of scores from 1-5. The tasks given to teachers performing usability testing is available in Appendix B.

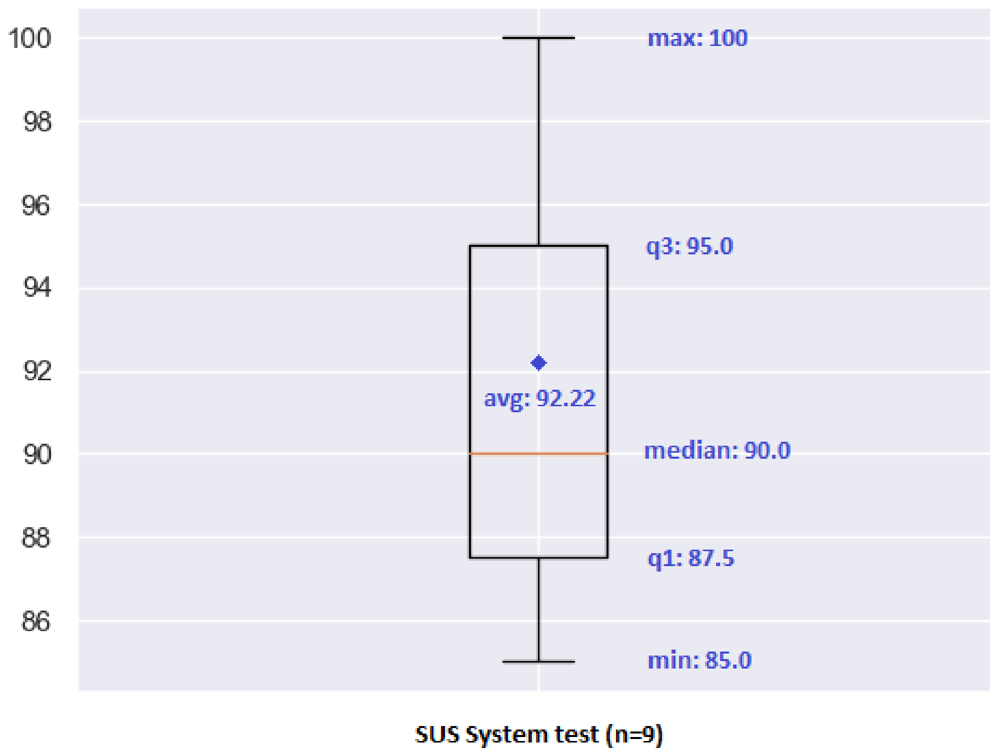


Figure 35: Boxplot of SUS scores based on usability testing of the system

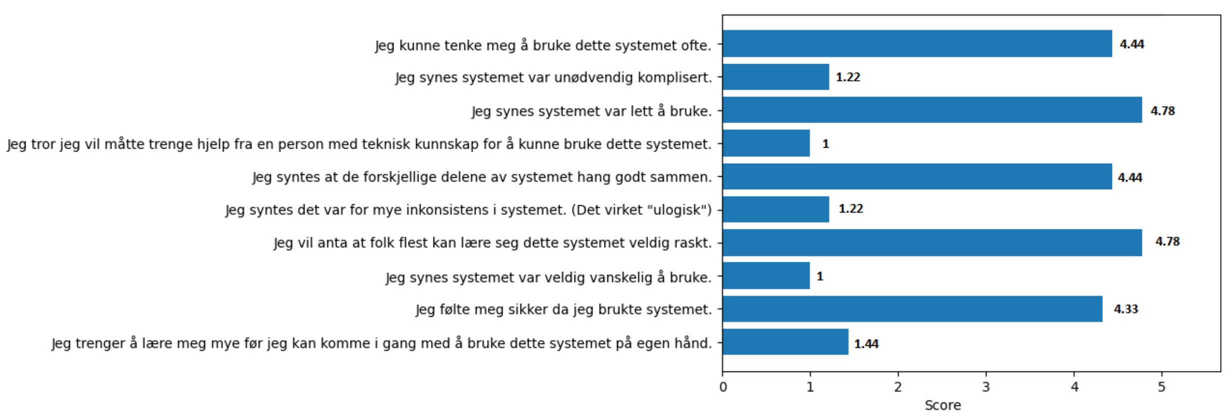


Figure 36: Average score for SUS questions based on usability testing of the system

7.2.2 Technology Acceptance

The questionnaire for the TAM questions each accepted a score range from 1-7. The average scores for each question is shown in Figure 37. However, the main findings from the TAM constructs used are the qualitative findings from the semi-structured interviews performed at the end of the testing sessions. A selection of answers given during the interviews are included in this section.

