Master's thesis

Petter Grø Rein
Tore Stensaker Tefre

Creating a Web Application Supporting Git in Software Development Courses in Higher Education

Master's thesis in Informatics
Supervisor: George Adrian Stoica
June 2021

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

NTNU
Norwegian University of
Science and Technology

Petter Grø Rein
Tore Stensaker Tefre

Creating a Web Application Supporting Git in Software Development Courses in Higher Education

Master's thesis in Informatics
Supervisor: George Adrian Stoica
June 2021

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

NTNU
Norwegian University of
Science and Technology

Abstract

Software engineering is one of many subjects taught in universities and colleges all over the world. The teaching often includes group-based projects, as many students will participate in group-based projects after graduation. This is also true at the Norwegian University of Science and Technology (NTNU), where several courses related to software development and software engineering include group-based practical activities. The learning management systems (LMS) used in many of the institutions teaching software development does normally not support functionality facilitating practical work in software development courses, creating a need for additional software, routines, and work tasks for the course staff in such courses.

Git repository hosting services such as GitLab and GitHub are often a part of the practical activities in courses related to software development, creating several possibilities and a number of challenges for both course staff and students. The combined use of LMS' such as Blackboard and the aforementioned Git repository hosting services often creates time-consuming tasks requiring time and attention from the course staff.

This study aims to augment the usefulness of Git repository hosting services in the teaching of software development. This has been done through the use of a design and creation strategy, resulting in a minimum viable product of a web application that helps course staff manage software development courses and give them insights about how their students of such courses work and learn.

Evaluated through expert interviews and a focus group, the application delivered as the artifact of the project were found to have a good potential for improving the software development education at NTNU, and possibly also in other contexts. The experts who participated in the project expressed a belief that the web application could be a useful contribution to others with the same challenges, something the team agreed to. Therefore, the source code of the web application has been published using open-source licensing terms to ensure that others may use or improve the web application according to their needs.

Keywords: software development education; course management; Blackboard; Gitlab; Git in education; teaching with Git

Sammendrag

Programvareutvikling er eit av mange fag som blir undervise på høgskular og universitet verda over. Undervisinga inkluderer ofte gruppebaserte prosjekt, ettersom mange studentar vil delta i gruppebaserte prosjekt etter fullført utdanning.

Dette er òg tilfelle ved Norges teknisk-naturvitskapelege universitet, der mange emne relatert til programvareutvikling har gruppebaserte aktivitetar som ein del av læringsopplegget. E-læringssystema som stettar undervisinga på mange av institusjonane som underviser programvareutvikling har vanlegvis ikkje funksjonalitet som stettar undervisinga av programvareutvikling, noko som skapar eit behov for utfyllande programvare, rutinar og arbeidsoppgåver for emnestaben i slike emne.

Tenester for lagring og handtering av Git-strukturar, slik som GitLab og GitHub, er ofte naudsynte for å fasilitere praktiske aktivitetar i emne knytt til programvareutvikling, noko som både skaper moglegheiter og utfordringar for studentar og emnestab. Kombinert bruk av e-læringssystem som Blackboard og dei tidlegare nemnde tenestene for Git-strukturar fører ofte til tidkrevjande oppgåver som krev både tid og ressursar frå emnestaben.

Denne studien siktar mot å auke den moglege nytta av å bruke tenester knytt til Git-strukturar i emne relatert til programvareutvikling. Dette har blitt gjort gjennom bruk av strategien «Design and creation», som har resultert i ein web-applikasjon som gjer det enklare for emnestaben i slike emne å administrere emne og ved å gje dei betre innsikt i korleis studentane i slike emne arbeider og lærer.

Resultatet av prosjektet har blitt evaluert gjennom fleire ekspertintervju og ei fokusgruppe, og applikasjonen har basert på denne evalueringa eit godt potensiale til å forbetre undervisninga av programvareutvikling på NTNU, og truleg òg i andre kontekstar. Ekspertar som har blitt intervjua har ytra eit ynskje om at webapplikasjon blir tilgjengeleggjort som fri programvare, noko opphavspersonane er einige i vil kunne vere eit godt bidrag til utdanningsinstitusjonar og andre, som kan dra nytte av arbeidet som er gjort, og vidareutvikle det etter behov. Kjeldekoden til webapplikasjonen er difor publisert fritt tilgjengeleg på internett.

Acknowledgements

We would like to thank our supervisor George Adrian Stoica for using his time and broad knowledge to guide and advise us throughout the project. We would also to thank our fellow master students for exciting discussions and feedback throughout the year.

This project would not have been possible to complete without help and insights from the people at IDI NTNU and NTNU IT, and we would like to thank them for their time and for their support and interest in the project.

We would also like to thank the participants in the project, providing their time, essential insights, knowledge, and experiences critical for the project to be successful.

Last but not least, we would like to thank our family and friends for being supportive and showing genuine interest in the project.

Contents

List of Tables

List of Figures

Acronyms

API	Application Programming Interface
CE	Community Edition (GitLab CE)
CLI	Command-Line Interface
ECTS	European Credit Transfer and Accumulation System
FAB	Floating Action Button
FS	“Felles Studentsystem” (The Common Student System)
HTML	HyperText Markup Language
IDI	Department of Computer Science at NTNU
ITGK	Informasjonsteknologi, grunnkurs (Information Technology, Introduction)
LA	Learning Assistant(s)
LMS	Learning Management System
MVP	Minimum Viable Product
NSD	Norwegian centre for research data
NTNU	Norwegian University of Science and Technology
PAT	Personal Access Token
REST	Representational State Transfer
RQ	Research Question
SLR	Structured Literature Review
SotA	State of the Art
SUS	System Usability Scale
TA	Teaching Assistant
URL	Uniform Resource Locator
WCAG	Web Content Accessibility Guidelines
WWW	World Wide Web

1 Introduction

1.1 Context

The Norwegian University of Science and Technology, commonly referred to as NTNU, is one of the leading institutions in Norway for educating new software developers and engineers, graduating a three-digit number of them every year.

Software developers and engineers often work in teams. A lot of the practical experience the students at NTNU have with programming before graduating comes from software engineering courses. Designing and developing software applications in groups is the essential learning activity.

During these group work courses, one of the most crucial activities for the academic staff working with the course is to make sure that the groups are working as intended, an exercise proving challenging in courses with several hundred students admitted. Learning assistants are, of course, heavily involved with the groups, but even they sometimes struggle to have a sufficient impression of how the groups are working together.

To ease the management of courses and increase learning outcomes, both the learning management system (LMS) Blackboard Learn, commonly referred to as just Blackboard, and the Git repository hosting service GitLab Community Edition is used for teaching software development processes to students at NTNU. However, none of them make it easy for academic staff to easily facilitate and monitor software development in groups for several hundred students, making room for potential improvement of the situation.

1.1.1 Teaching of software engineering at NTNU

All students of Informatics and Computer Science at NTNU take part in TDT4140, where "students will demonstrate that they are able to plan and manage small software engineering projects using agile methods such as Scrum and XP (extreme programming), and how to contribute to project teams as programmers, testers, managers, designers, documenters, architects etc." [1]

Students enrolled in the program Informatics also participate in IT1901, which "gives knowledge and skills in agile application development in teams." [2]

In courses such as TDT4140 and IT1901, the focus is on teaching students to use Git, as defined in section 1.1.4, as efficiently as possible and using issues and milestones. Issues are tasks defined by the group that should be done in a project, and milestones are a method used to organize and track issues and set dates for when issues should be finished. Services such as GitLab often also provide the functionality of having issues and milestones and link them with changes done by participants in Git.

