CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

From Paper to Tablets: Digitalizing Exams

Designing an Intuitive and Efficient Tablet-based Exam Experience

Bachelor's thesis in Computer Science and Engineering

Mona Chabokdavan
Gabriele Frattini
Junyi Kuang
Oscar Lindfors
Viktor Lübeck
Natalie Stein

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Bachelor's thesis 2025

From Paper to Tablets: Digitalizing Exams

Designing an Intuitive and Efficient Tablet-based Exam Experience

Mona Chabokdavan
Gabriele Frattini
Junyi Kuang
Oscar Lindfors
Viktor Lübeck
Natalie Stein

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Bachelor's thesis template for LaTeX for CSE

Paper to Tablets: Digitalizing Exams
Designing an Intuitive and Efficient Tablet-based Exam Experience

Mona Chabokdavan Gabriele Frattini Junyi Kuang Oscar Lindfors
Viktor Lübeck Natalie Stein

Supervisor (Handledare): Goran Trninic, Department of Computer Science and Engineering
Examiners: Patrik Jansson and Arne Linde, Department of Computer Science and Engineering
Graded by teacher (Rättande lärare): Thommy Eriksson, Department of Computer Science and Engineering

Bachelor's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Bachelor's thesis template for LaTeX for CSE
From Paper to Tablets: Digitalizing Exams
Designing an Intuitive and Efficient Tablet-based Exam Experience

Mona Chabokdavan, Gabriele Frattini, Junyi Kuang, Oscar Lindfors,
Viktor Lübeck, Natalie Stein

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The increasing digitalization of education has highlighted the limitations of traditional paper-based exams, particularly regarding efficiency, security, and sustainability. Existing digital exam platforms, such as Inspera, often fall short in subjects requiring sketches, formulas, or structured formatting. This thesis addresses the need for a more flexible solution with improved usability that supports both students and teachers.

The aim of the project is to further develop a digital exam application for Android tablets that enables both freehand drawing and structured text input. Building on a previous project, this work focuses on improving usability and strengthening security, with the overall goal of reducing administrative workload and enhancing the exam experience.

Through usability testing, key usability challenges and security concerns were identified and addressed. New features such as a lasso tool, recovery codes, automatic submission, and time-restricted access were implemented to meet these needs. While participants generally found the interface intuitive, some aspects such as switching between drawing and scrolling require further refinement.

The final prototype combines core exam functionality with a clean and efficient interface. Results indicate that tablet-based exam systems can improve flexibility, accessibility, and user experience. Future development should explore accessibility adaptations and integration with external administrative systems.

Keywords: Digital exam platforms, Usability, Security, Android tablets, Structured text input, Freehand drawing

Sammandrag

Digitaliseringen inom utbildningssektorn har tydliggjort begränsningarna med traditionella pappersbaserade tentor, särskilt när det gäller effektivitet, säkerhet och hållbarhet. Dagens digitala tentaplattformar, såsom Inspera, är ofta otillräckliga för ämnen som kräver ritningar, formler eller avancerad formatering. Det finns därför ett tydligt behov av en mer flexibel, användarvänlig och mångsidig lösning som kan möta studenternas och lärarnas behov.

Syftet med detta examensarbete är att vidareutveckla en digital tentamensapplikation för Androidsurfplattor som erbjuder både ritfunktioner och strukturerad texthantering. Arbetet bygger vidare på ett tidigare projekt och fokuserar på att förbättra användbarheten och stärka säkerheten. Målet är att minska den administrativa arbetsbördan och förbättra tentamensupplevelsen.

De huvudsakliga problemen som identifierats rörde bristande användarvänlighet och säkerhetsbrister i befintliga lösningar. Dessa adresserades genom användbarhetstester och implementering av nya funktioner såsom lasso-verktyg, återställningskoder, automatisk inlämning och tidsstyrd åtkomst. Testerna visade att applikationen generellt var intuitiv, men att vissa funktioner, exempelvis textinmatning och växling mellan rit- och rullningsläge kunde förbättras ytterligare.

Den färdiga applikationen kombinerar central funktionalitet för tentor med en tydlig och effektiv gränssnittsdesign. Resultaten visar att surfplattebaserade tentamenssystem kan ge ökad flexibilitet, tillgänglighet och användarupplevelse. Framtida utveckling bör inkludera ytterligare tillgänglighetsanpassningar och integration med externa system för studieadministration.

Acknowledgements

We would like to begin by expressing our sincere gratitude to our supervisor Goran Trninic, at the Department of Computer Science, Chalmers University of Technology, for the valuable guidance and continuous support throughout this project.

Our heartfelt thanks also go to the previous group for generously sharing their work with us and helping us get started.

We are equally grateful to everyone who participated in user testing and evaluating the product. A special thanks goes to UX Designer Frida Ylitalo, whose valuable input during the development process was greatly appreciated.

Finally, we sincerely appreciate Maria Sunnerstam, at the PIL unit, University of Gothenburg, for her time during the interview and for providing valuable insights and information relevant to our work.

We acknowledge the utilization of AI-powered tools during the development of this thesis, including OpenAI's ChatGPT, Microsoft Copilot for text editing, and QuillBot for grammar and style recommendations. All AI-generated content was reviewed, rewritten, and edited by the authors. The responsibility for the final content and conclusions of this thesis lies entirely with the authors.

Mona Chabokdavan, Gabriele Frattini, Juny Kuang,
Oscar Lindfors, Viktor Lübeck, Natalie Stein
Gothenburg, May 2025

Contents

List of Figures

List of Tables

1
Introduction

This chapter introduces the background and motivation for the project, outlines the project's purpose and scope, and presents the key research questions. Furthermore, it reviews related work and existing solutions within digital examination systems. Lastly, the chapter explains the starting point of the project, its connection to an earlier prototype, and the specific conditions and limitations that shaped the direction of the work.

1.1 Background

As the world becomes more digital, education systems must be integrated. Therefore, an efficient system is crucial to facilitate the examination writing process and optimize the grading process for students and teachers. Although the classic pen-and-paper format is still the most common way to write exams, more technical solutions have been introduced to conduct online exams in recent years.

Educational institutes deliver digital exams with the help of a third-party platform [1]. For example, Inspera is a program that allows students to take exams on a computer, and it has been adopted by many institutions. However, it currently does not fully meet the required needs. Inspera, like many other examination platforms, is designed for computer-based use and primarily supports text input. This format lacks the flexibility required to serve as a standardized method for all types of exams. Inspera has limitations when it comes to drawing figures or writing formulas, which are essential in exams for subjects such as mathematics or chemistry [2].

Because traditional exam platforms struggle with flexibility, especially for subjects that need more than simple text or multiple-choice answers, tablets would be a better option to consider that could provide features to write, draw, and interact with, as well as offering stylus and touch capabilities.

This project builds on previous work completed by a group of computer science students who initiated this product as part of a course project. The product is called "Tentamina Applikation" and is a prototype that handles digital exams on tablets. The application offers features for writing and drawing figures and a "Teacher Portal" where teachers can create the exams [3]. By extending this foundation, the current project aims to address key challenges and improving the system for effective use during exams.

Two areas noted for improvement were usability and security. In the context of this project, usability refers to creating an intuitive exam experience for students and ensuring easy access to editing tools during the test. This is important since it ensures that students can express themselves accurately without interference. Furthermore, proctors and teachers are not considered our primary target group thus, improving the usability for them is not the primary concern.

At the same time, the sensitive nature of exam data introduces strict requirements for security. In the context of this project, security refers to ensuring that students' work is saved during the exam, that access to the exam is restricted to the correct time and authorised users, and that interruptions can be recovered without data loss. This is important for building a reliable platform where students can feel confident that their work will not be lost, while also preventing cheating. These combined challenges set the direction for the work carried out in this project.

1.2 Purpose

The aim is to further develop the exam application by providing an intuitive and secure platform for digital exams on tablets. Features such as editing tools for drawing and writing are offered to simplify the examination process for students. The idea is to provide the student with enough tools to be able to complete exams in different subjects where diagrams, formulas, or sketches can be a part of the answer.

To achieve the purpose of this project, the following research questions will be addressed:

- How can a tablet based digital examination platform be designed to serve as an effective substitute for pen and paper exams without limiting the students' work?
- What security measures should be implemented to ensure that a tablet based examination platform is as secure and reliable as pen and paper exams?
- What can user testing of the interface reveal about usability challenges, and how can this feedback be used to improve the exam experience?

1.3 Related Work

In recent years, examinations have experienced major transformations with the use of digital assessment platforms such as Inspera. However, these platforms often lack the necessary functionality and usability for certain subject areas [2]. Well-designed tools are essential for digital assessments to accurately reflect the syllabus.

1.3.1 Digital Examination Platforms and the DISA Project

The University of Gothenburg started a project to modernize the education system in Sweden by moving from paper exams to digital ones in 2014. The goal was to have 65% of the exams digital by 2019 and to make the exams more efficient, secure, and accessible [4]. Several educational institutions across Sweden currently utilize the Inspera digital platform for assessments.

Inspera is a computer-based platform designed specifically for text-based examinations, providing a secure and reliable exam environment. This system uses the Safe Exam Browser (SEB) to maintain exam security and offers functionalities such as anonymous grading, exam scheduling, question creation, and integration with external systems such as Ladok [5]. While widely used, it is limited to desktop usage and lacks drawing capabilities, which can make it less suitable for subjects requiring visual representations [2].

To obtain additional insights into digital assessments, a personal interview was conducted with Maria Sunnerstam, a former project leader for the DISA at the University of Gothenburg, on March 27, 2025. During the interview, Sunnerstam shared that the School of Business, Economics, and Law initiated the digital exam project in 2013 and expanded it across the entire institute by 2016.

Early pilot projects, supported by the Pedagogical Development and Interactive Learning (PIL) unit, tested available systems after establishing connections with the Danish company Wiseflow and the Swedish company DigiExam. Ultimately, the decision was made to choose Inspera, which was a Norwegian company. Sunnerstam noted that some important questions were addressed during the project, such as the concerns about ensuring security, students' experience with the platform, and teachers' views on the platform's functionality. These factors were crucial for a successful implementation of a system that was both efficient and reliable.

1.3.2 Implementation and Security

The small-scale tests demonstrated the project's operational and technical issues, while it was helpful for the institution to prepare the ground for the adoption of DISA and shift to digital exam tools. Sunnerstam explained that the university decided not to have fixed computers in exam halls due to the complexity and regular service costs. For security reasons, it was important that digital exams could not simply be set up anywhere. Institutions should not be allowed to organize a digital exam remotely since security cannot be ensured. Instead, DISA was implemented within controlled and secure exam rooms equipped with private networks. This configuration helped to find out about any attempts to connect external devices. There were particular concerns that students may find some ways to allow external parties to take remote control of devices during exams. One important question that came up was whether students should bring personal laptops or if the university should supply computers on-site.