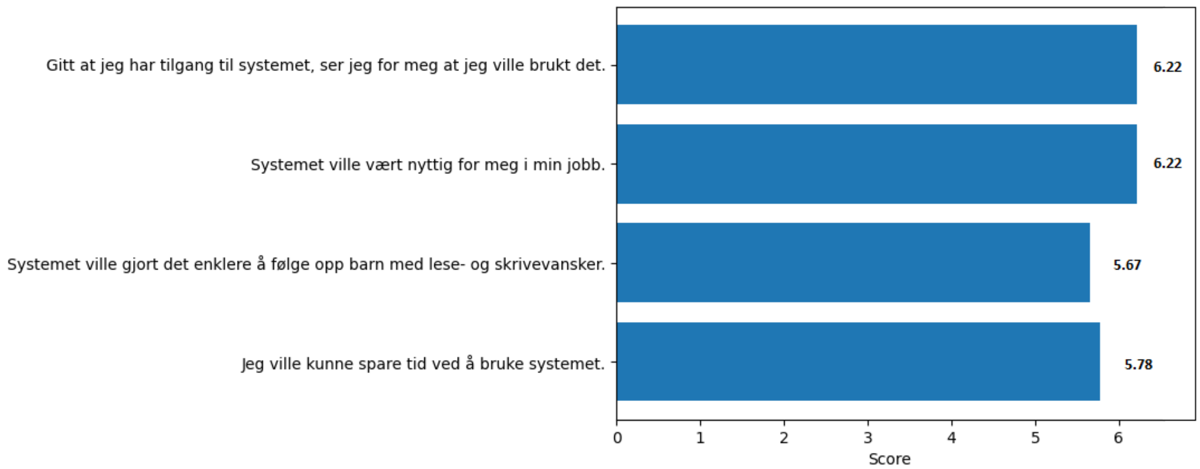


Figure 37: Average score for TAM questions

Given that I have access to the system, I imagine that I would use it
Hmm, yes I think so actually, this is part of our job. If the test results are valid and have been thoroughly tested, and if the municipality approves of the system. Then it would be very practical, it's always practical with things that are stored digitally, because everything you have in paper just adds to the workload, and I receive a lot of papers. And especially if you could include more tests in the platform, then the parts regarding comparing results, and storage would be easier. And I also wish that there would be an easy way to include the results in the student folders. For example, in Logos there's different templates for reports, and they're rather large and impractical really. My experience is that there are very few special educators that I have in my network that actually use those reports. They're so comprehensive, and the parents don't understand it. So its important that the parents also understand what we have done, what results have we received, and what are we going to do next.

Based on the answer it seems likely that the system would be of some use already. However, to increase the likelihood of use, some additional elements should be included, this being mainly a report generation functionality, that would allow teachers and special educators to print out an easy to understand and well formatted report of not only the results, but also what tests have been performed, and what other tests or remediation tactics will be performed from now on.

The system would be useful for me in my job

Yes, especially if you could store more results and have more tests in the system, and you could generate a good report, and the student page with all the results is easy to use and to understand. The more you can avoid having individual pieces of paper, and having the results consolidated in one place, the better it would be.

The administrative parts of the system seemed to be particularly interesting with regards to perceived usefulness. Multiple answers gave similar indications, with teachers and special educators wanting a system that incorporated more of the standard tests that they usually perform. As one of the issues they face are having to use multiple different systems, with tests both digital and on paper, and then manually consolidate the results and create a report on their own.

The system would make it easier to track the children's challenges with reading and writing

Well, if you could see the individual student's development, if you can map... What we usually want to do is to have different kinds of mapping tests that we perform on a yearly basis. If this was easy to administrate, so that you could perform this on all the students in a class at the same time, then you'd probably want to do it again, so that you could compare the results, so that you could visualize their development. Then you could perform a mapping, perform some form of remediation, and then perform another mapping test and compare the results.

Seeing as the system as of now only has the motion and form tests, being able to track challenges with reading and writing specifically isn't currently possible. However, the answers gave some insight into what the teachers and special educators thought of as important regarding tracking these challenges. Most importantly, if reading/writing tests are implemented it would be very important that they are easy to administrate, and that they can be used for screening a full classroom simultaneously.

Should remediation maybe have a place in this? Not necessarily that remediation strategies are generated, but having a place where we special educators can enter a remediation plan ourselves. The danger of performing mapping children is that no remediation tactics are implemented based on what the mapping tells us, and then there's no real point in performing the mapping at all. So having a place to enter for example a 6 month remediation strategy, and having a log of what has been tried earlier. Usually us special educators use these programs, not the teachers, so how do we make the results available for the teachers? This obviously concerns how the data is stored, and how it can be accessed.

The special educators noted here an issue that can occur with mapping, namely that no remediation takes place after receiving the results. So enabling them to keep track of this by adding functionality that allows registering/writing remediation tactics, and having a log of what has been tried earlier could increase the systems perceived usefulness. Especially if it could be used as a shared system for teachers and special educators, not only to test children, but also to plan and implement remediation strategies.

The more the teachers are able to perform themselves of the tests, the quicker it is for issues to actually be uncovered, and dealt with. So, that would really help us special educators in our day to day job. The children in my school are on a waiting list to be able to see special educators, so if teachers could use the system, it could really help reduce the number of students having to wait, and give us something more concrete to work with.

This answer pinpoints one of the current issues with testing in some primary schools today, namely that the teachers themselves rarely perform tests to uncover dyslexia. As such, many children end up waiting in line for long periods of time without receiving help. Being able to cut down this waiting time by having teachers perform tests could potentially allow remediation to take place at a much earlier stage. However, to be able to do this, the tests need to be easy to perform, and the teachers need clear guidance in how the tests are to be performed and administrated.

I would be able to save time by using the system

Yes, I'd say so, that's basically the reason why we want screening tests. It depends on... yes and no, new tools and things that are implemented always take time. And even so, we do a lot of different things, so in a big picture including the other work we do.. it wouldn't necessarily take less time if we add something new and remove something else. We already have other mapping tools that we would use either way, it could be time saving, seeing as it's done on a computer instead of on paper, and that you don't need to calculate scores yourself. For example, the word chain test, you need to calculate the scores yourself after the test has been performed. And if you're doing this test on 25 students.. Well, in that context it would definitely be time saving. But, it's difficult to say for sure.