1.1.2 Hierarchy of Academic Staff in Software Engineering Courses at NTNU

At NTNU, including the Department of Computer Science, a hierarchy of academic staff typically exists within a course. While assistants in the courses are often referred to as Teaching Assistants (TA) at many institutions, some of the relevant courses at NTNU refer to the assistants in software development courses at NTNU as Learning Assistants (LA). This approach has been described in a paper released in the proceedings of the conference "Læringsfestivalen 2020" an overview of the hierarchy for the course "TDT4110 - Information Technology, Introduction", commonly referred to as ITGK at NTNU. This hierarchy is similar to the hierarchy in other courses at NTNU, where software development in groups is taught and is presented in Table 1.

Role	Description	Example: ITGK 2019
Teacher staff	Professors/associate professor(s) in charge of planning, teaching, and assessing the course.	8 professors/associate professors were involved, teaching the course in six parallels.
Head LA(s)	Student(s) with a minimum of 180 ECTs and usually has experience from LA positions. In charge of implementing the assignments/projects, organizing the labs, and exercise lectures. They can sometimes be Ph.D. students. Employed in a 50% position.	4 master-level students and 2 PhDs with at least two years of experience in the course.
Teaching LA(s)	Student(s) with at least one year of experience as a LA. In charge of answering emails, publishing information, coordinating the labs, giving feedback, etc. Employed for 10 hours/week.	12 students with between 1-3 years of experience as LAs. Pairs of LAs were given specific responsibilities, such as managing Blackboard, checking and publishing assignments, tech support, lab organization, etc.
Student LA(s)	Student(s) who have taken the course before. Usually, they need to have received a C or higher. In charge of answering questions, assessing assignments/projects, manning the labs, etc. Employed for 6 hours/week.	114 students, whom each had the responsibility of 18-25 students. They are in the labs for 6 hours every week. All students must have their assignments approved by their LA. These assignments do not count towards their grade.

Table 1 – The LA hierarchy at IDI in example course ITGK [3]

1.1.3 Blackboard at NTNU

Blackboard [4] is the LMS currently in use at NTNU. Blackboard was elected to be the new LMS of NTNU in 2016 [5], following the merger of NTNU and the university colleges in Gjøvik, Ålesund, and Trondheim.

Being an LMS, Blackboard offers functionality for course staff and students so that they can exchange information, create and deliver exercises, and discuss their courses. Features from Blackboard Ultra enabling remote education through video lectures have also been introduced during the covid outbreak.

When Blackboard was introduced at NTNU, the university stated that the selection of a new LMS would mostly be a minor change from the previous LMS, but that selecting Blackboard as their new LMS would include added functionality related to social learning activities and standard interfaces towards other data sources and web services. [6]

1.1.4 Git

Git is a version control system facilitating the tracking of changes done in files and directories [7]. Git was developed by Linus Torvalds when he was working on the Linux kernel and needed a distributed system for tracking changes done by the various developers of the Linux project.

Git enables the developers in a project to copy, commonly referred to as "clone" or download a "repository", which can be seen as a type of folder containing project files, down to their own computer. The downloaded repository contains the code and history of all the changes done to the project. A developer can make changes to the project and save the changes into the projects as a "commit". A "Commit" is an entry in the history graph of the project that contains the changes done, a title, and a description made by the developer when he saved the "commit". The changes will then still be only on the developer's computer. However, as Git is a distributed system, it enables you to "push" or upload your new commits to the place you initially downloaded them from. The initial place mentioned is typically the service that provides the distributed functionality of Git, some popular alternatives being GitLab, GitHub, and Bitbucket.

Making work in progress directly on the project is not good when others are working on it concurrently. Git provides "branches" for this problem. You take the main branch of the project and "branch" out into a local branch of your own. A "branch" is a pointer to a snapshot of your changes, and when you are done with what you were working on, you can "merge" the branch containing your changes with the main branch.

The Git repository hosting services GitLab and GitHub, and most others also offer the functionality of issues, milestones, and pull/merge requests. Issues are a way you can collect user feedback, report bugs, and organize tasks. Often in an issue, you can have discussions about the task or specific code in the project. Milestones are usually a collection of issues that the developers have grouped and set a due date for. Milestones provide information about the project's progress, and you can see which issues need to be done or already are done. Merge requests are used when you have created a branch of the project, and you want it to be merged with another branch, and you want others to see it. You can then create a merge request on the Git repository hosting services used in the project with a description of the work you have done, a link to issues finished, and a comparison of your branch and the branch you want to merge into. The other developers in the project can then easily see your work and discuss it if they wish. This is done because you often will let the other developers in a project see the changes you have made and approve them. Commits, issues, and merge requests may be linked together if the commit title or description includes the ID of the created issue. This will provide a link between completed work and what issue it concerns. Afterward, when you create a merge request, you will automatically get a list of issues worked on in the merge request.

1.1.5 GitLab at NTNU

GitLab [8] is one of several Git repository hosting services aimed at storing Git repositories and adding related functionality. Most Git repository hosting services are provided as a cloud service, but GitLab has the option to host your own instance and have a free price tier option. An advantage of hosting your own instance is that you have much greater control of your users' data and that it is possible to integrate your existing authentication service.

IDI at NTNU has deployed its own instance of the GitLab Community Edition, commonly referred to as Gitlab CE. GitLab CE is open source, but GitLab Enterprise is built on top of it, containing closed proprietary code and an extra set of functionality.

GitLab CE is used at NTNU to manage Git repositories in practical activities such as student projects and coding exercises in courses related to software development and engineering. The IDI GitLab CE instance aims to facilitate those courses, but anyone with an NTNU user account can use the instance to host their Git repositories freely.

1.2 Motivation

The project started as a master thesis proposal at IDI, and the task presented was the following:

"The goal of this project is to design and develop a web-based user interface for supporting educational activities in courses that use GitLab for student projects. Typically courses that use GitLab for managing student groups projects could benefit from having support for setting up and initialize tens or hundreds of repositories, visualizing activity, harvesting snapshots. The application should leverage the GitLab REST API and potentially Blackboard API to offer the needed features to the staff working in such courses. The student(s) will work to design, implement, and evaluate a web application that can be used by professors and teaching assistants. The students should contribute with novel and original ways to deal with the various requirements and especially with the visualization of the data throughout the semester. The students will have to work with the main stakeholders for eliciting requirements.

The project involves a study of research and relevant literature on similar software, design, and implementation of a functional prototype and evaluation of the developed prototype at different levels, including user testing."

It was expressed from course staff that much time was spent administrating instead of teaching and that a new tool may provide features and information that could save time and improve the course. For the teaching staff, tasks could be automated, and they can focus on the lectures. For the LA's, better information about their groups can be given to provide better feedback and follow the progress of their groups better. All this will give the student a better experience, and hopefully, they will learn more.