After carefully considering the financial implications, the University of Gothenburg chose to provide a certain number of loaner laptops for students who lack access to a suitable device or face technical problems during the exam. The first approach followed a Bring Your Own Device (BYOD) model. The DEX project mainly focused on academic integration and secure delivery through the Safe Exam Browser (SEB). DISA uses a BYOD model, which allows students to use personal laptops or borrow one if needed, making the process more flexible. However, it is mainly built for use on desktops and laptops. Despite some efforts to add drawing features, these remained basic and were not fully integrated into the system [6].

As highlighted by Sunnerstam, during the DEX project, an external company performed security auditing, and the results were kept private to protect the system. Additionally, the project required training in managing digital exams and identifying academic dishonesty. The grading interface proved beneficial for teachers, particularly for large courses with numerous assessments. Students appreciated the flexibility of digital platforms in accommodating individual needs, such as not having to worry about handwriting difficulties, having extended exam time, special seating, and support for tools like screen readers.

1.3.3 Challenges and Limitations

The interview with Sunnerstam provided valuable insights about using these systems on a large scale. Digital exam platforms have several common limitations. These include a high level of technical setup for installation, lockdown, and network configurations. Limited support for tablet-based processes can also reduce adaptability in certain situations.

Sunnerstam points out that platforms like Inspera allow specific tools during exams, like SPSS for statistical analyses or a pharmaceutical system that Sahlgrenska was permitted to use for accessing patient data. However, she also notes unresolved challenges, such as keyboard layout switching during language exams and other environmental factors like inadequate lighting or poor screen contrast in examination halls.

The experience with digital examination platforms indicated the potential and limitations currently existing and highlighted the importance of further development in digital assessment solutions.

1.4 Starting Point: Inherited Student Project

The starting point of the project was a partially developed digital examination system, implemented by another student group at Chalmers University of Technology [3]. The system consisted of two main components, an examination application and a web-based teacher portal. This previous work served as a foundation for the current project. Although it provided a valuable starting point, the system also had some limitations. The primary problems were related to usability issues in the student exam interface, which lacked intuitive navigation and essential functionalities. Additionally, there were security vulnerabilities that allowed students to access exams without proctor supervision, start exams before the allowed time, and retake exams multiple times. These issues formed important areas for improvement in this project.

The following sections describe the technical conditions at the time of handover, as well as the status of the system, including known issues and areas identified for improvement by the previous group.

1.4.1 Inherited Codebase and Architecture

At the time of the project handover, the previous group provided a codebase structured into two main parts: the teacher portal and the exam application. It consisted of a single repository where both the teacher portal and the exam application were located. The exam application was primarily developed in Android Studio using Kotlin as the programming language and Jetpack Compose as the framework. On the other hand, the teacher portal was structured as a web application with Node.js and Express to enable REST API communication with the exam application and the database. The previous group also used JSON as the data format for interactions between the client and server. Contents of the exam were stored in the database, while submitted exams were stored in a folder in the Exam API. The exam application also used a backup system where progress was saved to an SD card on the tablet [3]. A more detailed explanation of the system's architecture, including the roles of Jetpack Compose, Node.js, and Firebase, is provided in section 2.1 System Design.

1.4.2 Known Issues and Limitations at the Handover

At the project handover, the previous group had identified several issues and areas for potential improvement. This section will present the status of the system based on the observations by the previous group, focusing on known problems and incomplete implementations.

Regarding the exam application, one issue was that the lock mechanism, which prevents the user from exiting the application, could be bypassed by the user. Another issue was related to PDF export, where manually inserted line breaks in text input fields were deleted, resulting in PDF formatting that differed from the intended layout.

The history control also showed two primary issues. One was that the undo function could remove multiple drawings at once, instead of removing them step-by-step, leading to potential exam data loss. The other issue was that the application did not keep a separate undo history for each question, which meant it could undo changes in the wrong place and cause unintended progress to be lost. It was also observed that saving progress via the backup system was a resource-demanding operation. Additionally, the application did not support error notifications in failure scenarios, for example, if an SD card was not detected or a submission failed [3].

Furthermore, one limitation of the teacher portal was the absence of integration with external platforms, such as Ladok, for handling student data. The interface of the Teacher Portal at the time of the handover can be seen in figures A.10, A.11, A.12, A.13 and A.14 in Appendix A. The portal also missed editing capabilities and supported only text-based questions. The Exam API was not hosted on a remote server so it ran locally on our computers which worked during development but not in a real production setting [3]. Figure 1.1 and 1.2 below showcase how the examination application interface looked like at the time of the handover. The design was minimalist, including a white canvas with drawing functionality as the primary view and a side menu for navigation between the exam questions. The color scheme, particularly in the login view, had very low contrast, which made it hard to see the recover button.

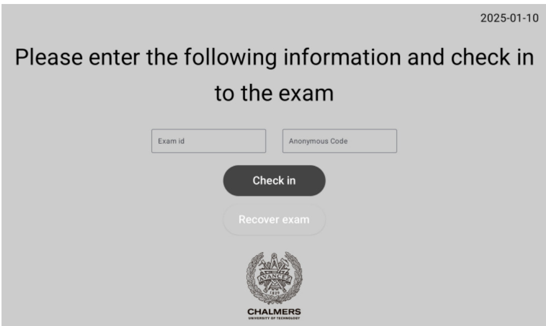

Figure 1.1: Tentamina Applikation: Login page.

Figure 1.2: Tentamina Applikation: Draw view and sidebar.

1.5 Scope

The application was considered to be developed exclusively for Android tablets. This decision was made in consultation with the stakeholders at Chalmers University of Technology, who have concluded that alternative tablets are not within the budget for the intended implementation. To stay consistent with the previous project group's approach, no development for other operating systems will be included at this stage.

Furthermore, integration with Ladok to identify which students will take the examination will not be implemented. This functionality is reserved for a future stage. For the present project, the primary focus will remain on the application's functionality.

Similarly, it has been determined that the application will not be hosted on Chalmers' servers. This decision aligns with the previous group's work, as the primary focus remains on the internal functionality of the application. The specifics of how the application communicates with servers and other devices will be addressed in future development phases.

The project will primarily focus on the design of the user interface and user experience, with security as a secondary priority. Some security features and considerations will naturally be integrated during the development process, in alignment with the functionality being implemented.

Another design choice is that the application will not display digital questions. Like most of the exams that require handwritten answers, the questions will be provided on paper during examinations. This decision was made because of the tablet's small screen size, which could interfere with the student's workflow. Instead, the tablet is intended to act as a direct substitute for pen and paper with the intention to optimize the writing experience.

2

Theory

This section presents the architecture of the exam application, which served as the foundation for the work. It also presents the theories behind the implementation of the examination interface and the theoretical background of the usability testing methods employed.

2.1 System Design

The Exam Application consists of three components:

- **Exam Interface**: The interface used by students to complete and submit exams.
- **Teacher portal**: A website where teachers create exams and review submissions.
- **API**: Backend server with all endpoints used by the exam interface and the Teacher portal.

Below, figure 2.1 illustrates the high-level architecture of the Exam Application and how the API, Exam Interface and Teacher Portal interact with each other.

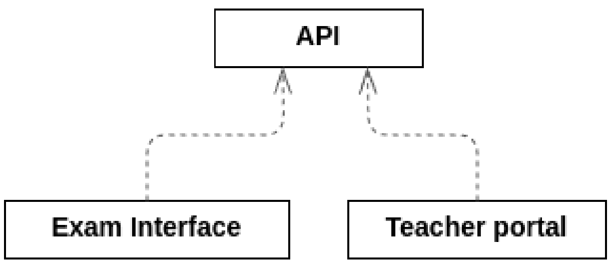

Figure 2.1: High-level architecture.

2.1.1 API

The Application Programming Interface (API) is built with Node.js and Express. Node.js is a runtime that enables the developer to write server-side applications in Javascript. Node's main advantage is its non-blocking, event-driven architecture, which lets it handle tens of thousands of concurrent connections on a single thread without needing to switch between threads [7].

Express is a minimalist web framework for Node.js used to build web applications and APIs quickly and with minimal overhead [8]. The API's purposes are to expose REST endpoints used by the Exam interface and the Teacher portal. The server has a connection to a Firebase real-time database for data storage and validation [9]. Student submissions are stored as PDF files on the server's disk.

2.1.2 Teacher Portal

Web interface that uses the API to render HTML pages server-side with Express. The Web interface allows teachers to create and edit exams. In addition, it is possible to view and download student submissions.

2.1.3 Exam Interface

The Exam interface is built with Jetpack Compose, Android's modern toolkit for building native user interfaces in a declarative and reactive style. Jetpack Compose simplifies User Inteface (UI) development by using composable functions instead of XML layouts. It allows developers to build UIs that automatically update when application state changes [10]. The application follows the Model-View-ViewModel (MVVM) architecture to ensure clean separation of concerns. In this architecture, the Model handles the business logic and data, the View is responsible for rendering the UI, and the ViewModel is a mediator that exposes observable state to the UI and decouples it from the Model [11].

2.1.4 High-Level Component Overview

- Presentation Layer: Responsible for the user interface and user interactions.
  - MainActivity is the entry point of the application that sets up the UI and manages navigation between different screens.
  - LoadingScreen, WaitingScreen, RichEditorScreen, and DrawingScreen are composable functions that represent various screens for different application states.
- View Model Layer: Intermediaries between the UI and the data models, ensuring that the UI remains responsive and can react to data.
  - TentaViewModel manages the state of the drawing canvas, user interactions, and the list of canvas objects.
  - ExamInfo handles exam-related data, including fetching exam details from the server and managing recovery modes.
- Model Layer: Encapsulate the data structures used throughout the application.
  - CanvasObject, Line, TextBox represent the drawable objects on the canvas.
  - SerializableCanvasObject, SerializableLine, SerializableTextbox are used for serialization and deserialization of exam data for JSON and PDF conversion.

In figure 2.2 below, the picture illustrates the high-level component architecture of the system.