The participants here noted that they don't believe the system would necessarily save time as it is, but they did see potential in it. For example, incorporation of tests performed on paper, with automatic calculation of scores would be highly time saving. This was further expanded upon with the following comment:

If more tools and tests are included in the system, then it would be really practical for us to use it. So including things like the 6 - 16 screening tests, and Våletesten are some standard tests that we always use. These include screening regarding language, task solving capabilities, memory and attention, and these do map reading, and can give some results regarding this. So, in short, the more of these we can have in one place, the better it would be. And if it's easy to administrate, and allows us to compare results over time, then that also makes the system much more attractive to use.

Many of the answers given point to a common theme, the participants have been using multiple different solutions, systems and tests, and have to manually consolidate results and scores to create a report. As such, the usefulness of the new system is not necessarily based on how the new tests it includes work, and how well they perform, but in how well it can help streamline the participants day-to-day job.

7.2.3 Feedback and Observations

Multiple insightful comments and answers were given during the testing and interview process. In addition, some observations regarding other issues were also made during the synchronous tests. There was no further development done after this iteration, so the observations made, and feedback discussed in this section addresses issues and limitations with the project in it's current state, while suggesting improvements that can be performed in the future.

Two testers noted the lack of Feide integration, and noted that this would make the system easier to use on their part:

One of the advantages of Feide integration is that I don't get shown any other students that those I have access to, for example, I don't need to know how Lisa in class 7A is doing, when I'm the contact teacher in the 3rd grade... There's also something about confidentiality there, I shouldn't know anything about other students other than my own

In addition there were some comments regarding the information page:

I think perhaps I would've preferred having information about how the tests work beneath or besides the actual tests. Maybe that would be more organized if the information was shown on the tests page. So that there isn't too much information on the information page, not that there was loads of information, but .. yeah, maybe it would seem more organized.

Several participants also noted that the scaling of the test scores felt unusual, as the test scores scale from 1-100, with 1 being the best result, and 100 being the worst, whereas they were used to the opposite:

Is it... is it possible to specify it more clearly? Often when we work with other tests the average.. if the score is below average, it's below the critical limit. So, would it maybe be possible to inverse the graph? So that the lower the score, the worse the result is?

Multiple participants also noted that it would be advantageous to group students together based on their class (e.g. 3A or 5B). This would allow users to first select the class in the student table, before all students in that class are shown.

An issue with the responsiveness of the UI was also discovered during the screen sharing process. For certain screen sizes and resolutions, the text in the cards of the home screen as shown in Figure 23 had an overflow issue, with text appearing inside the button section of the card. This issue was noticed on two separate devices, an older Acer laptop, and a HP Elite Probook. The participants were asked if they knew the screen dimensions and resolutions of the devices, but were unfortunately not aware of this information or how to retrieve this.

Finally, when starting a test, several participants opted to scroll down the student list to find the correct student, rather than using the built-in search and filter functionality. When asked about it, participants noted that now that they looked at it again, they could see that it was possible to write in the field, but they hadn't thought about doing it. This might indicate that the wording in the field should be changed. Furthermore, during observations, it was clear that the student list appearing here was not alphabetically sorted.

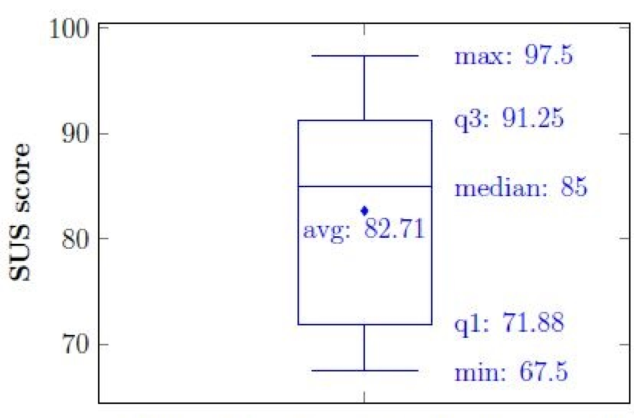
8 Evaluation

This chapter evaluates and discusses the results of the usability tests performed on the platform to see if the platform has fulfilled the requirements that were specified in Table 3.

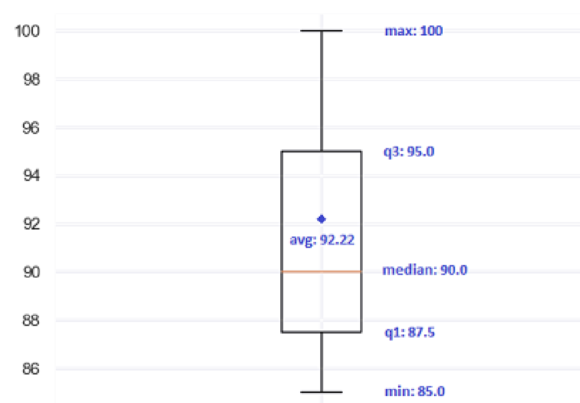
8.1 Overall Evaluation

The usability tests performed on the platform as described in subsection 7.2, was for the most part very positive. As the tests were conducted with screen sharing enabled, it was possible to observe that none of the participants had any major issues using or understanding the platform.

Furthermore it is of interest to compare the new SUS scores of the platform, with the SUS scores received in the previous Magno iteration. The previous Magno iteration as described in subsection 4.2 performed usability testing on the motion and form tests, which are a part of the newly developed platform.