Based on our knowledge about the State of the Art solutions, they didn't meet the needs of the course staff, or were not applicable in the context at NTNU, or were a hassle to use. Therefore, we saw this project as an opportunity to look into how existing solutions were designed and worked. This insight combined with data from our user-centered process focusing on course staff to design and implement a new and hopefully more useful solution.

As students ourselves, we know how confusing it can be to use Git, and we hope we can investigate solutions that will make all the courses a better experience for everyone.

1.3 Research Questions

The main goal and objective of this project was decided to be the design and implementation of a minimum viable product (MVP) of a web application that supports educational activities in courses that use the Git repository hosting service GitLab and the learning management system Blackboard for student projects.

Initially, our problem was investigating how an application can augment the usefulness of source code repositories hosting platforms for practical activities related to software development courses. However, after researching our problem and getting feedback from various stakeholders, the team pivoted into having two research questions in this project, as listed below.

Research question 1 (**RQ1**):

- How can the process of managing software development courses where both learning management systems and Git repository hosting services are involved be improved?

Research question 2 (**RQ2**):

- How can course staff in software development courses be helped to better supervise and guide their students and student groups by utilizing data in Git repositories?

1.4 Findings

As described thoroughly in section 7.1, there are strong indications that the course staff in software development courses at NTNU are excited about using a combination of Git repository hosting services together with the web application developed as part of this project to improve the practical activities in their software development courses.

The results of the project can be seen as the achievement of the goals defined throughout the project, as well as how well the two research questions have been answered. The evaluation has shown that the application designed and developed as part of the project seems useful for course staff at NTNU, indicating that it also could be useful for similar situations in other contexts.

The research questions of the study have also been answered through the web application, all the way from the process of designing and implementing it to the expert evaluation. The mentioned evaluation has together with other observations made the team fairly confident that the web application both improves the process of managing software development courses where both learning management systems and Git repository services are involved, referring to RQ1, while also helping course staff better supervise and guide their students and student groups by utilizing data in Git repositories, referring to RQ2.

1.5 Outline of the Report

Before the chapters start, the report contains an abstract in both the English and Norwegian language, together with acknowledgements and a list of all the acronyms used throughout the thesis.

Chapter 1 starts with an introduction of the context and the motivation for the thesis project and introduces the research questions and the report outline.

Chapter 2 contains a description of the chosen research method for the project. The description describes the overall process, the data gathering methods used, how the gathered data was used, alongside information about participants and the ways they have been involved in the project.

Chapter 3 presents the State of the Art review and our brief literature review. The review looked into the existing state of the art, sometimes referred to as SofA, and papers and problems related to the project.

Chapter 4 contains information about the data gathering methods that happened before the development of the application started. This includes interviews conducted and explains what kind of information was obtained during the interviews and how it could be of use for the later work in the project. The chapter also includes information about how the requirements for the application were defined.

Chapter 5 describes the design and development iterations. The chapter includes data about new functionality in each sprint, feedback from users in each sprint, and information about the choices and changes made throughout the implementation process.

Chapter 6 contains the evaluation of the application the project evolves around. The evaluation was formative and summative and includes evaluation methods such as expert interviews, user tests, and a focus group.

Chapter 7 presents the results of the project and discusses several aspects connected to the project. This includes the user-centered design process, the agile development process, the literature review, and the structure of the work together with the participants and stakeholders, together with the potential for future use of the project.

The thesis concludes with chapter 8, containing the conclusion itself. The chapter includes a summary of the results and an explanation of the work that has been carried out throughout the project. A section presenting the possibilities for future work is also included and should be of interest to students looking to build on the work of this project.

2 Research Method

In this chapter, the research method of the project is discussed. The project is based on the "Design and creation" research method, following the five steps described in detail in this section: awareness, suggestion, development, evaluation, and conclusion.

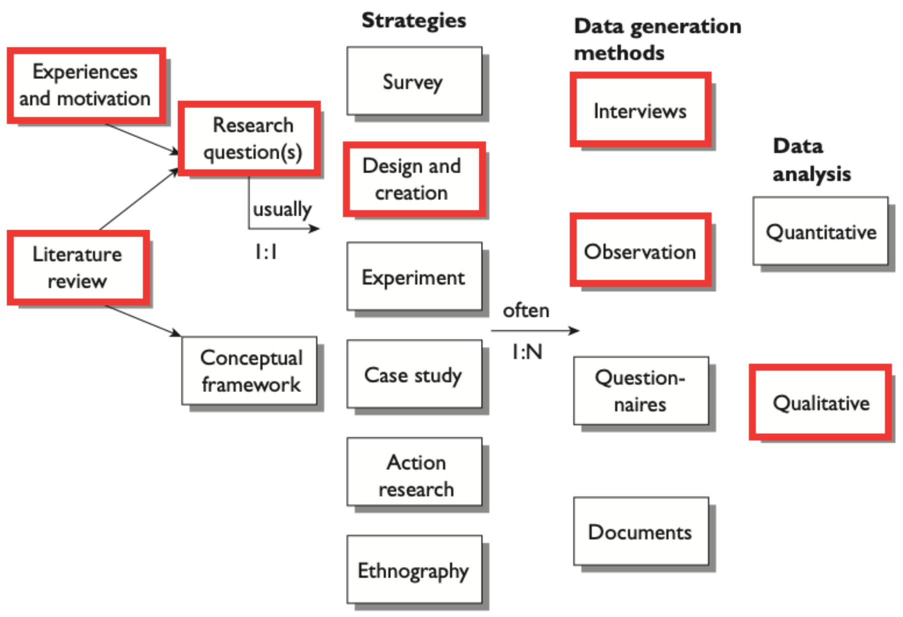


Figure 1 - Research Methods framework from Oates [9, p. 33]

As presented in Figure 1, the research method is based on a number of data gathering methods. A number of interviews in various forms have been carried out as part of the project, including expert interviews and a focus group, together with observation-based data gathering methods such as user tests. The project has also included a review of the current State of the Art related to the challenges presented in the two research questions of the project. A description of the user-centered design approach and the agile development process is also included in the chapter, together with an assessment of the ethics related to the project.

2.1 Design and Creation Process

The project is based on the design and creation research strategy. The research strategy uses an iterative process involving five steps: awareness, suggestion, development, evaluation, and conclusion. These are explained in the following sections and are illustrated in detail through the Gantt chart in Figure 2.