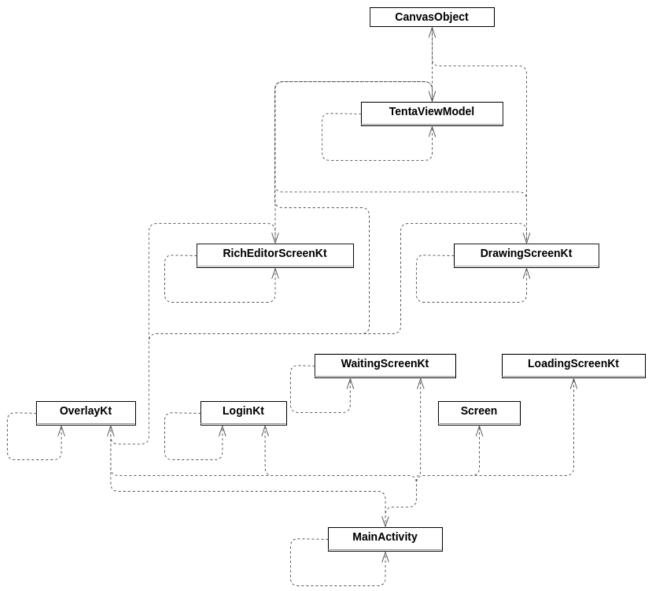

Figure 2.2: High-level component architecture.

2.1.5 Data Flow

When a student starts an exam, the following steps take place:

1. The student checks into the exam using an anonymous code and a course code.
2. API verifies credentials and the exam data is queried from the database.
3. As the student writes or draws answers, the work is saved in-memory and synced to a SD card every 5 seconds so that exam progress can be restored if the tablet accidentally turns off.
4. Once the exam is submitted, the answers are converted to a PDF file and sent to the API where its stored on disk with other student submissions for that exam.

Below, figure 2.3 illustrates how data flows within the system.

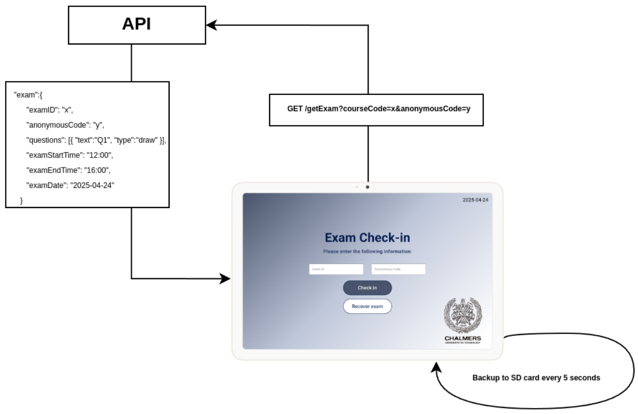

Figure 2.3: Data flow.

2.2 Design Patterns and Principles for Exam Interface

To create an intuitive and effective user interface that the user easily can navigate through and understand well, it is important to use well-known and well-proven design patterns. These patterns help solve common design problems and support better user interaction.

The canvas plus palette is a well-known design principle described in the book "Designing Interfaces", which involves a clear separation between the workspace (canvas) and the toolset (palette) [12]. This design principle is common in creative applications, where the primary goal is to produce or create something. The canvas is normally placed in the middle of the screen, since it is the most central part of the application, while the editing tools are positioned nearby to ensure easy and efficient access.

Progress indicator is another common design principle. The purpose of progress indicators is to show the user how much progress has been made. This often occurs in situations where the user is in a time-consuming operation [12]. This kind of indicator is normally done as filled lines or step-by-step progress bars, but it can also be done with check-marks or other simple symbols and icons. The goal is to give users a clear sense of the progress that has been achieved [12].

2.3 Usability Testing

Usability is a measure of how well a specific user in a specific context can use a product to achieve a defined goal effectively and satisfactorily [13]. In the context of this project, usability refers to creating an intuitive exam experience for students and ensuring easy access to editing tools during the test. Usability testing is a crucial part of the development process. It is the main way to verify that the application is functional and works properly in real-life situations. The goal of the testing is to identify problems, improve the design, and ensure the application meets the users' needs. Studies have shown that testing with 5 to 8 users is often enough to uncover the majority of usability issues, as additional users tend to reveal similar findings [14]. There are several different methods for conducting user tests, for example, the observation- and the think-aloud- method. These methods can be combined to achieve better results.

2.3.1 Heuristic Evaluation

Heuristic evaluation is a usability inspection method in which a small group of evaluators assess an interface based on established usability principles, commonly known as "heuristics." These principles, such as visibility of system status, user control and freedom, and consistency, serve as guidelines to identify common usability issues. This method is particularly relevant for similar design projects, as it allows early identification of problems without requiring large-scale user testing. It is especially useful when access to real users is limited or when time and resources are constrained during the initial design phase [15].

2.3.2 Observation

Observation is a usability testing method where the test leader observes how participants interact with the system without interfering. The goal is to get an objective view of the experience, without influencing the natural behavior of the participants. By doing this, it is possible to capture reactions and see how users actually behave and think while using the product [16].

A basic distinction is made between direct and indirect observation. In direct observation, the researcher is present and observes users in real-time, while in indirect observation, data is analyzed retrospectively through recordings or documents. Both types of observation can be conducted either in the field, the user's natural environment, or in a controlled environment such as a laboratory [16].

Observation is often combined with other methods such as interviews, questionnaires, or the think-aloud technique, where participants express thoughts verbally while performing tasks.

2.3.3 Think-Aloud Method

The think-aloud method involves asking participants to verbally express their thoughts while completing tasks during a test [17]. Users describe the experience interacting with the application, as well as explaining their thought process while solving the tasks. The users are encouraged to describe what is confusing and what works well. This insight helps uncover design issues that might be hard to detect through observation alone.

3
Method and Material

This chapter describes the methods and materials used in the development and evaluation of the exam application. It begins with a user-centered problem analysis using user stories, which guided the creation of the first version of the application. This is followed by an explanation of the usability testing conducted in the second iteration, including the setup, participants, and data collection methods used to evaluate the application's core functionality.

3.1 Heuristic Framing of Design Challenges

In the absence of certified usability experts, the project group adapted the heuristic evaluation method by conducting an internal review based on established usability principles. Acting as informal evaluators, the group assessed the earlier version of the application against these heuristics to identify initial areas for improvement. This analysis served to frame the early understanding of usability needs and guided the formulation of the design challenges. It also provided the foundation for the use of user stories in the next phase, enabling a translation of abstract usability principles into concrete, user-centered development goals.

3.2 User-Centered Problem Analysis

To better frame the problems that need to be solved for students, the user story framework was used to define expectations and challenges as detailed in table B.1 of the Appendix B. User stories are short statements about a feature written from a user's perspective. The statements are not primarily about the feature itself, but rather what the user needs [18]. This method was used during the first iteration of the project and served as the foundation for creating the Minimum

Viable Product (MVP). This approach made it possible to formulate relevant questions and identify areas that required improvements while still leaving room for how the implementation will be developed.

3.3 Usability Testing

When the first iteration of the project was done, it was time for usability testing to actually evaluate the product on real users. The purpose of the usability testing is to evaluate the usability and core functionality of the application. During the test, the focus was on the application's ability to handle freehand drawing, text input, navigation, toolbar functionality, and overall system responsiveness.

The test questions were constructed to resemble real-world exam situations, with each task targeting one or more specific features of the application. This included tasks such as drawing shapes, writing and formatting text, switching between questions, and submitting the exam. After completing the tasks, participants filled out a structured questionnaire evaluating individual experiences.

3.3.1 Material

The usability tests were conducted using a Samsung Galaxy Tab S6 Lite, a small keyboard of the model R-Go Compact Break Keyboard and a digital pen called S Stylus Pen.

The version of the application that the participants were tested on was the MVP version. This iteration of the application is a version that contains the basic functionality for a functional examination platform, that the group members of this project has defined. The MVP version of the application enabled the participants to interact with the application by drawing and writing answers on digital paper, incorporating key functionalities such as erasing, undo, redo, adjusting pen and eraser size, inserting small text boxes for additional comments, and responding to fully text-based questions. Below, figure 3.1, 3.2, 3.3, 3.4, and 3.5 show the interface of the MVP version of the application that was used during the usability testing.

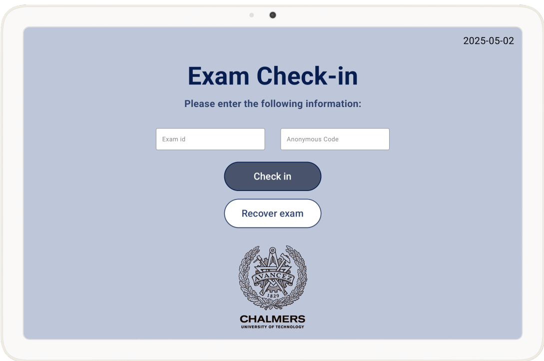

Figure 3.1: MVP: Login page.

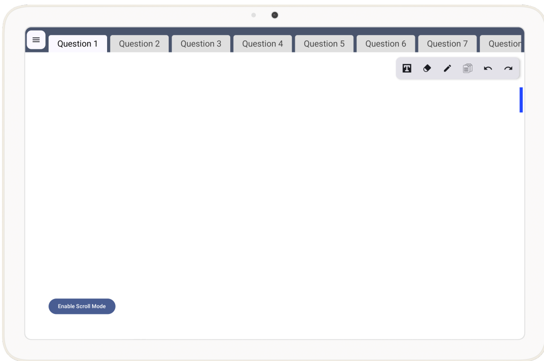

Figure 3.2: MVP: Draw view.

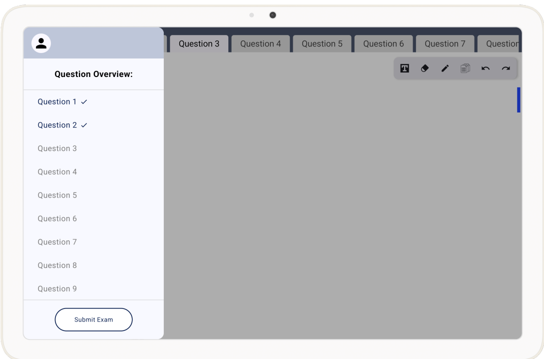

Figure 3.3: MVP: Sidebar.

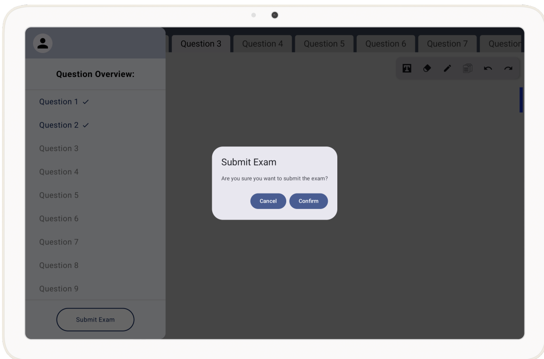

Figure 3.4: MVP: Submit exam.