(a) Box plot of SUS scores from the previous magno iteration



(b) Box plot of SUS scores for the newly developed platform

Figure 38: box plots of sus scores for the new platform and the previous Magno iteration

As can be seen in Figure 38 there has been a significant increase in the SUS score for the newly developed platform. The previous iteration received an average SUS score of 82.71, with an 85 score median, with the new platform receiving a 92.22 score average, and a 90.0 score median. This is nearly a 10 point increase in the average SUS score even though the overall complexity of the system has increased. Comparing the box plots, we can also see that the box plot shown in Figure 38 (b) is much more compact when compared to the box plot shown in Figure 38 (a). This indicates that the users were much more in agreement regarding the usability when compared with the previous iteration.

The SUS score indicates that the users find the platform very easy to both understand and use. The finished platform re-iterated on the navigational structure proposed and developed by Thea Johansen and Maja Kirkerød in the second Magno iteration [24] which is shown in Figure 12, and also made use of the logo they designed. In addition, the user interface was also inspired by Google's material design, as it was assumed that this would be easy to use and understand, seeing as the intended user group already used Google's tools in the workplace. The usability tests confirm that this assumption was correct. Other issues, and problems identified during the tests can be found in subsection 7.2. These issues were discovered through qualitative feedback and observations, whereas the SUS score was derived using quantitative feedback from the questionnaires. Even though there are still several issues, and points of improvement, the platform received a SUS score of 92.22 which is considered to be excellent.

Overall, the feedback received regarding the design was that it was not only very easy to use and understand, but that it was also pleasing to both use and look at as well. In addition, most participants already perceived the platform as being useful, and noted cases where they could imagine using it as it is. Furthermore, several participants also gave remarks regarding the possible potential in the platform as a single platform solution. So, although considered to be useful in its current state, the usefulness could be greatly improved by implementing more tests in the platform, so as to replace tests currently performed on paper, and tests performed with several different platforms and tools.

The platform and server were developed with the goal of making integration of other tests simple, and to not restrict framework and technology choices for future tests. The only known limitations are that the test has to be built and compiled into an HTML file, with corresponding JavaScript and CSS files. As such it is not currently possible to use local applications or programs which do not compile into HTML, JavaScript and CSS, such as local java programs or native mobile applications. Generally speaking, the platform and server do not restrict choices when developing new web based tests, but it does have restrictions when it comes to development of new native or local applications for mobile devices and PCs. Native applications could in theory use the API to directly fetch student information, and post results to and from the server, but they would not be truly integrated with the platform itself, and would not be able to use the platform UI to initiate the tests.

In addition, the project aimed to develop the platform in such a way that integration of new tests was simple. Currently, for a new test to be integrated there are a few things that are necessary to perform as described in subsubsection 6.5.1. As it stands, there are no large changes necessary to integrate new tests, and it is not necessary to create any new components or new logic either. And it should not be necessary for a developer to be familiar with the technologies or frameworks to be able to integrate a new test, as it should for the most part be down to copy-pasting existing code, and adding a few lines of text for headers, paths and links to make it work. In addition, thorough documentation regarding the integration of new tests with step-by-step instructions are included in the code repository. However, the difficulty will depend on the developer in question, and it is difficult to quantify and evaluate whether or not it is simple without observing and interviewing developers attempting to integrate new tests with the platform. In hindsight, integration of new tests could have been further simplified by completely separating tests from the platform, and storing the test information, headers, risk average and links in the database, and generating the tests screen and student screen based on this information.

8.2 Fulfillment of Requirements

This section provides an evaluation of the platform and server with regards to the requirements shown in subsection 4.4. The fulfillment of requirements is summarized in a table, which references the requirements by id, description and an evaluation of the requirement fulfillment, which are labeled as attained, party attained or unattained.

Table 3: Fulfillment of functional and non-functional requirements for the platform and server

Id	Descriptions	Fulfillment
FR1	It should be possible to register an account with an email address and a password	Attained
FR2	It should be possible to log in to a registered account by providing the correct email/password combination	Attained
FR3	The platform should allow users to register which school they work at	Attained
FR4	The platform should be responsive, and elements should adapt to fit the current screen size and resolution	Partly attained
FR5	The platform should be able to start motion and form tests and receive test results from these	Attained
FR6	It should be possible to add students to the platform	Attained
FR8	Users should be able to view all the students that have been added to the platform, and attend the user's school	Attained
FR9	Student test results should be viewable in a table or list	Attained
FR12	The platform should be able to display detailed results for each student	Attained
FR14	The database should store data necessary for the users	Attained
FR15	The platform should be able to communicate with the server API	Attained
FR17	The back-end API should be able to store user information in the database	Attained
FR18	The back-end API should be able to store student information in the database	Attained
FR21	The server should be able to serve both the platform and the motion and form tests	Attained
FR7	It should be possible to remove students from the platform	Unattained
FR10	The table or list showing test results should be sortable	Attained
FR11	The table or list showing test results should be searchable	Attained
FR13	The platform should be able to give text based information regarding a student's test results	Attained
FR14	It should be possible to both use and navigate the platform without any prior instructions	Attained
FR19	The platform should have Feide integration for authentication and authorization	Unattained
FR20	The platform should fetch user and student information from Feide	Unattained
NFR1	The platform should have a minimum SUS score of 80	Attained
NFR2	The server and platform should be loosely coupled to the tests	Attained
NFR3	Developers should have access to documentation which is thorough and precise.	Attained

9 Dicussion, Conclusion and Further Work

This final chapter presents a discussion around the project, in addition to a conclusion and what further work is recommended.