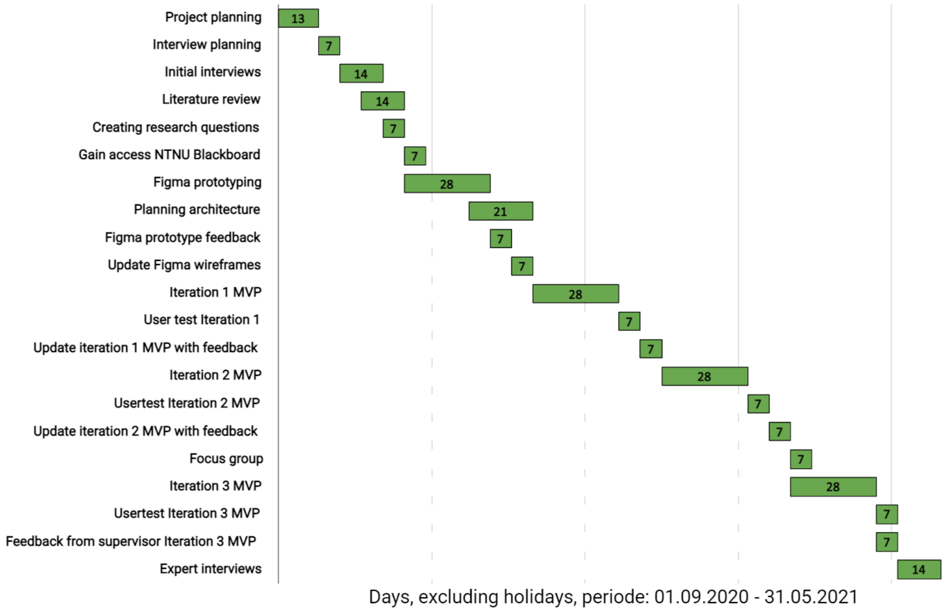


Figure 2 - Gantt chart of project schedule

2.1.1 Awareness

The first step in the research project was to become aware of the problem, which comprises the first of the five steps in the iterative design and creation research process: awareness. The step is described by Oates as follows:

"Awareness is the recognition and articulation of a problem, which can come from studying the literature where authors identify areas for further research, or reading about new findings in another discipline, or from practitioners or clients expressing the need for something, or from field research or from new developments in technology.” [9, p. 111]

This project started with our supervisor defining a project proposal that could be changed before starting the project and deciding the research question. After joining the project, both literature and similar solutions were explored, and interesting papers and attractive solutions were examined. However, it was decided that the project still had potential, as the existing solutions were determined to have a potential for improvement. The problem was decided to exist, and the research questions was decided to represent the problem well.

2.1.2 Suggestion

The second step in the research project is suggestion. According to Oates, “Suggestion involves a creative leap from curiosity about the problem to offering a very tentative idea of how the problem might be addressed.” [9, p. 112]

In this project, our research about papers and existing solutions became important, together with our own creativity. Several stakeholders and potential end-users were also involved in the project and participated in interviews. This made us even more aware of the problem, and some of the interview participants also had ideas on how the problem could be approached.

Several potential approaches were considered, but based on similar solutions, research, our own skillset, and potential, it was decided that we believed that the problem could be approached by designing and creating a web application prototype that integrates with Git repository hosting services and learning management systems. This was also the proposal initially created by our supervisor before the start of the project. A further explanation of why a web application prototype was decided to be the best approach is included in section 4.9.

Based on our approach, a research question was developed and articulated, resulting in expanding the literature review and search for similar solutions.

2.1.3 Development

During the third step of the research project, "the tentative design idea is implemented" [9, p. 112]. This project resulted in a minimal viable product of a modern web application developed in three iterations based on a user-centered design process. This made it possible to adapt to expectations and desired functionality while maintaining an efficient system development speed.

2.1.4 Evaluation

During the fourth step, the web application prototype was evaluated. As suggested by Oates, this is done by assessing the worth of the web application prototype and any deviations from expectations. [9, p. 112] During this step, we were eager to find out if the web application prototype could be of use in software development courses and if it has deviations from possible expectations or not.

The evaluation was done using several methods as described in chapter 6, including user tests, a focus group, and expert interviews.

Participants in the evaluation were selected based on their relevant background and involvement in NTNU courses where software development was taught in one or more ways.

2.1.5 Conclusion

During the fifth and last step of the design and creation step, "results from the design process are consolidated and written up, and the knowledge gained is identified, together with any loose ends – unexpected or anomalous results that cannot yet be explained and could be the subject of further research." [9, p. 112]

In this step, it became clear that the project had great potential for further work, discovered throughout the project because of various needs in software development courses.

2.2 State of the Art Review

A literature review was performed to find the State of Art solutions. We decided on some topics, used the keywords from the topics to search, evaluate the found papers, revise the keywords until we were happy with the findings. To find papers, Google Scholar was used, Google Scholar let you search multiple databases at once, among ACM Digital Library and IEEE Explore Digital Library. More about our literature review can be found in section 3 State of the Art Review.

2.3 Data Gathering Methods

Inspired by the concept of methodological triangulation, where it is essential to "employ different data gathering techniques" [10, p. 264], it was decided that the project should contain several data gathering methods. As the project is based on containing a user-centered design process, we took inspiration from data gathering techniques commonly used in user-centered design processes, mentioned in the following sections.

2.3.1 Initial Expert Interviews

During the first weeks of the project, two important course coordinators were interviewed about the project. This included the coordinators of the NTNU courses IT1901 and TDT4140.

The interviews were decided to be semi-structured so that the subjects were able to add in on matters they cared about and wanted to talk about. As described by Oates, semi-structured interviews lead to participants being able to speak with more detail on raised issues and introduce issues of their own that are relevant. [9, p. 188] We believed that this would make it easier for us to understand the real challenges we wanted to work with. The purpose of having these meetings was to understand what course coordinators spent their time doing and to get an impression of in what ways we may be able to simplify some processes. The subjects for our meetings were decided in collaboration with our supervisor.

Due to covid and people working from home, only one interview was conducted in person. This interview was taped using the built-in recording app on an iPhone. The most interesting parts were later transcribed, followed by the deletion of the recording. The other interview was arranged using a digital meeting in Microsoft Teams, where the audio was recorded for partial transcription before the recording was deleted.

2.3.2 Focus Group

A Focus group is a form of a group interview. The benefit of a focus group is that it allows diverse or sensitive issues to be raised that might otherwise be missed, for example, in the requirements activity to understand multiple points within a collaborative process or to hear different user stories. [10, p. 272]

2.3.2.1 Participants

As previous data gathering methods have been aimed at professors, learning assistants were decided to be the target population for the focus group. This is connected to the theory of triangulation and related to the previously mentioned concept of "methodological triangulation", where the idea is to gather information from multiple sources and, through the use of various data generation methods [9, p. 37], forming a representative population of possible end-users for the solution.

To help us populate the focus group session, our supervisor provided us with a list of possible participants. At the same time, the team also explored potential participants from other courses to ensure a broader set of experiences was considered and taken into account during the focus group.

As three to ten participants are described as suitable for a focus group [10, p. 271], we decided to aim for a target population of four. This would enable one of the participants to withdraw from the project without causing significant problems for the focus group. In addition, selecting a relatively small population size was also believed to make the focus group planning easier regarding the various covid restrictions applied at the time.

Participant number	Course background	Course role
1	IT2805 – Web Technologies	Head learning assistant
2	TDT4310 – Intelligent Text Analytics and Language Understanding	Head learning assistant
3	IT1901 – Informatics, Project I	Teaching learning assistant
4	TDT4140 – Software Engineering	Student learning assistant

Table 2 - Focus group participants

2.3.2.2 Roles During Focus Group

Literature suggests that focus groups should be led by a trained facilitator [10, p. 271], so one of the project team members took care of the tasks related to being the facilitator. The facilitator's most important tasks are mentioned to be guiding the discussion while encouraging quiet people to participate and stop verbose participants from dominating the discussion. [10, p. 272]

The other team member will also co-operate as a facilitator while contributing by introducing topics in the agenda and demonstrating the application before the second discussion round.

The whole session will be recorded, subject to the participants' approval, and the participants will contribute by defining a summary after each discussion session of the focus group.