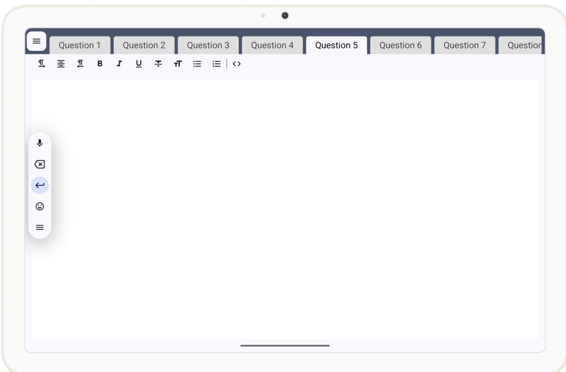

Figure 3.5: MVP: Text view.

During the test, participants were given a paper exam with nine questions that required them to use the tablet exam application. Each question focused on a specific feature of the application.

3.3.2 Test Procedure

Each testing-session took place in a quiet room, this could for example both be on campus or within the home of the test leader. Only the test participant and the test leader were present during the session, which ensured a focused and controlled environment for the testing.

This setting allowed for a clear interaction between the user and the application, minimizing distractions and external influences that possibly could impact the test results.

The test procedure had the following structure, each participant:

1. Received verbal instructions and an introduction to the setup.
2. Logged into the application and completed a digital exam consisting of 9 tasks.
3. Submitted the exam digitally within the app.
4. Filled out a feedback form using Google Forms.

The direct observation methodology was used for the testing, within a controlled environment. The test leaders refrained from assisting or guiding participants through the tasks, while the participants were solving them. This approach was chosen to allow participants to experience the application as users naturally would, without external influence.

The think-aloud method was applied in addition to the observation method. This technique allows users to express ideas while interacting with the system. This method was valuable in providing information about user expectations and usability challenges to the test leader.

The test leader monitored and documented the performance of each participant on each task. Every task was assigned a score from 0 to 3, as follows below in table 3.1.

This rating serves as a complement to the participants' own evaluations in the form, providing a clearer picture of which parts of the

Table 3.1: Observation-based task performance scoring.

Score	Description
3	Task completed smoothly without any problems
2	Task completed after some hesitation or retries
1	Task completed with help from the test leader
0	Task could not be completed within the system

application worked well and which parts presented issues. The specific questions that were used in the usability testing are presented in figure A.1, A.2, A.3 and A.4 in Appendix A. A summarized version of the covered functionalities and features of the questions can be seen in table 3.2 below.

Table 3.2: Test scenarios.

Test Scenarios Description
Log in with the course code and anonymous code.
Navigate through questions and make sure that no answers are lost.
Connect a keyboard and answer a text question and verify that line breaks and formatting works as expected.
Draw freely by using a digital pen in the drawing mode.
Verify that "undo" and "redo" work and only affect the current question.
Verify that the eraser works.
Place a small text-box and verify that it works as expected.
Submit the exam.
Choose a different background style such as grid or lined, then verify that it works properly in drawing mode.
Verify that the top menu clearly shows which question you're on and that scrolling through questions works as intended.

3.3.3 Participants

15 individuals participated in the usability testing, recruited through the project group's personal and professional networks. The goal was to capture a broad range of user types to get a diverse dataset. The participants varied in:

- Age (from 22 to 53).
- Previous experience with digital exams (first-time and experienced users).
- Technical proficiency (self-assessed on a scale from 1 to 5).
- Functional variations (e.g. dyslexia, visual impairments).
- Male and female.

- Participants who are both currently studying, and participants who are working.

Research shows that between 5 and 8 participants are usually sufficient to identify most usability issues [14], it was chosen to test with 15 participants. The reason for this was to ensure that no important insights were missed, especially since several of the participants were recruited through the personal network of the team, which could potentially influence the results. This is discussed further in section 5.2 Reflections on the Usability Testing.

3.3.4 Data Collection

From the usability testing, there were two main types of data collected:

- **Quantitative data:** Structured form responses (e.g., Likert scales, yes/no questions) on usability aspects such as navigation, drawing, toolbar functionality, and system feedback.
- **Qualitative data:** Open-ended feedback questions from the form, verbal think aloud- comments, and test facilitator notes.

The quantitative responses were compiled and analyzed using Excel to identify averages and response trends. Qualitative data was thematically analyzed to identify common patterns and insights. This analysis formed the basis for the results presented in chapter 4.

3.4 Design and Prototyping

The initial interface design of the application was created in the design interface tool Figma. These prototypes served as a foundation for the application's user interface and were iteratively refined during the development of the examination application.

To design the interface of the examination application, known design principles and design choices were applied. The primary design approach was based on the canvas plus palette design principle, as the application is a creative application where the users draw on a canvas. However, because the examination application requires more navigation than typically needed associated with the canvas plus palette pattern, some modifications were made to this pattern. Scrollable tabs were added and used for navigation between questions, as well as a side menu to store additional buttons and information that are not essential on the main canvas view.

4
Results

After conducting usability tests on the interface of the examination application prototype, meaningful feedback was obtained to guide further development. The main concerns included the clarity of the interface, text and drawing modes, and the control features for submission and navigation. This feedback helped evaluate the application's usability, identify problem areas, and validate the strengths of the core features. The findings are summarized below.

4.1 Usability Test Results

The usability tests of the interface of the examination application provided both quantitative and qualitative insights into the strengths and weaknesses of the platform. Participants generally found the application intuitive, although several areas for improvement were identified.

4.1.1 Quantitative Results

Overall, the application received favorable ratings across key usability areas, indicating that its core design effectively supports user interaction. A summary of the results from both evaluation methods is presented in table 4.1 and 4.2 below.

Table 4.1: Average ratings grouped by feature category based on participant responses.

Feature Category	Average Rating (1–5)
General usability	4.51
Drawing with stylus	4.35
Text input	2.85
Toolbar functionality	4.46
UI clarity	4.85
Hardware experience	4.12
Overall impression	4.31

Table 4.2: Test leader scores from observation.

Feature Category	Average Rating (0–3)
General usability	2.87
Drawing with stylus	2.54
Text input	2.38
Toolbar functionality	2.45
UI clarity	2.69
Hardware experience	2.42
Overall impression	2.62

The usability evaluation combined both participant self ratings and observation based scores, offering a thorough overview of the application's performance. For both evaluation methods, individual test questions were grouped into broader feature categories based on the intended purpose and function within the application to give more meaningful comparisons between related functionalities.

The participant self ratings were generally positive, with most feature categories receiving averages above 4.0. In particular, UI clarity received the highest average rating of 4.85, followed by general usability (4.51) and toolbar functionality (4.46). Drawing with the stylus and hardware experience were also rated highly, at 4.35 and 4.12 respectively. However, text input received a noticeably lower average of 2.85, indicating room for improvement in handling structured text and formatting.

These findings were supported by the test leader's observation based scoring. The average overall score across all tasks was 2.62 out of 3, showing that most participants were able to complete the tasks with minimal difficulty. General usability tasks scored particularly well at 2.87, while drawing with stylus (2.54), toolbar functionality (2.45), and hardware experience (2.42) received similar ratings, all slightly below the overall impression average of 2.62. Text input again emerged as the weakest area with an average score of 2.38, confirming that it was a less intuitive part during interaction. These results suggest that, while the core functionalities of the application performed well, improvements in text handling would further enhance the overall user experience.

4.1.2 Qualitative Results

The qualitative feedback collected through the think aloud method and open-ended questionnaire responses provided deeper insights into usability. Participants found the application intuitive for basic tasks such as navigating between questions and drawing freehand. However, users also highlighted areas where the functionality could be expanded, such as the need for better text formatting tools, the ability to zoom while drawing, and more advanced selection tools like a lasso function.

Several participants mentioned uncertainty around the saving process when moving between questions, which created unnecessary stress during the test. Additionally, users wanted clearer feedback before finalizing the exam during the submission process. While technical performance was generally stable, some users reported minor input lag when performing extended drawing tasks. A summary of the most top three most frequently mentioned responses to selected open-ended questions from Google Forms is presented below in table 4.3.

Table 4.3: Summary of qualitative results from open-ended questions.

Question	Top Responses
Did you miss any functionality?	• Clear "submit" confirmation. • More advanced drawing tools (for example lasso, copy, shapes and colors). • Ability to zoom in/out.
Did you experience any screen response issues?	• Unable to scroll. • Problems with the text box in draw view (drag and drop does not work). • Unfamiliarity with Android devices.
Best part of taking the exam digitally?	• Easy to correct mistakes. • Faster and more efficient workflow. • Less paper clutter.
Negative aspects of the digital exam experience?	• Concerns about saving and reliability. • Difficulties with scrolling. • Technical issues.
Other comments or suggestions?	• Unclear scroll button. • Better submission overview of unanswered questions. • Issues related to typing in text view.

Overall, participants described the application as accessible and easy to use once familiarized, but identified specific usability issues that, if addressed, could improve the overall user experience.

4.1.3 Identified Usability Issues

When analyzing the user tests, both quantitative results and qualitative, some usability issues could be detected. These issues can be seen below in table 4.4.

Table 4.4: Identified usability issues.

Usability Issue	**Description**
Unclear "Submit" Confirmation	Participants expressed uncertainty about whether the exam had been successfully submitted.
Problematic Enable Scroll-button	Several participants found it challenging to switch between scroll mode and draw mode on the exam. This lead to confusion on how to scroll. Left handed people also unintentionally pressed the button while using the pen.
Lack of Advanced Drawing Tools	Participants wanted more advanced features for the pen based questions, such as lasso-feature or the ability to draw shapes like circles or squares.
Issues in text view	Participants experienced minor bugs and limitations in the text view. Bullet points reset after pressing enter, formatting icons were unclear, and a deleted character sometimes reappeared when switching questions. Selecting specific text for formatting was also restricted.

These issues were prioritized for improvement in the next sprint of the project's development.

4.1.4 Identified Strengths and Well-Functioning Features

When receiving the feedback, there were some aspects that worked as it was intended without any problems. In this following table, the features that worked out as intended and met the needs of the users is displayed below in table 4.5.

Table 4.5: Well-functioning features identified during testing.

Well Functioning Feature	Description
Information Button	Navigating to the information button that is located in the menu-bar were no issue for the participants.
Navigating of drawing-tools	Participants navigated the different drawing tool options without difficulty. The ability to easily select the preferred paper type, as well as adjust the pen and eraser sizes according to the needs.
Submit Exam	Participants had no problem navigating and finding the "Submit" button. The submission process was done without difficulties.

These well-functioning features demonstrate that several core functionalities of the examination application already met the expectations of the participants. These features required little to no revision and were kept largely in the same state.

4.2 Final Examination Application

This section presents the final version of the exam application developed during this project. Following the usability testing phase, a range of changes were implemented in the examination application to enhance the application based on user feedback, both regarding the visual interface and new features.