9.1 Discussion

The main focus of this thesis has been to develop a platform that would allow Magno to function as a dyslexia screening tool that could be used by teachers and special educators in primary schools. Usability testing was performed in two stages, the design was tested with a combination of remote asynchronous usability tests with SUS questionnaires, and monitored testing with semi-structured interviews. Following this, remote synchronous usability testing was performed on the platform, with a SUS questionnaire and a questionnaire based on certain TAM constructs being filled in afterwards, and a semi-structured interview being conducted with the participants. The main reason for the remote testing was due to requests from participants, where they all had busy schedules and preferred to perform the test remotely.

There are some limitations relating to the participant selection process. One limitation stems from three of the initial participants of the system tests being drawn on from my professional network, and it is possible that this may have impacted their responses on the SUS questionnaire. In addition, the remaining participants were selected using purposive and snowball sampling, where the initial participants recommended other people with similar backgrounds (e.g. teachers, special educators) that could participate as well. This sampling resulted in a lack of participants in the age-range of 50-66, as there was only one participant that could be within this age group. This is based on what was observed during the remote synchronous tests, as no information regarding age was collected. In addition, some of the participants were students studying to become teachers, and although they previously had been teaching at schools during their placements, they may still have lacked the experience of being the contact teacher for a class at schools. As such, they may not have been fully familiar with how contact teachers follow up children with dyslexia, and how they are tested.

Although the semi-structured interviews revealed several points of improvement, there was a limitation with regards to the information retrieved. As the interview questions were mainly centered around the TAM constructs, and observations made during the test process, they did not receive any specific information regarding how the motion and form tests themselves were perceived, and if there were any changes necessary for these to be used. As such, some important insights regarding the tests themselves may have been lost due to a narrow focus on the new platform in the interview process.

9.2 Conclusion

In summary, the new platform was well received by teachers and special educators, but there are still several recommended improvements and changes that should be implemented to increase the usefulness of the platform. The SUS score increased from an average of 82.7 to 92.2 when compared with the results of the previous Magno iteration. This score meets the requirements set out in the non-functional requirement of having a minimum SUS score of 80, and it is considered an excellent rating. In addition, the tests are loosely coupled with the platform, and do not restrict the usage of frameworks or technologies when it comes to web applications. It was difficult to come to a conclusion regarding if integration of new tests is simple or not, however, several steps were taken to increase the modifiability of the platform, such as having a high degree of modularity, and loose coupling with the tests. Several research questions were also created to aid in guiding the research and development in the right direction, these will now be reviewed to determine if they have been adequately answered or not.

RQ-1 *How can the existing Magno tests be integrated into an application platform?*
The existing Magno tests were integrated into the application platform by utilizing both the architecture of the platform they were intended for, and the existing functionality for sending results to a server. The Svipgerð.is architecture used an express server for hosting both the platform, and the tests, and this architecture was re-iterated upon. This allowed communication between the test and platform to easily be achieved through the usage of a cookie containing the id of the student performing the test. Other elements that were needed were a way to initiate the tests, this was achieved through the usage of a router on the server side, and a link on the platform. The link opens a new browser tab and sends a request to the url, and the server responds with the corresponding test using the router.

RQ-2 *How should the platform be developed to not restrict framework and technology choices for future tests?*
One of the key points in not restricting framework and technology choices was to have the tests be loosely coupled. In this case, that meant that the test shouldn't care or need to know what technologies or frameworks the server or platform uses, and vice versa. As such, there are

no direct couplings between the test and the server and platform. The communication takes place through an API, and through the usage of cookies. It is also possible to communicate with the server and have results recorded and shown in the platform by only utilizing the server API to fetch student information, and send test results. This allows native mobile applications and desktop programs to be used with the platform as well. However, this has not been tested and could impact the usability of the platform, as its currently not possible to create a link that starts a native mobile application or desktop program. As such, the tests screen would not contain a card which allows a user to initiate the test.

RQ-2.1 *How should the platform be developed to make integration of new tests simple?*
The platform was developed with a high degree of modularity, with separate components and modules to simplify integration of new tests. This included trying to reduce the number of steps and changes necessary to successfully integrate a test. This allows tests to be integrated in the back-end simply by adding the test's HTML, JavaScript and CSS files to the public folder, and adding a route. To simplify integration in the front-end re-usable components were developed. As such, the changes required for integration is the use of re-usable components, and setting custom textual headers and links for the components, and adding a risk average in a settings file.

RQ-3 *How should the new platform be designed and developed for it to be well accepted by teachers and special educators?*
For the platform to be well accepted by teachers and special educators it was important to include them early on in the design process. This allowed for more continuous feedback regarding the design and functionality, which in turn improved the overall usability and acceptance of the platform. In addition, the usage of TAM constructs for perceived usefulness, and intention to use allowed for very valuable feedback regarding the usefulness of the platform, which directly impacts how well the platform is accepted by teachers and special educators. In addition, the utilization of two separate project iterations allowed for feedback and observations to influence the development of the platform itself.

RQ-3.1 *How does the new administrative platform impact the overall usability?*
The new administrative platform had a positive impact on the overall usability. Through the usability test performed on the platform we see that the SUS score increased from an average of 82.7 for the separate existing Magno tests, to 92.2 for the administrative platform with the tests.