2.3.2.3 Location and Setup

Inspired by the idea of using a facilitator to ensure that speech time during the focus group is as evenly shared as possible, we attempted to be strategic regarding the room setup for the focus group. We wanted the participants to feel at home, so we booked a meeting room in a building where all participants are familiar. With regards to the selected room, some things went into consideration. First and foremost, the covid-19 pandemic meant that we had to comply with several rules and recommendations, most notably the requirement that all participants should keep a distance of minimum one meter at all times. Furthermore, a large screen or a projector was needed for the planned demonstration of the product.

After an appropriate room was located and reserved, the participants were informed about the location, and the preparations could proceed. Inspired by how the facilitator should work to distribute speech time during the focus group as evenly as possible, we decided to consider how the participants were seated. As we wanted the discussion to flow as freely as possible if the distribution of speech time did not become an issue, we placed the facilitator and the observer in the back of the room so that if the participants were looking at either the demonstration or each other, they would feel that it was a more casual conversation than if the facilitator and the observer were sitting in the middle. This also seems to be in line with Oates, stating that "The seating should be arranged so that everyone is visible to everyone else, and the researcher should not be in a focal position." [9, p. 195]

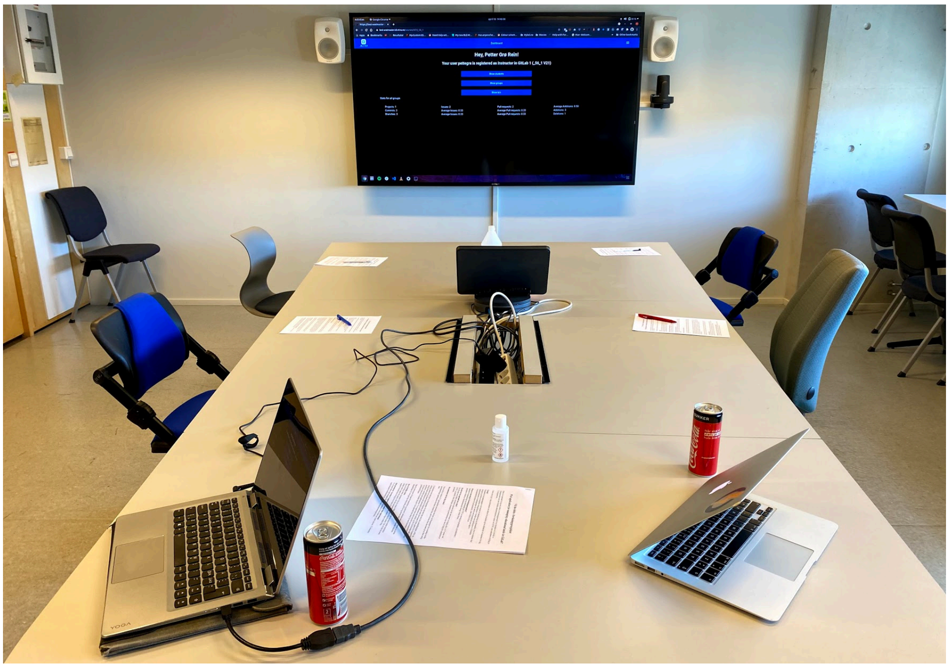


Figure 3 - Focus group setup

2.3.2.4 Program

As suggested by literature [10], the focus group will be based on a predefined agenda but executed with the flexibility to enable participants to focus on what they think is essential.

Topic	**Allotted time**
Project introduction	5 minutes
Format and participant introduction	5 minutes
Discussion about experiences and pain points	20 minutes
Summary of discussion	5 minutes
Presentation of prototype	10 minutes
Discussion about prototype	20 minutes
Summary of discussion	5 minutes
Presentation of plans, ideas, and future work	10 minutes
Discussion about plans, ideas, and future work	20 minutes
Summary of discussion	5 minutes
TOTAL	105 minutes

Table 3 - Planned focus group program

The first part of the focus group contained an introduction with information about the project, the researchers, why the focus group is vital for the project, and the rights of the participants.

After introducing the project, the participants were presented to each other by making each participant of the focus group talk about their background and their connection to NTNU courses.

Following the participant introduction, the focus group pivoted into discussing their experiences from being learning assistants in NTNU courses. The facilitator was here especially focused on the following topics:

- Time-consuming tasks in their courses and roles
- Any pain points related to managing the course
- The participant's role in the courses and their typical tasks and challenges they encountered in their work
- The number of students they worked with and supervised and how they organized their work
- The collaboration between the learning assistants and the course coordinators
- Group management in Blackboard and GitLab
- How they supervised groups in practical activities, and what metrics they were focusing on

The focus group was then scheduled to proceed with a demonstration of the application with functionality from iteration 1 and 2. The participants were invited to discuss the functionality implemented and provide feedback on functionality that could be changed, added, or removed.

After completing the demonstration of the application, the program continues with an overall discussion about the application. For this segment, the following topics were defined as possible conversation starters:

- If the application could be useful in their courses
- If the included functionality serves a purpose
- Functionality that the participants thought could be useful to add
- How the application could impact the education of software development at NTNU

Towards the end of the focus group, we aimed at discussing the future of the project and presented what we wanted to implement during the upcoming sprint:

- Visualization of group activity
- Comparison of activity stats with an average of all students / all group members

We also wanted to propose some ideas for further work and check if they had any feedback. Finally, we also wanted to ask if the participants had any ideas for functionality that could be useful. Our thoughts included the following:

- Initialization of template code in all group repositories, based on an existing Git repository
- Facilitating updating all repositories based on an existing template repository
- Publishing feedback to the group through the creation of issues or merge requests in GitLab
- One-click deployment of the software for group repositories using GitPod or Vercel
- Automated rule-based grading of exercises and deliverables
- Allocation of group members based on given criteria
- Comparison pages with the ability to select multiple students and groups

2.3.3 Expert Evaluation Interviews

Towards the end of the user-centered design process, new interviews with experts were conducted to find the value of the completed work and identify possible future work and applications of the project.

2.3.3.1 Program

The expert evaluation interviews included introductory questions for new participants in the project to understand better their experience and how it may influence their answers in the expert interview.

- What courses have you been involved in, and what was your role in those courses?
- What tools were used in your course for managing students? How did you use Blackboard?
- Did the students use Git in your courses? Was it managed by themselves or set up by the course staff?

The interview then proceeded to a demonstration of the application, scheduled to take around 20 minutes. The demonstration was paused after coherent functionality was shown to ask the expert about the perceived usefulness of the functionality and if the expert had any feedback about functionality that could be added or removed.

Following the demonstration, the experts were asked questions about their overall impression of the application, its usefulness regarding our research questions, and deviations from expectations, if any.

To evaluate whether or not the application creates value for potential users, despite not having possibilities for a real-world user test-based evaluation, it was also decided to seek question inspiration from the System Usability Scale (SUS) [11]. SUS is a widely used standardized questionnaire for assessing perceived usability, reported to have been used in 43% of post-study questionnaires in industrial usability studies [12].

While SUS in its original form is paper-based and consists of respondents participating by answering a form with a Likert scale, we adapted some of its statements into more dialogue-focused questions that better suit our evaluation style. These were included in the interviews when appropriate.