4.2.1 Implemented Feautures and Final Examination Application

The final version of the exam interface uses the Canvas Plus Palette design principle as primary design pattern, with some additions such as navigation tabs and a side menu.

The interface features clickable and scrollable tabs for navigating between questions directly within the main canvas area. To complement this, a side menu has been included, providing an overview of all questions and the current status (answered or unanswered). The Progress Indicator pattern was applied within the side menu for this purpose.

Following the usability testing, several new features were implemented in response to specific feedback from participants. One common request during the tests was for the ability to select and reposition handwritten content or drawings. As a result, a lasso tool was implemented in the final version. This feature allows students to outline, select, and move elements on the canvas, enhancing flexibility in editing and organizing answers.

In addition, some participants expressed the need to copy and reuse drawn elements, especially in tasks requiring repeated visual structures. Based on this feedback, a copy tool was added, enabling users to duplicate and paste content within the canvas.

Feedback from testing in the text view also led to the resolution of a bug where a deleted character would reappear after switching between questions, ensuring consistent behavior when editing answers in the text input.

To support better control over the workspace, a clear page tool was introduced. While undo functionality already existed, testers found it inefficient when wanting to erase all content from a page. The clear page tool allows students to start over with a single action. A picture of the toolbar with the different tools is illustrated in figure 4.10.

During the testing, exam security and fairness was identified as an important aspect to address. In response, the final application also includes a time management system to control access to the exam. Examiners define the date, start time, and end time for each exam in the teacher portal. This information is stored in the database and linked to the exam ID. When a student attempts to log in, the application checks this data to determine if access should be granted. If the student is too early, too late, or the exam date is incorrect, access to the exam is blocked.

To make these restrictions clear, several warning modals have been implemented. These notify the student immediately if the login attempt is made outside the allowed time frame or if the exam data is missing. In situations where the login is made on the correct date but before the start time, the student is instead redirected to a waiting screen. This ensures that the user remains within the app environment and avoids confusion while waiting for the exam to begin. Additional modals are also shown if an invalid anonymous code is entered, helping to guide students during logins.

To ensure that exams end in a controlled way, the application includes an automatic submission mechanism. If the exam duration runs out while the student is still working, the interface locks and displays the submission screen.

As another aspect of exam security, participants raised questions about how resubmissions or exam recovery would be handled. In response to this, a restriction on duplicate exam submissions was implemented, ensuring that each student can only submit once, reducing the risk of cheating.

Additionally, the need for better control when an exam is interrupted, for example due to technical issues, was identified. As a result, a recovery code system was implemented, requiring proctor authorization to resume an interrupted exam, adding an extra layer of security to the process.

Some participants also highlighted that the absence of a visible timer made it difficult to keep track of the remaining time. To improve this, a countdown timer was added, accompanied by a 15-minute warning message before the exam ends.

Additionally, visual changes related to the scroll button in the draw view were made after the usability testing. During the testing, many participants experienced confusion regarding the button, finding it difficult to locate and understand how to use it, which can be seen in section 4.1.3 in table 4.4. As a result of this, the button was redesigned as a "toggle button" instead of a regular button. The toggle button switches between two different labeled states: "Draw" and "Scroll," with the currently active option highlighted. Another issue regarding the scroll button was the position of the button, left-handed participants activated the button unintentionally while writing with the pen during the testing, which also can be seen in 4.1.3 Identified Usability Issues in table 4.4. As a result, the button itself was also relocated to the top left corner, instead of the bottom left corner. The result of the toggle button can be seen in figure 4.3.

The final interface design also includes a submission pop-up dialog when handing in the exam. The pop-up dialog includes a warning if the student has unanswered questions left, a check-mark that the student needs to check before being allowed to submit, and a note that informs the user about not being able to edit the answers once submitted. This was done as a result of the feedback regarding that the participants were unsure about whether the answers had been handed in correctly in section 4.1.3 Identified Usability Issues. The result can be seen in figure 4.9.

Below, in figure 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 4.8, 4.9, and 4.10, the user interface of the final examination application is presented, including improvements based on feedback from the usability testing, as well as improvements to meet exam requirements. Additionally, figure A.5, A.6, A.7, A.8, and A.9 illustrates the updated version of the teacher portal which can be seen in Appendix A.

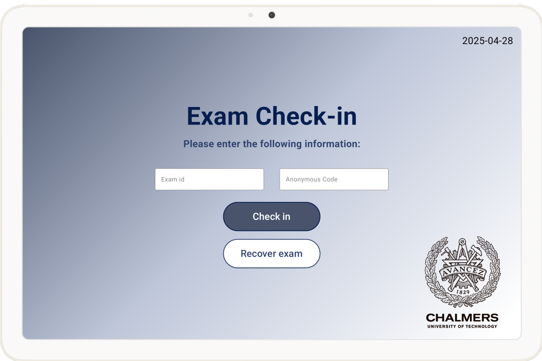

Figure 4.1: Exam Application: Login page.

Figure 4.2: Exam Application: Recovery page.

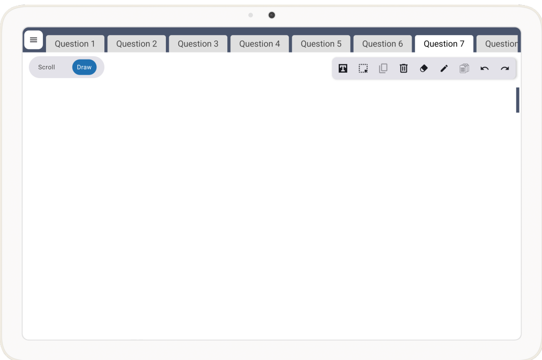

Figure 4.3: Exam Application: Draw view.

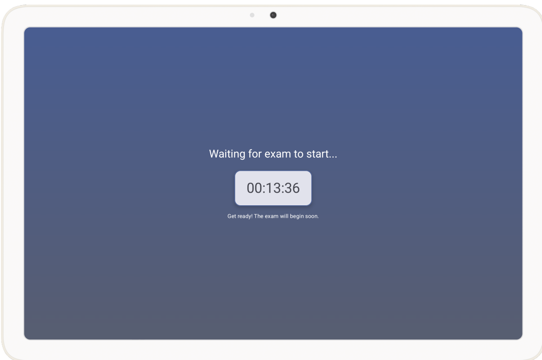

Figure 4.4: Exam Application: Waiting screen.

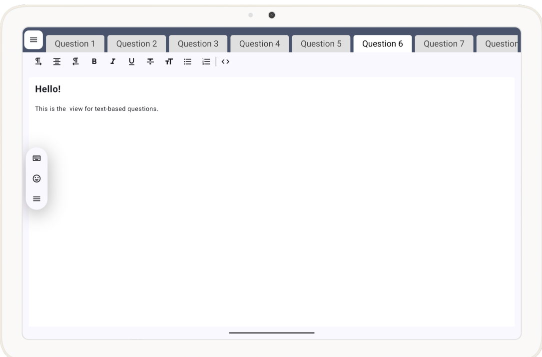

Figure 4.5: Exam Application: Text view.

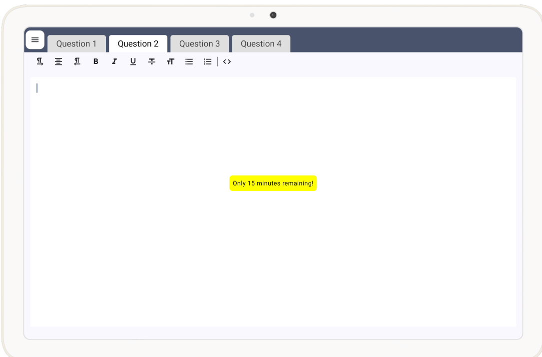

Figure 4.6: Exam Application: 15-minute warning view.

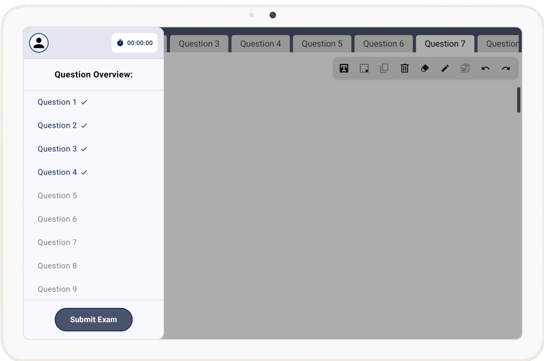

Figure 4.7: Exam Application: Sidebar.

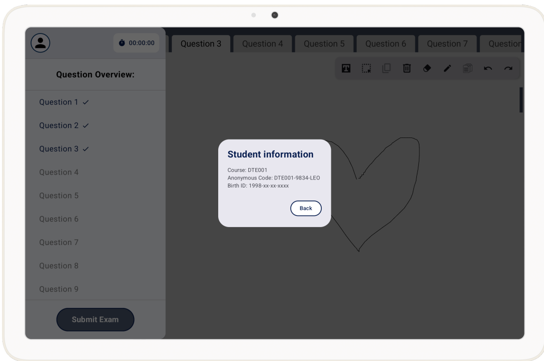

Figure 4.8: Exam Application: Student information.

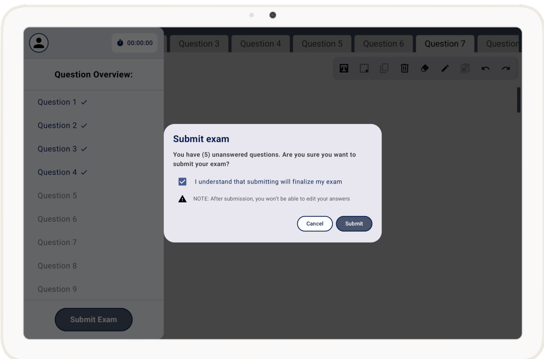

Figure 4.9: Exam Application: Submit view.

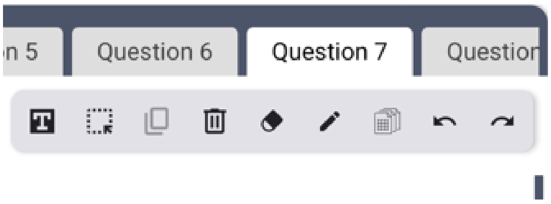

Figure 4.10: Exam Application: Toolbar. Exam Application: Toolbar. Tools from left to right: 1) Textfield, 2) Lasso, 3) Copy, 4) Clear, 5) Eraser, 6) Pen, 7) Paper, 8) Undo, 9) Redo.