9.3 Further Work

This final section will discuss recommended future work with the project, proposing concrete steps which should be taken going forward.

The previous iteration of Magno did not have time to validate the tests after implementation. As such the Magno motion and form tests must be validated against the original Magno Java application before they are used in a school-setting. This would include recruitment of test subjects with separate levels of reading ability: subjects with low reading competence and another group with high reading competence. These results should be compared to previous research on the Java app, this is done to confirm that those with low reading competence perform worse than those with high reading competence with the new tests.

Furthermore, there are several security aspects that need to be addressed, this mainly pertains to authorization. This would need to include a white-listing of e-mail domains that would be allowed to register on the platform, and an e-mail confirmation to verify that the e-mail address exists, and that the user has access to it. This could also be achieved through Feide integration. In addition, several of the endpoints are accessible without authorization, and an authorization scheme for these should be implemented, this could be done using cookie based authorization, or through other means.

The remaining proposals are summarized below.

- Further improve responsiveness of the UI, and validate this with multiple devices to prevent uncommon devices from having overflow issues with cards in the home screen.
- Add report generation functionality, allowing the student results to be printed out, allowing special educators to incorporate the tests and results in their normal workflow.
- The scoring of the tests should be inverted to conform with existing tests. E.g. a result of 11 now, should equal a result of 89.
- The tests should be adapted for use with young children, as Klevstuen worked on in her specialization project. The list of derived requirements are included in Appendix C.
- The student table visible in the StudentOverview screen should first show classes (e.g. 3A, 5B), and upon selecting a class show all students in that class.
- The Student screen should include a subscreen or tab which allows for remediation tactics to be logged and entered.

- Functionality for deleting a student from the system should be added.
- The platform should be integrated with Feide, this would allow all authentication and authorization to be performed using Feide, and also allow the system to fetch student information directly from their systems. This would also improve on confidentiality, with teachers only being able to view students that they are responsible for.
- More tests and tools such as Orddelingstesten should be developed and integrated with the platform to improve the usefulness.

Bibliography

[1] AHMED S Alghamdi, ALIH Al-Badi, ROOBAEA Alroobaea, and PAMJ Mayhew. A comparative study of synchronous and asynchronous remote usability testing methods. *International Review of Basic and Applied Sciences*, 1(3):61–97, 2013.

[2] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. What happened to remote usability testing? an empirical study of three methods. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1405–1414, 2007.

[3] Anette Andresen and May-Britt Monsrud. Assessment of dyslexia – why, when, and with what? *Scandinavian Journal of Educational Research*, 0(0):1–13, 2021. doi: 10.1080/00313831.2021.1958373. URL `https://doi.org/10.1080/00313831.2021.1958373`.

[4] International Dyslexia Association. Definition of dyslexia. URL `https://dyslexiaida.org/definition-of-dyslexia/`. accessed: 2021-11-13.

[5] BDA. Identifying and teaching children and young people with dyslexia and literacy difficulties. URL `http://www.thedyslexia-spldtrust.org.uk/media/downloads/inline/the-rose-report.1294933674.pdf`. accessed: 2021-11-13.

[6] John Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2):29–40, 2013.

[7] Fred D Davis, RP Bagozzi, and PR Warshaw. Technology acceptance model. *J Manag Sci*, 35(8):982–1003, 1989.

[8] deCode Genetics. About decode genetics. URL `https://www.decode.com/company/`. accessed: 2021-12-4.

[9] Rachel N. Denison, An T. Vu, Essa Yacoub, David A. Feinberg, and Michael A. Silver. Functional mapping of the magnocellular and parvocellular subdivisions of human lgn. *NeuroImage*, 102:358–369, 2014. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2014.07.019. URL `https://www.sciencedirect.com/science/article/pii/S1053811914005916`.

[10] Guinevere F. Eden, John W. VanMeter, Judith M. Rumsey, José Ma. Maisog, Roger P. Woods, and Thomas A. Zeffiro. Abnormal processing of visual motion in dyslexia revealed by functional brain imaging. *Nature*, 382(6586):66–69, Jul 1996. ISSN 1476-4687. doi: 10.1038/382066a0. URL `https://doi.org/10.1038/382066a0`.

[11] Expressjs.com. Express. URL `https://expressjs.com/`. accessed: 2022-06-12.

[12] Feide. Om feide. URL `https://www.feide.no/om-feide`. accessed: 2022-06-12.

[13] Roy Thomas Fielding. Representational state transfer (rest). `https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm`, 2000.

[14] Figma. About figma. URL `https://www.figma.com/about/`. accessed: 2022-06-12.

[15] S Gao. High level modeling and evaluation of multi-channel services. thesis for the degree of philosophies doctor. 2011.

[16] GitHub.com. Github. URL `https://github.com/`. accessed: 2022-06-12.

[17] Google. Google account dashboard. URL `https://myaccount.google.com/`. accessed: 2022-06-14.

[18] Peter C. Hansen, John F. Stein, Sam R. Orde, Jonathan L. Winter, and Joel B. Talcott. Are dyslexics' visual deficits limited to measures of dorsal stream function? *NeuroReport*, 12(7), 2001. ISSN 0959-4965. URL `https://journals.lww.com/neuroreport/Fulltext/2001/05250/Are_dyslexics__visual_deficits_limited_to_measures.45.aspx`.