SUS statement	Converted question
I think that I would like to use this system frequently.	Do you agree that the system could have been used frequently in your courses?
I found the system unnecessarily complex.	Do you agree that the system seems unnecessarily complex?
I thought the system was easy to use.	Do you agree that the system seems easy to use?
I would imagine that most people would learn to use this system very quickly.	Do you agree that professors and learning assistants would learn to use this system very quickly?
I needed to learn a lot of things before I could get going with this system.	Do you think you need to learn a lot of things before you could get going with this system?

Table 4 - Questions inspired by System Usability Scale

2.4 Data Analysis

All interviews were recorded after approval from the participants. Recording was done to avoid having to take notes during the interviews, which was avoided because the team believed it could have influenced the natural flow of the interviews.

As the interviews were semi-structured, the participants could talk about aspects of the project they were focusing on. While being very interesting, those aspects were not always very useful related to our questions. The interviews were therefore not transcribed as a whole due to time and resource constraints. However, the interviews gave us an overall impression about the needs, that we were designing and implementing solutions to facilitate, which was the most important motivation for conducting interviews. The recordings of the interviews were also listened to while we defined the initial requirements and needs from the stakeholders. When something of high interest was expressed, we transcribed the section of interest, and some of those quotes have been included in this thesis.

The several data gathering methods used in this research were done to approach "methodological triangulation" [10, p. 264], described as the act of gathering data through the employment of different data gathering techniques. Interviews were done to get insight into how course coordinators and learning assistants work, the insight was used to set the initial goals for the artifact. Furthermore, user tests were conducted to get feedback on the functionality that was being implemented, and the results of the user tests were used to improve the artifact. The expert interviews were done to get feedback on the final artifact and to establish a foundation for analyzing how the artifact meet the goals of the project and this study.

2.5 User-centered Design Process

The project follows a user-centered design process as described in this section. The user-centered design process in this project is inspired by the steps presented in the ISO standard ISO 9241-210 (2010). [13]

2.5.1 Usability

One of the main goals in user-centered design processes is to increase the usability of an application.

Usability is in the standard ISO 9241-11:2018 defined as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [14].

According to Sharp et al. [10, p. 19], usability is broken down into the six goals included in Table 5.

Goal	Description
Effectiveness	A general goal refers to how good a product is at doing what it is supposed to do.
Efficiency	Refers to the way a product supports users in carrying out their tasks.
Safety	Involves protecting the user from dangerous conditions and undesirable situations.
Utility	Refers to the extent to which the product provides the right kind of functionality so that users can do what they need or want to do.
Learnability	Refers to how easy a system is to learn to use.
Memorability	Refers to how easy a product is to remember how to use once it is learned.

Table 5 - Usability goals as explained by Sharp et al.

2.5.2 Personas

Personas are described as "rich descriptions of typical users of the product under development on which the designers can focus and for which they can design products." [10, p. 403] Each persona is characterized by a unique set of goals related to the application and includes some personal details such as the hobbies of each persona so that they appear more like real potential users the application can be designed for.

As Sharp et al. [10, p. 404] mentioned, a persona primarily has two goals: to help the designers make design decisions and to remind the team that real people will be using the product, which is how the personas have been used in this project.

The main inspiration for the personas included in this project has been the insights originating in the initial data-gathering phase of the project, including the interviews and the learning assistant hierarchy presented in section 1.1.2.

2.5.3 Scenarios

Scenarios are described as «human activities or tasks in a story that allows exploration and discussion of contexts, needs, and requirements. [10, p. 408] A scenario typically involves one of the previously described personas created for the project and explains when, where, and how the story involving the persona occurs. The scenario ends when the goal, e.g., what the persona wants or needs to fulfill, is reached.

The scenarios were created to reflect a couple of real-life needs related to the project, discovered through interviews with the stakeholders.

2.5.4 User Stories

User stories were created based on the personas and scenarios created, together with the requirements included in chapter 4. According to Sharp et al., requirements are statements about an intended product that specifies what it is expected to do or how it will perform [10, p. 388], and user stories can be seen as a way to define requirements. A user story represents a small chunk of value that can be delivered during a sprint and may also be used to capture usability and user experience goals. [10, p. 388] The user stories implemented are described in chapter 5.

2.5.5 Design Guidelines

When designing and implementing, the team has been focusing on complying with a set of design guidelines derived from some of the design principles put to paper by Jakob Nielsen and Don Norman, together with the guidelines for accessible design described in WCAG 2.1. [15]

The design principles of Jakob Nielsen origins from a paper published [16] in 1990, commonly referred to as the "10 heuristics", as listed in Table 6.

Heuristic	**Design guidelines for this project**
Visibility of System Status	Users should be informed about what the system can offer, is currently doing, and has completed.
Match between system and the real world	The application should be based on concepts and terminology likely to be already known by the users.
User control and freedom	The users should be able to confirm all changes done by the application before they are done and not complete actions unless confirmed by the user.
Consistency and standards	The terminology and concepts used in the application should be consistent with those used elsewhere, such as in Blackboard and GitLab.
Error prevention	The application should help the users avoid any problems connected to the use of the application.
Recognition rather than recall	The application should give the users the information they need to make good decisions.
Flexibility and efficiency of use	The application should facilitate that experienced users efficiently do their everyday tasks without negatively impacting the user experience for less experienced users.
Aesthetic and minimalist design	The design should not be cluttered with unnecessary details and information.
Help users recognize, diagnose, and recover from errors	The users should be informed about errors if they impact the task they try to accomplish and should be informed about the possible way to deal with them.
Help and documentation	The users should have the information needed to complete their tasks using the application. The application should be designed in a way that makes additional documentation unnecessary for most users.

Table 6 – Nielsen heuristics and derived guidelines

The design principles of Don Norman origins from the book "Design of Everyday Things" [17], and are commonly referred to as "Don Norman's Design Principles", as listed in Table 7.

Keywords	Design guidelines for this project
Visibility	The elements the users most likely should or will focus on should be the most visible elements. Elements that are probably not of interest should be removed or hidden away.
Feedback	The application should inform the user what their action leads to and what consequences this action may have.
Constraints	The application should guide the user into making sensible actions and prevent errors from happening as much as possible.
Mapping	Interactive elements in the application should work in a way that is closely mapped to the concepts they are replicating in a way so that users quickly understand how they can be interacted with.
Consistency	The application should have a consistent design for elements so that the user recognizes the information and possibilities they represent.
Affordance	The application should be designed using elements that are easy to understand the functionality of.

Table 7 - Don Norman's Six Design Principles and derived guidelines

In relation to the WCAG standard the team has been following the guidelines in a best-effort method, and the Wave Chrome Extensions has also been used to detect and identify WCAG issues and errors throughout the application.

2.6 Agile Development Process

Agile development [18] is a development methodology based on iterative development, where discovering requirements and developing solutions evolve in iterations done by self-organizing cross-functional functional teams. The common agile frameworks that have inspired this project's development process are Scrum [17] and Extreme Programming [18]. It is widespread to select parts from the frameworks that fit your needs and not follow them exactly as written. Scrum is designed for teams of ten or fewer that work to solve problems during periods called sprints. Extreme Programming is for two people developing in iterations. An iteration may consist of one or more scrum sprints, depending on the number of problems that need to be solved before meaningful progress has been made.