5
Discussion

In this chapter, the development process and the final outcome of the application will be discussed. The results from usability testing and how they influenced the prioritization of features and the final product will be explained. An analysis on how the initial prioritization was set up, how it evolved based on user feedback, and evaluating whether our early assumptions about missing functionality were accurate or not. Additionally, development insights from working in sprints and ethical considerations tied to the application's intended use will be presented. This chapter ends with a summary of the limitations of the current solution and suggestions for future improvements.

5.1 Reflections on the Technical Architecure

As previously mentioned, the project was initially designed to support only Android tablets. However, the choice of Kotlin as the programming language makes future expansion to other platforms significantly easier. Kotlin's support for multiplatform development allows shared code to run across Android, iOS, web, and desktop applications. This means that if the project scope expands or shifts to other platforms in the future, the transition will be smoother thanks to the language's cross-platform capabilities.

Kotlin not only simplifies development but also encourages code reuse and consistency. Since most group members had prior experience with languages like Java or C, many object-oriented programming concepts such as classes, inheritance, and interfaces were already familiar. Kotlin's syntax is designed to be intuitive for Java developers, offering a more concise and expressive alternative that enhances readability and reduces boilerplate code. However, Kotlin's smaller developer community compared to Java presents certain challenges. There are fewer tutorials, examples, and community-driven resources available, and some libraries, particularly in specialized or niche areas, are less mature. This can result in additional time spent implementing basic functionality from scratch, which limits the time available for developing more interesting or complex features.

While the Kotlin language itself was relatively easy to learn, understanding the inherited codebase posed a greater challenge. As noted earlier, the application was not built from scratch but it was inherited from a previous project group. Although starting fresh was an option, choosing to build upon the existing code was the preferred option due to the teams limited initial experience and the constrained development timeline.

This decision brought its own set of complexities. In addition to learning Kotlin, the team needed to interpret the architectural choices and logic made by the previous team. Much of their code had been written under time pressure, which impacted readability and maintainability. A particular difficulty was the high cognitive complexity in many parts of the code, where large blocks of logic were contained in just a few files rather than being modularized. Specifically, the core functionality was heavily concentrated in two files: DrawingScreen and TentaViewModel.

One clear area for improvement would be refactoring the code into smaller, more manageable functions and adopting a more structured approach to feature implementation. However, rewriting the entire application from the beginning would likely not have produced better results at that stage. The insights gained during the development process had not yet been acquired, and starting from scratch would probably have introduced similar challenges. Furthermore, a significant amount of time would have been required to replicate existing functionality, likely resulting in fewer implemented features.

One of the key architectural choices made by the previous group was to route all database interactions through a centralized API. An alternative approach that was considered was to handle database interactions directly within the Exam interface. Coupling the server-side logic directly with the Exam interface could have made sense if it were the only client interacting with the database. However, since the teacher portal also needed access to the database to create and manage exams, a decoupled architecture was clearly more appropriate. Separating the backend logic from both clients ensured greater flexibility and scalability, which made retaining the centralized API an easy and well-justified decision during development.

Another observation made was that the API was implemented in JavaScript, whereas the Exam Interface was developed in Kotlin. Ideally, both the API and the client could have been written in Kotlin, as Kotlin is not only well-suited for mobile development but also supports server-side applications, including API development. Using a single language across the entire codebase would have improved consistency, reduced context switching, and made it easier to understand the overall system architecture. However, since more team members were more familiar with Node.js and JavaScript than with Kotlin for backend development, it was decided to retain the existing API implementation rather than rewrite it in Kotlin. As mentioned in section 2.1.1 API, Node.js is a performant runtime to use on the server so the team was confident that load on the server was not either going to be a problem.

5.2 Reflections on the Usability Testing

Through usability testing, a lot of valuable feedback was gathered, as a result, some usability issues that otherwise would go unnoticed was discovered. During application development, usability tests played an important role in the development process and helped polish the product into its final version.

A total of 15 participants took part in the usability testing. Due to the limited availability of only two setups with tablets, pens, and keyboards that had to be shared among group members, usability testing was conducted with individuals from personal networks, as this was the most practical solution. This approach made the testing process both faster and smoother than it would have been if usability tests had to be arranged and scheduled with unfamiliar participants.

As mentioned earlier, research shows that 5–8 participants are often sufficient to uncover most usability problems [14], which suggests that our sample size of 15 participants was was relatively large. However, because most participants were recruited through the team's personal and professional networks, the group lacked full diversity. To compensate for this potential bias, the number of participants was increased to 15. This was done to ensure more reliable results and to reduce the risk of overlooking important usability issues.

If this project were to be redone, the aim would be to recruit a more balanced mix of participants, including participants both familiar and unfamiliar to the team. This would strengthen the qualitative and quantitative aspects of the test data. Testing with a broader range of participants would also help ensure more objective feedback while still preserving the open communication that the think aloud method benefits from. Some of the participants in the usability testing already had a basic idea of the project being developed, and a few of the participants had even seen early versions of the application. This familiarity with the application might have influenced their feedback and overall experience during testing. For this reason, a more balanced mix between know and unknown people would likely be a better approach to the usability testing.

However, it is also important to acknowledge that there were certain advantages with testing only on people that were acquainted. In particular, this seemed to be specifically beneficial when using the think aloud method. By knowing the participants on a personal level, it appeared to help participants feel more comfortable speaking freely and sharing their thoughts as they completed the tasks. Based on experience, all participants were sincere and had no problem communicating, both positive and negative thoughts during the tests. It is believed that this approach benefited the quality of the qualitative data, which might not have been obtained if testing had been conducted with unfamiliar participants. Despite the relatively small test panel, and consisted consisting solely of known participants, it can be confirmed that obtaining valuable insights about the results was important.

Another important aspect of usability testing was that only one iteration could be conducted within the time frame of this project. At the beginning of the project, the interview with Maria from GU (see section 1.3.1 Digital Examination Platforms and the DISA Project), together with the Figma designs, user stories and early discussions, contributed to shaping the MVP version of the exam application. The MVP was used for usability testing, and after gathering feedback, it was refined into the final product. However, an additional iteration of usability testing would be particularly beneficial for this project. Conducting a final test on the refined version could help ensure that the results from the initial testing were accurately interpreted and that the implemented changes effectively addressed the identified issues and improved the product.

5.3 Development Insights and Feature Prioritization

The usability tests significantly influenced the direction of the development process. Initially, the feature prioritization list was based on experiences group members had with other digital examination tools and drawing programs. Since most people in the group consist of computer science and IT students, these priorities reflected what was thought users would need most. From the team's perspective, additional editing functionalities were wanted, as these features were familiar from other drawing programs.

Thus, the core features implemented prior to user testing generally functioned as intended, as reflected in the participant scores presented in section 4, Results. However, the feedback from usability testing showed that some features that were previously rated as medium or even low priority by the team turned out to be much more important for other users. For example, the confusion around switching between scroll and draw mode, and the lack of a confirmation dialog before submission, were not emphasized in the original plan but became central usability issues during testing. Participants also requested features such as a clearer overview of answered and unanswered questions and the ability to select and manipulate drawn elements, suggesting that tools like the lasso and two-finger scrolling were more important than what had been anticipated.

As a result, priorities were changed after the usability testing accordingly. The team implemented features like the submission confirmation modal, clearer navigation overview, improved scroll handling, and the lasso tool, all based directly on participant feedback. These changes reflect a shift from an assumption-based roadmap to a user-informed development strategy.

The decision to let the feedback guide development was driven by the realization that initial prioritization lacked the diversity of real exam-takers' perspectives. The usability tests brought in participants with varying technical backgrounds and experiences. By adapting to the feedback, the final version of the application better addressed real user needs and frustrations than the original plan would have allowed.

5.4 Reflections on the Final Examination Application

Allowing the results from usability testing to guide refinements led to a more intuitive and secure final examination application than the initial version. One of the most impactful changes made to the application was changing the scroll button to a toggle button. This change not only clarified the tool's functionality but also enhanced the overall interaction experience. Additionally, the relocation of the toggle button to the top left corner, instead of the bottom left corner, resolved issues for left-handed users who previously activated it accidentally, which leads to a more inclusive design.

The improved version of the submission pop-up dialog is now more informative and hopefully increases users confidence when submitting exams. By clearly indicating any unanswered questions and requiring explicit confirmation before submission minimized the risk of user uncertainty and unintentional submission errors.

Another important aspect of the final examination application was the change in navigation. The decision to include a side menu was initially influenced by the version of the application received at the start of the project. In that version, the side menu primarily served as a navigation tool, allowing users to switch between questions.

To avoid disrupting the user's main workflow when navigating, the side menu was repurposed to give an overview of question progress and to provide a space for essential information such as the countdown timer and student information button. The side menu now clearly indicates which questions are answered or unanswered, meanwhile scrollable tabs for navigating between questions were placed directly at the top of the canvas view. By solving it this way, users are able to switch between questions without leaving the main workspace, enhancing their focus and efficiency. At the same time, the side menu became a dedicated space for buttons and the countdown timer that did not fit or were not suitable within the main canvas area. This way ensuring a cleaner and less cluttered interface.

Furthermore enhancements to the exam's overall security were made. One important improvement was the introduction of the recovery code, which must be provided by a proctor to resume an interrupted exam. This prevents cheating methods such as intentionally triggering a recovery, to log in to another person's exam mid exam or students attempting to log into each other's exams using anonymous codes. Additionally, the countdown timer with a final 15-minute warning further supports a realistic and time-aware examination environment.

The final version also took into account some of the emotional challenges students face during examinations. Stress and time pressure are common, to help address these issues, features that support time management were implemented. For example, a 15 minute warning appears near the end of the exam to help students stay aware of the remaining time. Another example is the countdown timer that is visible in the side menu, allowing students to keep track of time. It is intentionally kept out of the main canvas view because some students might become stressed when constantly seeing a timer.

Furthermore, the addition of advanced drawing tools, such as the lasso tool, copy tool, and clear page tool, provided users with greater flexibility in how they respond to questions. These tools transformed the application from a basic digital canvas into a more advanced examination environment.

In summary, the iterative approach driven by usability feedback enabled the evolution of the application from a basic prototype into a refined examination tool. As a result significant improvements have been made compared to the original version provided.

5.5 Ethical Considerations

Ethical issues were defined to guarantee a responsible and sustainable approach. Several key factors were discussed in relation to the decisions that determine the final product. Beyond evaluating the final product, the project's advantages and limitations were also reflected to identify its strengths and address potential challenges. This was important for identifying strengths and evaluating how to address challenges, which contributed to shaping a better product that can meet the needs of a wider range of users.