[19] Heroku. About heroku. URL `https://www.heroku.com/about`. accessed: 2022-06-12.

[20] Neil Humphrey and Patricia M. Mullins. Self-concept and self-esteem in developmental dyslexia. *Journal of Research in Special Educational Needs*, 2(2), 2002. doi: https://doi.org/10.1111/j.1471-3802.2002.00163.x. URL `https://nasenjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1471-3802.2002.00163.x`.

[21] ISO. Systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models, . URL `https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en`. accessed: 2021-11-24.

[22] ISO. Ergonomics of human-system interaction — part 11: Usability: Definitions and concepts, . URL `https://www.iso.org/obp/ui/#iso:std:63500:en`. accessed: 2021-11-24.

[23] Fredrik Jenssen. Magno: An application for detection of dyslexia - from native to web app. 2020.

[24] Thea Johansen and Maja Kirkerød. Magno: An application for detection of dyslexia. 2017.

[25] Pernille Andresen Klevstuen. Magno: Screening young children for dyslexia. 2021.

[26] Teri Lawton. Improving magnocellular function in the dorsal stream remediates reading deficits. *Optometry and Vision Development*, 42(3):142, 2011.

[27] Emily M. Livingston, Linda S. Siegel, and Urs Ribary. Developmental dyslexia: emotional impact and consequences. *Australian Journal of Learning Difficulties*, 23(2):107–135, 2018. doi: 10.1080/19404158.2018.1479975. URL `https://doi.org/10.1080/19404158.2018.1479975`.

[28] M S Livingstone, G D Rosen, F W Drislane, and A M Galaburda. Physiological and anatomical evidence for a magnocellular defect in developmental dyslexia. *Proceedings of the National Academy of Sciences*, 88(18):7943–7947, 1991. ISSN 0027-8424. doi: 10.1073/pnas.88.18.7943. URL `https://www.pnas.org/content/88/18/7943`.

[29] material.io. Material design. URL https://material.io/. accessed: 2022-06-12.

[30] Microsoft. What is azure?, . URL https://azure.microsoft.com/en-us/overview/what-is-azure/. accessed: 2022-06-12.

[31] Microsoft. Microsoft announces immediate availability of windows millennium edition (windows me), . URL https://news.microsoft.com/2000/09/14/microsoft-announces-immediate-availability-of-windows-millennium-edition\-windows-me/. accessed: 2021-12-1.

[32] Microsoft. Visual studio code, . URL https://code.visualstudio.com/. accessed: 2022-06-12.

[33] mui.com. Material ui. URL https://mui.com/material-ui/getting-started/overview/. accessed: 2022-06-12.

[34] NHS. Dyslexia. URL https://www.nhs.uk/conditions/dyslexia/. accessed: 2021-10-21.

[35] Nodejs.org. Nodejs. URL https://nodejs.org/en/. accessed: 2022-06-12.

[36] Dysleksi Norge. Dysleksi, . URL https://dysleksinorge.no/dysleksi/. accessed: 2021-10-11.

[37] Dysleksi Norge. Dysleksi, . URL https://dysleksinorge.no/rettigheter-skole/. accessed: 2021-10-22.

[38] Dysleksi Norge. *Faglige retningslinjer for kartlegging, utredning og oppfølging av elever med dysleksi*. 2017. ISBN 978-82-90503-15-9. URL https://dysleksinorge.no//wp-content/uploads/2017/08/faglige-retningslinjer-versjon-23.pdf.

[39] Benny J. Oates. Researching information systems and computing. 2006.

[40] José Rolando Guay Paz. *Microsoft Azure Cosmos DB Revealed: A Multi-Model Database Designed for the Cloud*. Apress, 2018.

[41] Robin L. Peterson, Bruce F. Pennington, Richard K. Olson, and Sally J. Wadsworth. Longitudinal stability of phonological and surface subtypes of developmental dyslexia. *Scientific Studies of Reading*, 18(5):347–362, 2014. doi: 10.1080/10888438.2014.904870. URL https://doi.org/10.1080/10888438.2014.904870.

[42] Franck Ramus, Stuart Rosen, Steven C. Dakin, Brian L. Day, Juan M. Castellote, Sarah White, and Uta Frith. Theories of developmental dyslexia: insights from a multiple case study of dyslexic adults. *Brain*, 126(4):841–865, 04 2003. ISSN 0006-8950. doi: 10.1093/brain/awg076. URL https://doi.org/10.1093/brain/awg076.

[43] Keith Rayner, Barbara R. Foorman, Charles A. Perfetti, David Pesetsky, and Mark S. Seidenberg. How psychological science informs the teaching of reading. *Psychological Science in the Public Interest*, 2(2):31–74, 2001. doi: 10.1111/1529-1006.00004. URL https://doi.org/10.1111/1529-1006.00004. PMID: 26151366.

[44] Reactjs.org. React, . URL https://reactjs.org/. accessed: 2022-06-12.

[45] Reactjs.org. React, . URL https://github.com/facebook/react/. accessed: 2022-06-12.

[46] Sir Jim Rose. Definition of dyslexia, 2009. URL https://www.bdadyslexia.org.uk/news/definition-of-dyslexia. accessed: 2021-11-13.

[47] Jeff Sauro. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC, 2011.

[48] Johannes Schumacher, Per Hoffmann, Christine Schmäl, Gerd Schulte-Körn, and Markus M. Nöthen. Genetics of dyslexia: the evolving landscape. *Journal of medical genetics*, 44(5):289–297, 2007. doi: 10.1136/jmg.2006.046516. URL https://doi.org/10.1136/jmg.2006.046516.