2.6.1 Development Process

Our development process was divided into three parts, where each part contained two sprints as known from Scrum, each lasting two weeks. From the initial expert interviews in section 2.3.1, it was created a list of tasks or "user stories" that we wanted to work on. The tasks together made up the three main features that we wanted to do in the project.

The main feature and its tasks were assigned two sprints, and the two sprints forming an iteration were done, user testing was conducted, and in some cases, a demo was planned with our stakeholder. This was done to get feedback on the functionality implemented as early as possible from our users to ensure that the functionality is useful and contributes to our internal sprint review.

After a sprint review, we had our retrospective where we discussed the feedback we got and updated whatever necessary of our requirements, tasks, and features we wanted to do next. Sometimes we could reorganize the order in which we wanted to do things, or we could get completely new features we wanted to implement.

Scrum includes daily standups where the development team meets for a short while too often talk about: what work did you do yesterday, what work will you be doing today, and are there any issues stopping you from doing what you want. Daily standups are a way for the team to get the opportunity to update each other, but as we were only two people working on this project, talking to each other every day, we didn't feel that this was necessary to do.

Scrum involves many different roles, and as we were only two people, we had to switch which role we had according to the setting we were in. So, for example, when doing the demonstration for our stakeholders, we had to be product owners. When we had a retrospective, we had to be both product owners and a part of the development team.

2.6.2 Coding practices

2.6.2.1 Usage of Branches in Git

In the project, three different types of branches in Git were used. These included the master branch, develop branch, and a large number of feature branches.

The master branch was our main branch during development that you would branch out from when adding new functionality. But after the delivery of this thesis, it will be the branch containing code from the last time a new functionality was completed and tested to be working and pushed to production. The feature branches were primarily used as working branches while new functionality was being developed and will continue to be after delivery. The develop branches were not used during our initial development. However, they will be where the new functionality is tested after merging in from the feature branches, before being confirmed as working and then merged into the master branch.

2.6.2.2 Pair Programming as Code Review

A practice many know from Extreme Programming as described by Kent Beck is pair programming [20]. In this project, pair programming has been used to increase efficiency when facing challenging issues while developing and to review code as it is being written, saving time in intensive development sprints.

2.7 Ethics

Ethics is a part of every research project, and in a modern Norwegian research project, The Norwegian Center for research data, commonly referred to as NSD, is typically involved.

As suggested by our supervisor, the project applied to NSD for approval before the project started. This was done in August 2020, and approval was received in August 2020 as well.

2.7.1 Consent and Participant Rights

All participation in the project was voluntary. Before participating in the project, participants have been informed about the project, why they are participating, and how they can withdraw their consent to participate in the project.

Participants have also been asked to sign a consent form approved by NSD, this was originally done through the standard paper-based consent form provided by NSD, but like a lot of the research needed to be done without physically meeting the participants, a digital form was created using the Nettskjema web application, where participants could provide their consent after authenticating with their NTNU user accounts.

All participants consented that they could be recognized based on their connection to one or more software development courses at NTNU, and most of them have also consented to being named in the thesis. The naming of participants in the thesis was not done because the team wanted the participants to speak as freely as possible, and there was a belief that participants being named would impact their statements.

2.7.2 Data Processing and Storage

During the project, we tried to limit the amount of data gathered and processed to only include data beneficial to the project.

In the application to NSD, it is necessary to define what personal data will be processed during the project. For our project, the data mentioned in Table 8 was marked for data processing.

NSD Personal data type	**Data processing purpose**
Name	Signature on a written consent form
Email address, IP address, or another online identifier	The email address could be stored for contact reasons
Photographs or video recordings of people	Performing user tests at the IDI usability lab was considered during application to NSD but was not carried out due to covid-19 regulations.
Sound recordings of people	Recording of interviews, focus groups, and user tests so that transcription could happen.
Background data that can identify a person	It could potentially cover great amounts of data, but a connection between participants and NTNU courses and institutes was our primary source of concern.
Other data that can identify a person	The primary concern here is data gathered from interviews and focus groups where statements and quotes can be connected to persons because they might be very specific.

Table 8 - Personal data cleared for processing by NSD

2.7.3 Data Storage

Data related to the project have mainly been stored on the NTNU Office 365 tenant, as described as possible in the Data storage guide published by the IT department at NTNU. [21]

3 State of the Art Review

A review of the current State of the Art was conducted in the starting phase of the project. The primary goal of the literature review was to gather knowledge on what State of Art solution existed, how Git is used in course facilizing/management, and how courses in which you must use Git are taught.

Our study of State of the Art contained RepoBee, GitHub Classroom, SourceControl.me, INGInious, GitInspector, and Git Analyzer. GitHub Classroom are a service targeted at education where the others are tools for doing a specific task that can be relevant in a course setting.

The team have also researched literature on how Git can be used in the context of higher education, presenting some advantages compared to a situation where only an LMS is used instead of a combination containing both an LMS and a Git repository hosting service. Another point of interest in our literature study was how the data from code repositories could be used for supervising students and possibly grading them, a topic touched mainly by the tool GitInspector.

The study represents a good foundation for proceeding to the initial data gathering in this study.

3.1 Method and Search Strategy

While the primary concern in our review was to identify what existed as State of the Art, we also conducted a brief literature review related to our research questions. We wanted to do a review so we could get some information about existing solutions, get some more data points, and be confident that a solution did not already exist before we started to make our research question.

3.1.1 Our literature review method

From the initial proposal and papers provided by our supervisor, we created some topics we wanted to know more about. We wanted to investigate experiences others have with teaching a course where the students needed to learn and use Git, to see if someone had used features of Git to help their course and which tools or features they used, and when reviewing groups, which information from Git they used. Papers about how to teach Git to students were not included in our search.

For the initial search of papers, we used keywords from the topics we created to search and reviewed the papers in the results to see if they fit our topics. If they did, we noted it down, or else we discarded it. Using the initial set of papers we were left with, we investigated their keywords to see if we could include them in our search terms or use them to redefine the terms we had. We looked at the papers' references to see if they also fit our topic and were of interest. We iterated our search terms until we were happy with the numbers of papers, or the results didn't change.

Every topic is in some way related to Git, meaning that some of the search terms resulted in papers for multiple topics, but they were not reviewed if they did not fit the topic we were searching for, but they were noted down and looked into later when we changed the topic.