One relevant ethical aspect was inclusion, meaning the product must be designed to ensure accessibility for all users. Particular attention was paid to making the application intuitive for individuals with color blindness. For instance, a white background with a black pen was used to avoid design choices with low contrast between text and background. Additionally, red and green color combinations were avoided due to this aspect of inclusion. Efforts were also made to collaborate with an external organization specializing in usability testing, which could have provided access to a broad network of experienced users. However, this approach proved unfeasible due to the high cost of their services.

Another ethical aspect to consider is inclusion regarding technical knowledge. At Chalmers, students of all ages and varying levels of technical knowledge engage with educational tools. Because of this, efforts was made to develop an app that was simple to use, with a limited number of functionalities to make it more inclusive. However, some users may still lack the technical knowledge to use the system effectively. Although this challenge is difficult to fully resolve, a future implementation of a tutorial pop-up could help by guiding users through the basic functions of the application. The design choices to support users with functional variations, such as the ability to enlarge text or use font types that are easier to read for individuals with dyslexia, were also explored. While this was not implemented in the current version, these accessibility improvements are recommended for inclusion in future iterations of the application.

Another important consideration was how the examination process with tablets would be implemented in practice. After evaluating the ethical and logistical implications, Chalmers decided that if the system is adopted, tablets will be provided to all students during exams. This decision eliminates the need for a BYOD model, which could have introduced significant fairness and accessibility issues. Requiring students to use their own devices would risk disadvantaging those without access to compatible or reliable hardware, potentially creating financial inequities and undermining the integrity of the examination process.

By providing standardized devices, Chalmers ensures that all students have access to the same technical environment, which supports a more equitable and secure exam experience. It also reduces the risk of students bypassing security features or accessing unauthorized resources. While supplying and managing a fleet of tablets introduces additional costs and logistical challenges, these are outweighed by the benefits of enhanced exam security, increased fairness, and a more consistent and controlled digital examination process.

5.6 Future Work

After the development and testing phases were completed, several additional features were requested by users. Although there was not sufficient time to implement them during the initial project timeline, these features remain important for the overall improvement of the application. In addition, there are broader design considerations and underlying system changes that will need to be addressed in the future. This section outlines and discusses the most significant of these proposed enhancements, explaining how they will contribute to a more complete application and how they bring it closer to the goal of deploying the application in real examination settings.

Due to time constraints, several planned features could not be implemented, particularly those aimed at expanding creative functionality and enhancing existing tools. One of the primary features left out was the addition of a fill tool. It would let users quickly colour enclosed areas with a single click, saving time compared to manually colouring with the pencil. Furthermore, it would also contribute to fewer canvas elements being used, which would in turn improve application performance, a topic that will be addressed in more detail later.

Another planned enhancement involved improving the pencil tool by introducing a multi-colour option to boost creativity and improve the user experience. Initial colours could include blue, yellow, and green, while red should be excluded since it is used for grading and comments. Thereafter, a real-time outline or shadow to show the size and area affected by the pencil and eraser tools would be needed. This would improve precision, control, and usability, helping users avoid mistakes during detailed work. This feature was also requested by users during the usability testing conducted in sprint 1. Lastly, the current delete button removes all elements without warning, though the action is undoable. Adding a confirmation modal would prevent accidental deletions. Additionally, users should be able to delete only selected areas using the select tool, offering more precision and reducing reliance on the eraser, which, as will be discussed later, contributes to making the application slower.

One such incomplete feature is the inclusion of multiple question types. At present, the application still lacks support for multiple-choice questions and a coding environment. These features remain important for future development, as they would significantly improve the application's overall usability. The impact of adding new question types is already evident through the successful implementation of text-based questions, as confirmed during usability testing and discussed in the Final Application chapter. Expanding the range of question types would help ensure that tasks typically done on paper can be carried out just as easily, or even more efficiently, within the application.

In addition to this, several limitations in the text view still remain after fixing a bug where a deleted character would reappear after switching questions. For example, bullet points currently revert to normal text after pressing enter instead of continuing the list, interrupting the formatting flow. The formatting toolbar also uses icons that were confusing to the participants. Users were also unable to highlight a specific section of text to apply formatting, such as making the text bold. Improving these areas would make text input more intuitive and closer to standard text editing experiences.

Another area for improvement is how the exam questions are presented to students. While the current application relies on paper questions, a future development would possibly be to integrate exam questions into the application's drawing area. Since the space of the tablet is limited, displaying the entire question at all times would reduce the workspace available for students to write their answers. The team believes a more efficient solution could be a button that allows students to show or hide the question as needed, providing them with a more uncluttered workspace. This expansion would enable paperless exams and create a fully digital examination, simplifying logistics for proctors and examiners, and removing the need to print exam material. As mentioned in the scope, integrated questions were not included in the current application due to priorities. The current design prioritizes an application with a focus on the writing experience, making integrated questions less suitable at this stage. Such integration could offer positive features such as interactive elements, adjustable text size, enhanced image clarity, and inclusive tools such as dyslexia friendly reading or high contrast themes.

Related to this, while integration of digital questions would have been possible on the current tablet, it is still important to consider if other models could offer a better solution. Since the current tablet model provides a relatively small screen space, displaying questions in the application could disrupt student's work during the exam. To solve this issue, it might be appropriate with larger screened tablets. This would however require larger investments from schools or universities, this because larger models tend to be more expensive. Here it becomes important to evaluate if this is a preferred solution, especially since the current tablet is a cost effective option, which makes it attractive for organizations.

Another feature that was not implemented in this project was the integration with Chalmers' servers for data storage. The previous workgroup recommended replacing Firebase with Chalmers' servers to improve security, reduce costs, and support long-term sustainability. However, since this project prioritized feature development and usability, the server integration was considered outside the current scope, as mentioned earlier. Nonetheless, this remains an important area for future development. In addition, integration with Ladok for retrieving results and verifying student registration or anonymous codes remains unimplemented. Connecting the application and teacher portal to Ladok would allow exam information and student data to be automatically retrieved and managed. This would help reduce administrative tasks and ensure that the system always uses up-to-date information. That said, such integration involves several steps to address data protection and security concerns, due to the sensitive nature of the data involved. Despite these challenges, the integration would offer significant benefits for both teachers and students.

The ability of the application to store multiple exams on the SD card without overwriting existing progress was another feature that was not implemented. Right now, exam progress is saved locally to the

SD card to allow recovery in case of unexpected interruptions. However, if a new user begins an exam on a card that already contains another student's data, the existing data is erased due to a mismatch in user IDs. To address this issue, each student's progress should be saved in a separate folder or "bucket" on the SD card. This would make it possible for multiple users to store and resume their exams independently, without the risk of overwriting each other's data.

Lastly, as noted earlier, the toggle between drawing and scrolling modes confused some users. To make the scrolling somewhat more intuitive, the team chose to implement the toggle button instead of a regular button to make it more intuitive.

Ideally, the most intuitive solution would have been to allow scrolling directly with the use of the fingers. However, because the current setup allows drawing with both fingers and the pen, this approach was not feasible without creating conflicts.

One solution to this could be to implement the "two-finger-scroll", to make it possible for the users to scroll in a more intuitive way using two fingers. However, while this improvement falls within the scope of the project, it would require fundamental changes to how tools and interactions are managed. As a result, despite appearing to be a minor visual update, it would demand a considerable amount of development time. For this reason, it was not prioritized during this phase of the project. However, given that users specifically requested this improvement and most core functionality is already in place, this issue should now be addressed as soon as possible. Doing so would align the application more closely with user expectations and remove a long-standing temporary workaround.

Additionally, drawing, erasing, and adding shapes all contribute to an increasing number of objects on the canvas, as mentioned earlier. This did not become apparent during user testing, as those sessions focused on short interactions and basic feature evaluation only. In a real exam setting, however, where students may work continuously for several hours, the growing number of canvas objects can start to degrade performance. This becomes especially apparent when using tools like the selection tool. Therefore, it is important to improve how these canvas objects are managed to ensure smooth performance, regardless of how much content has been added. Without such optimizations, the application may suffer from stuttering or slowdowns on the tablets used, leading to a subpar and potentially frustrating user experience.

6
Conclusion

In conclusion, this thesis demonstrates that the development of an efficient and intuitive digital examination platform for tablets is achievable. The platform serves a greater variety of educational fields by offering digital writing, drawing, and text editing, along with secure storage of students' answers. Compared to traditional methods, the application provides a more flexible and efficient solution, which benefits both students and teachers. It ensures that exam data is handled in a secure and organized manner, with backup mechanisms that minimize risks in the event of technical problems. While the MVP effectively provides the core features, it also highlights key areas for improvement, such as enhancing accessibility features and introducing more advanced administration tools. Further development could focus on integration with external systems like Ladok and utilizing the department's own servers to enhance security and reduce costs. Usability testing was a crucial part of the process to ensure that the final result aligned with practical user needs rather than technical concerns. As the platform grows, ethical factors, including privacy, equity, and diversity, will remain essential. Although this project may not currently be used as a real-life examination application, it clearly highlights the potential to create a functional and intuitive examination system that can be conducted on tablets. This work aims to contribute to the broader shift and transformation of higher education toward more adaptable, accessible and intuitive assessments.

Bibliography

[1] R. J. Martin, M. Liebherr, R. Boumann, S. Schweig, F. E. Kracht, and D. Schramm, "Digital, online, take-home – university students' attitude towards different examination formats", *IEEE German Education Conference (GeCon)*, 2022. DOI: 10.1109/GeCon55699.2022.9942732.

[2] M. J. Hauge, "Improved method of creating automatically assessed diagram problems for exams", M.S. thesis, NTNU, 2022.

[3] H. Al-Saadi, A. Emanuelsson, Ö. Gündogdu, K. Ha, and C. Long Tran, "Tentamina applikation", Chalmers Tekniska Högskola, Göteborg, Sverige, Project report, 2025.

[4] L. Jerdenius, "The university of gothenburg aims to digitise 65% of all exams in 2019", *Inspera Blog*, 2018. [Online]. Available: https://www.inspera.com/digital-transformation/gothenburg-case-study-blog-post/.

[5] C. T. Högskola. "Before examination". Accessed: 2025-04-02. (2025), [Online]. Available: https://www.chalmers.se/en/education/your-studies/examinations-and-other-summative-assessments/before-examination/#.

[6] M. Sunnerstam, "Digital examination (dex): Införandeprojekt 2015–2016", Göteborgs universitet, Tech. Rep., 2016, Upphandling och införande av universitetsgemensamt stöd för digital salstentamen. [Online]. Available: https://gunet.sharepoint.com/:b:/s/sy-grp-pil-web/EaDk82Rp_adKjlMKxtxD7AgBtfTeapkj8uSvDTzHIiHAUg?e=aE3A8e.

[7] S. Ahmed, *A comparative analysis of node.js (server-side javascript)*, https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1004&context=csit_etds, Master's thesis, St. Cloud State University, 2015.