[49] Scrum. Scrum.org. URL https://www.scrum.org/resources/what-is-scrum/. accessed: 2022-06-12.

[50] Margaret Snowling. Dyslexia as a phonological deficit: Evidence and implications. *Child Psychology and Psychiatry Review*, 3(1):4–11, 1998. doi: https://doi.org/10.1111/1475-3588.00201. URL https://acamh.onlinelibrary.wiley.com/doi/abs/10.1111/1475-3588.00201.

[51] John Stein. The magnocellular theory of developmental dyslexia. *Dyslexia*, 7(1):12–36, 2001. doi: https://doi.org/10.1002/dys.186. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/dys.186.

[52] John Stein. What is developmental dyslexia? *Brain Sciences*, 8(2), 2018. ISSN 2076-3425. doi: 10.3390/brainsci8020026. URL https://www.mdpi.com/2076-3425/8/2/26.

[53] John Stein. The current status of the magnocellular theory of developmental dyslexia. *Neuropsychologia*, 130:66–77, 2019. ISSN 0028-3932. doi: https://doi.org/10.1016/j.neuropsychologia.2018.03.022. URL https://www.sciencedirect.com/science/article/pii/S0028393218301155. Developmental dyslexia: From genes to remediation.

[54] John Stein and Vincent Walsh. To see but not to read; the magnocellular theory of dyslexia. *Trends in Neurosciences*, 20(4):147–152, 1997. ISSN 0166-2236. doi: https://doi.org/10.1016/S0166-2236(96)01005-3. URL https://www.sciencedirect.com/science/article/pii/S0166223696010053.

[55] Typescriptlang.org. Typescript. URL https://www.typescriptlang.org/. accessed: 2022-06-12.

[56] Lionel Sujay Vailshery. Developer framework survey. URL https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/. accessed: 2022-06-12.

[57] Richard K. Wagner, Ashley A. Edwards, Antje Malkowski, Chris Schatschneider, Rachel E. Joyner, Sarah Wood, and Fotena A. Zirps. Combining old and new for better understanding and predicting dyslexia. *New Directions for Child and Adolescent Development*, 2019(165):11–23, 2019. doi: https://doi.org/10.1002/cad.20289. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cad.20289.

[58] Bjørnar Håkenstad Wold. App for early detection of dyslexia. 2016.

Appendix

A Standard SUS Form

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

Figure 39: Standard SUS questionnaire

B Usability Testing Tasks

Magno er et system som består av en liten samling av tester som skal hjelpe til med å kartlegge sannsynligheten for at en elev på barneskolen har dysleksi. Systemet er opprettet for å hjelpe lærere og spesialpedagoger til å enkelt kunne administrere testresultater, og gjennomføre disse testene. Under kommer noen punkter som det er ønsket at du skal prøve. De kan gjøres stegvis og du trenger ikke lese gjennom alle oppgavene før du starter.

1. Du er en lærer som ønsker å opprette en bruker på nettsiden og du jobber på Huseby Barneskole.
2. Du ønsker nå å logge inn på nettsiden for å ta i bruk systemet, det første du vil gjøre etter du har logget inn er å finne mer informasjon om hvordan testene fungerer, og hva resultatene fra disse betyr.
3. Du har en elev som heter Kari Nordmann som går i klasse 4A som du ønsker å legge til i nettsiden for å kunne gjennomføre tester med henne senere.
4. Du ønsker nå å gjennomføre en Motion Test med Kari Nordmann. (For denne oppgaven trenger du ikke en annen person til å gjennomføre testen, den gjennomfører du selv)
5. Nå som testen er ferdig, ønsker du å se resultatene til Kari Nordmann på eleven sin oversikt, og hva resultatet kan indikere.
6. Du er nå ferdig å bruke systemet, og ønsker å logge ut.

C Adapting Magno for children

Klevstuen derived several important requirements to adapt Magno. Those that were deemed to be of high priority by her have been listed in Table 4.

Table 4: High Priority Derived Requirements [25]

Id	**Description**
FR1	The user should be able to choose if they are a supervisor or an individual test taker
FR2	The individual test taker should be able to understand the tests through a tutorial
FR3	The individual test taker should be able to see and interpret own results
FR4	The supervisor should type in the name and age of the child before conducting a test
FR5	The supervisor will be presented with clear and precise instructions for how to use Magno, introduce the tests, and guide the child
FR6	The supervisor will be able to try the tests before supervising a child
FR7	The child will be shown some test examples before taking the test
FR10	The supervisor should be able to see the test result after a child has completed a test
FR12	The application should have a test completed page to display to the child after test completion
FR13	The result page will give a more detailed explanation of the test results and possible scores
FR14	The child can choose between different moving objects in the motion test
FR17	When clicking a frame, feedback will be added to indicating to the child that a choice has been made
FR18	The frames in the motion test and motion test tutorial for children will be enlarged
FR19	The frames in the form fixed test and form fixed test tutorial for children will be enlarged
FR20	The level of coherent motion at the beginning of the motion test and motion test tutorial will be increased
FR21	The child will hear background music during the tests
NFR1	The application should support the Norwegian language
NFR2	The tests intended for children shall be designed in such a way that it relieves the children from stress related to time restrictions

NTNU
Norwegian University of
Science and Technology