[8] MDN, *Express/node introduction*, 2025. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/Introduction.

[9] Firebase, *Firebase realtime database*, 2025. [Online]. Available: https://firebase.google.com/docs/database.

[10] Google, *Jetpack compose overview*, 2025. [Online]. Available: https://developer.android.com/jetpack/compose.

[11] Google, *Guide to app architecture*, 2025. [Online]. Available: https://developer.android.com/jetpack/guide.

[12] J. Tidwell, C. Brewer, and A. Valencia, *Designing Interfaces: Patterns for Effective Interaction Design*, 3rd ed. Sebastopol, CA: O'Reilly Media, 2020.

[13] I. D. Foundation, *What is usability?*, 2025. [Online]. Available: https://www.interaction-design.org/literature/topics/usability.

[14] J. Nielsen and T. K. Landauer, "A mathematical model of the finding of usability problems", in *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, ser. CHI '93, Amsterdam, The Netherlands: Association for Computing Machinery, 1993, pp. 206–213, ISBN: 0897915755. DOI: 10.1145/169059.169166. [Online]. Available: https://doi.org/10.1145/169059.169166.

[15] J. Nielsen, "Heuristic evaluation", in *Usability Inspection Methods*, J. Nielsen and R. L. Mack, Eds., New York, NY: John Wiley & Sons, 1994.

[16] H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 5th ed. Hoboken, NJ: John Wiley & Sons, 2019.

[17] M. W. Jaspers, T. Steen, C. van den Bos, and M. Geenen, "The think aloud method: A guide to user interface design", *International Journal of Medical Informatics*, vol. 73, no. 11, pp. 781–795, 2004. DOI: 10.1016/j.ijmedinf.2004.08.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1386505604001820.

[18] M. Cohn. "User stories". Accessed: 2025-02-11. (2025), [Online]. Available: https://www.mountaingoatsoftware.com/agile/user-stories.

A

Appendix: Figures

Exam: Digital Exam 1

Course code: DTE001
Examiner: X
Date: 2025-XX-XX
Anonymous code: DTE001-9834-LEO

Q0
Start by logging in to the exam.

Q1 Drawing Question

Choose the paper layout that suits you best.

Then, draw two circles as round as you can.

Also, add a text box and write something in it.

Figure A.1: User-test exam, displaying question 0 and question 1.

Q2 Text Question

Write a short text about what you have eaten today, with a suitable heading.

Feel free to use bullet points and bold text where appropriate.

Q3 Drawing Question

Skip this question.

Q4 Drawing Question

Change the paper to grid paper and draw a triangle. Then, undo the triangle and draw a rectangle instead.

Undo the rectangle as well.
After this, is there a way to recover the rectangle again?

Figure A.2: User-test exam, displaying question 2, 3 and 4.

Q5 Text Question

The exam proctor approaches you and asks you to identify yourself.

Can you find information about yourself somewhere in the app?

Write down your student ID number as your answer to this question.

Q6 Text Question

Write the letters A to F with 5 lines of spacing between each letter.

Q7 Drawing Question

Imagine that you are working on a very long calculation.

What you draw or write here does not matter.

Explore how it would feel to have to write a very long answer using the scroll function.

Figure A.3: User-test exam, displaying question 5, 6 and 7.

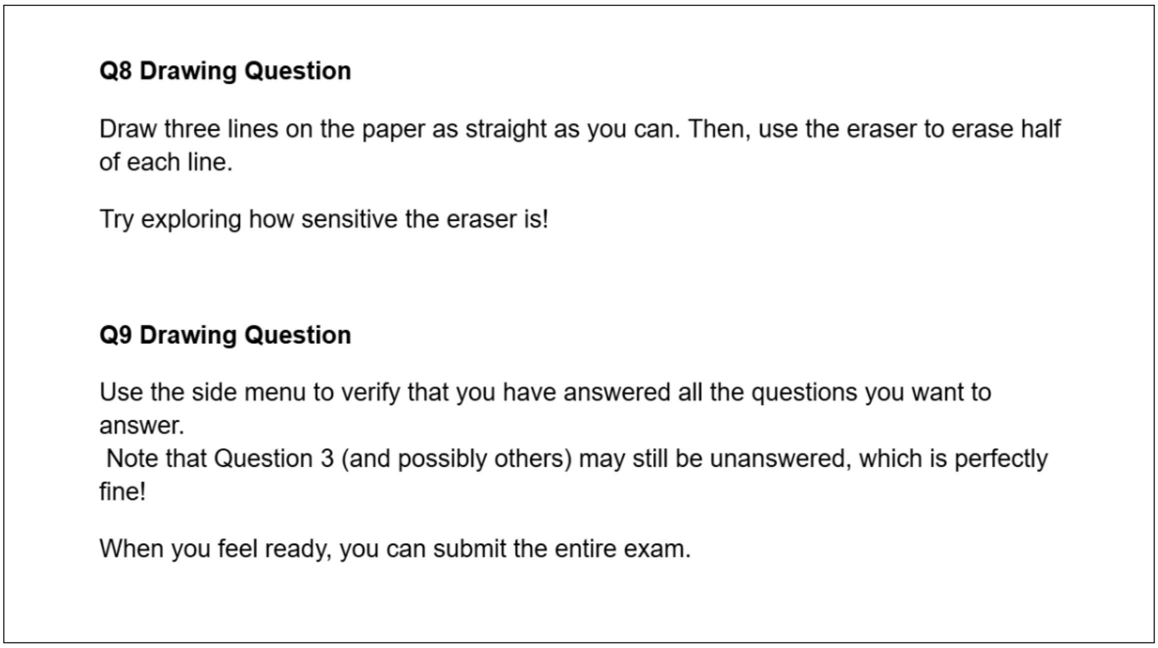

Figure A.4: User-test exam, displaying question 8 and question 9.

Figure A.5: Final Teacher Portal: Login view.

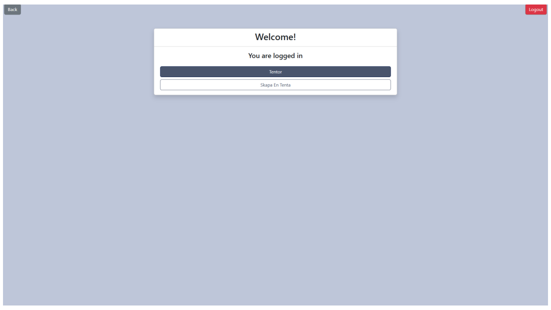

Figure A.6: Final Teacher Portal: Post login view.

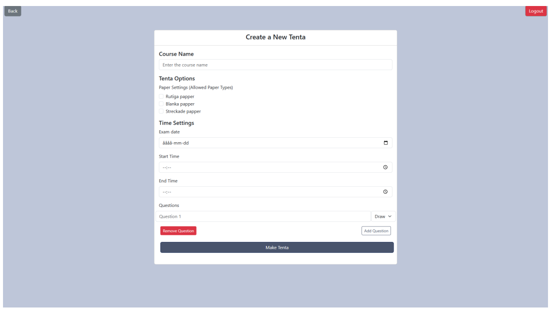

Figure A.7: Final Teacher Portal: Create exam view.

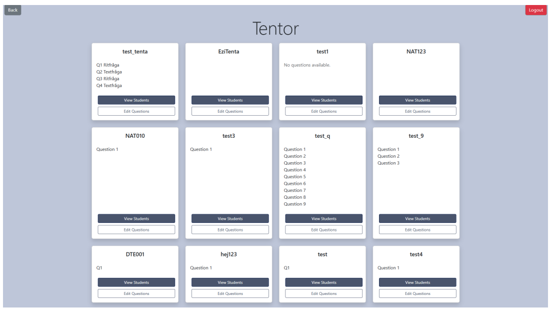

Figure A.8: Final Teacher Portal: Created exam view.

Figure A.9: Final Teacher Portal: Download exam view.

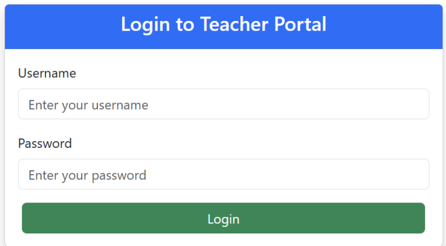

Figure A.10: Initial Teacher Portal: Login view.

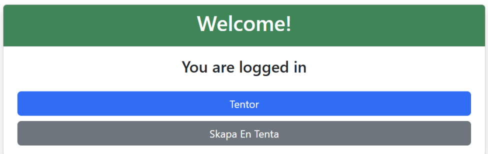

Figure A.11: Initial Teacher Portal: Post login view.

Figure A.12: Initial Teacher Portal: Create exam view.

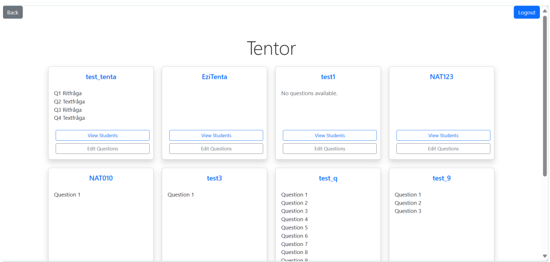

Figure A.13: Initial Teacher Portal: Created exam view.

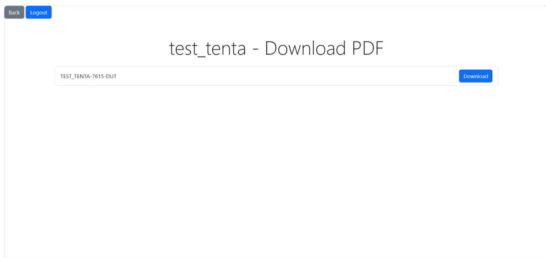

Figure A.14: Initial Teacher Portal: Download exam view.

B
Appendix: Tables

Table B.1: User story descriptions.

User Stories
As a student, I want to be able to easily navigate between questions without losing my work, so that I can focus on answering the exam efficiently.
As a student, I want to be able to undo and redo changes in my answers, I want them to be limited to the question I am currently on so that I do not accidentally undo actions in a question I am not actively viewing.
As a student, I want to be notified about the remaining time, so that I can manage my time effectively during the exam.
As a student, I want to use different background styles (e.g., graph paper for math exams), so that I can format my work properly.
As a student, I want to be able to easily draw geometric shapes and write mathematical symbols so that my answers are clear and precise.
As a student, I want to receive error messages if something goes wrong with editing my work or handing in my submission.
As a proctor, I want a clear and secure way to handle recovery requests so that I can authorize only legitimate cases.
As an examiner, I want to define exam start and end times, so that the exam follows a structured schedule.