Criteria for papers to be reviewed, a paper should match at least one or more criteria to be included:

- The paper is about insight into teams or groups using Git
- The paper is about insight into Git data
- The paper is about how Git is used in education
- The paper is about managing large number of Git repositories

Topic	Databases	Search terms
Teaching software development where Git is used	acm.org, ieee.org	"Git classroom", "Git teach higher education"
Using Git to manage software development education	acm.org, ieee.org	"Git teaching tool", "Git teach higher education", "Git analytic"
Measuring work quality in groups using Git data	acm.org, ieee.org, web.wpi.edu	"Git teach higher education", "Git analytic"

Table 9 - Search strategy for literature review

3.2 State of the Art Tools

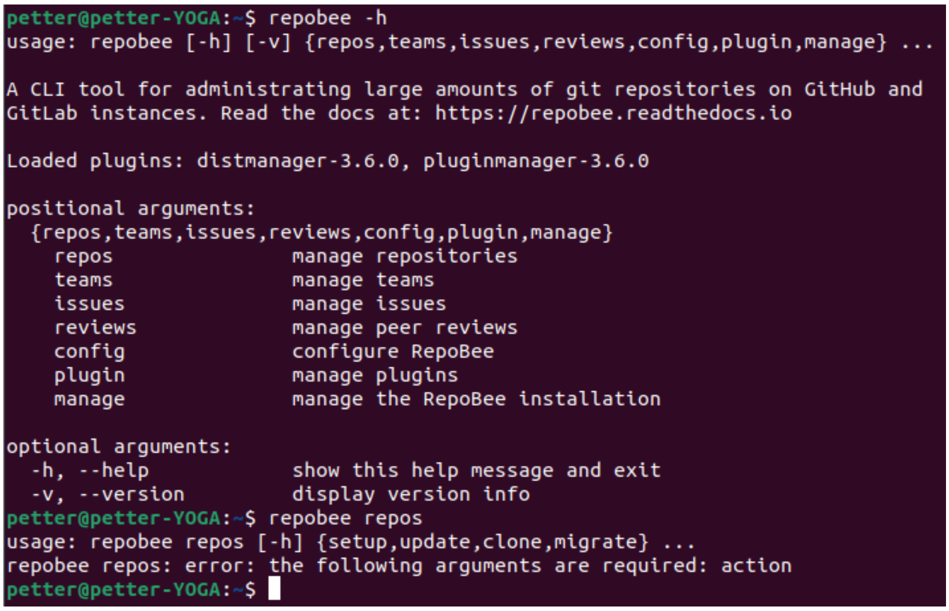
There exist a few solutions to manage a course on a Git hosting platform. But there seem to be only a few that have a GUI or are a Web application. Down below are some selected solutions. They all provide different functionality and do different things. They either have something to do with managing groups and repositories on a Git repository hosting service, showing information and insights about groups on a Git repository hosting service, or giving feedback and evaluation to students.

3.2.1 RepoBee – CLI Tool for Managing Large Groups and Repositores

RepoBee is a command-line tool created by HTH: Royal Institute of Technology [22] that allows teachers and LA's to create large amounts of repositories, clone template repositories with instructions and example code into student's repositories. This is done by using the API of GitLab or GitHub. RepoBee is a generic tool for automation and batch processing of

RepoBee solves the problem of manually creating group repositories for every group in your course, but you must still provide it manually with the roster from the LMS. When setting up the groups, you will get a "Master Organization" this is usually accessible for the whole course. Under this "Master Organization" you will have a "Target Organization", which is a course round. Under this again, you will have the student's repositories.

The missing feature of RepoBee and the downside is the absence of a GUI. It can be a bit of a learning curve to use the command line to manage groups. This is confirmed by Larsén and Glassey, stating that "Another TA noted simply that a weakness was, "No GUI" (TA2). As a next step in the project, we do have plans to generate a GUI from the command-line interface to further reduce the learning curve for both teachers and teaching assistants." [22, p. 539]



```
petter@petter-YOGA:~$ repobee -h
usage: repobee [-h] [-v] {repos,teams,issues,reviews,config,plugin,manage} ...

A CLI tool for administrating large amounts of git repositories on GitHub and
GitLab instances. Read the docs at: https://repobee.readthedocs.io

Loaded plugins: distmanager-3.6.0, pluginmanager-3.6.0

positional arguments:
  {repos,teams,issues,reviews,config,plugin,manage}
    repos               manage repositories
    teams               manage teams
    issues              manage issues
    reviews             manage peer reviews
    config              configure RepoBee
    plugin              manage plugins
    manage              manage the RepoBee installation

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         display version info
petter@petter-YOGA:~$ repobee repos
usage: repobee repos [-h] {setup,update,clone,migrate} ...
repobee repos: error: the following arguments are required: action
petter@petter-YOGA:~$ 
```

Figure 4 - RepoBee command line interface

3.2.2 GitHub Classroom – Web Service for Managing Git Classrooms

GitHub Classroom [23] is a web service GitHub offers that builds upon their Git repository hosting services. You can easily in a GUI import your roster from an LMS, you can create group repositories, you can copy template repositories with example code into group repositories, you can provide feedback in terms of pull requests on the work students generate, you can set up an automated test to provide feedback on assignments. Our impression is that the only thing GitHub Classroom falls through on is in the information and the analysis of groups state and their work, that you are locked into using GitHub, and it does not support Blackboard when the thesis was written. GitHub Classroom could in the future be a complete solution to our problem if NTNU were using GitHub. It is a good starting point for inspiration about how a solution may look like.

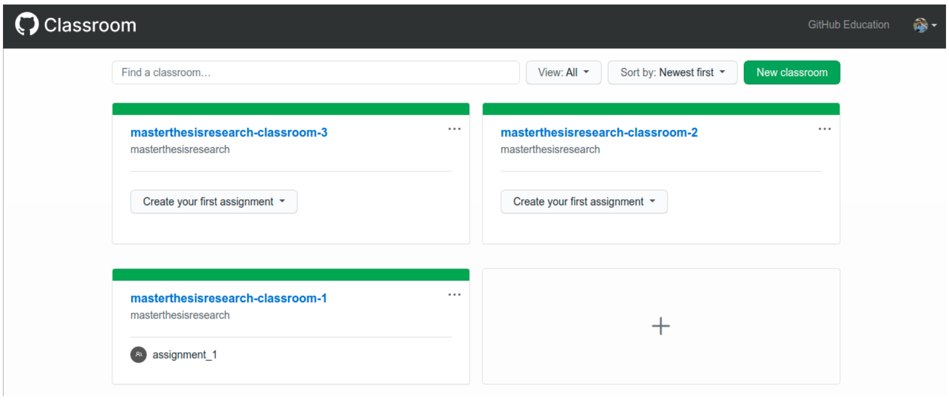
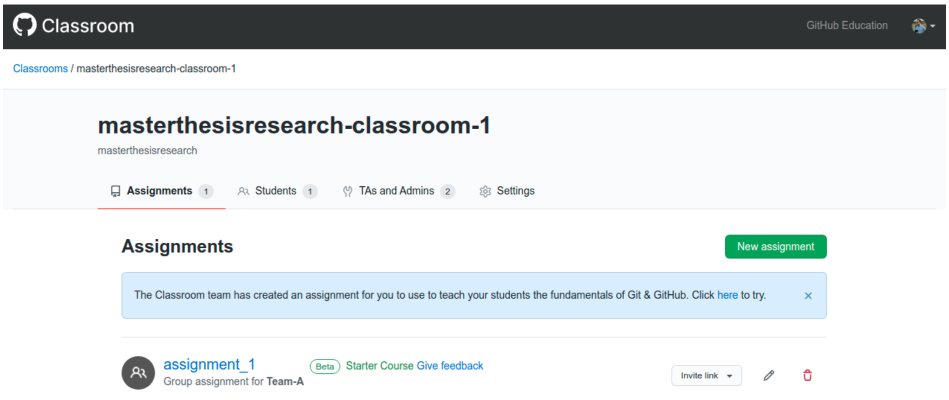


Figure 5 - GitHub Classroom dashboard



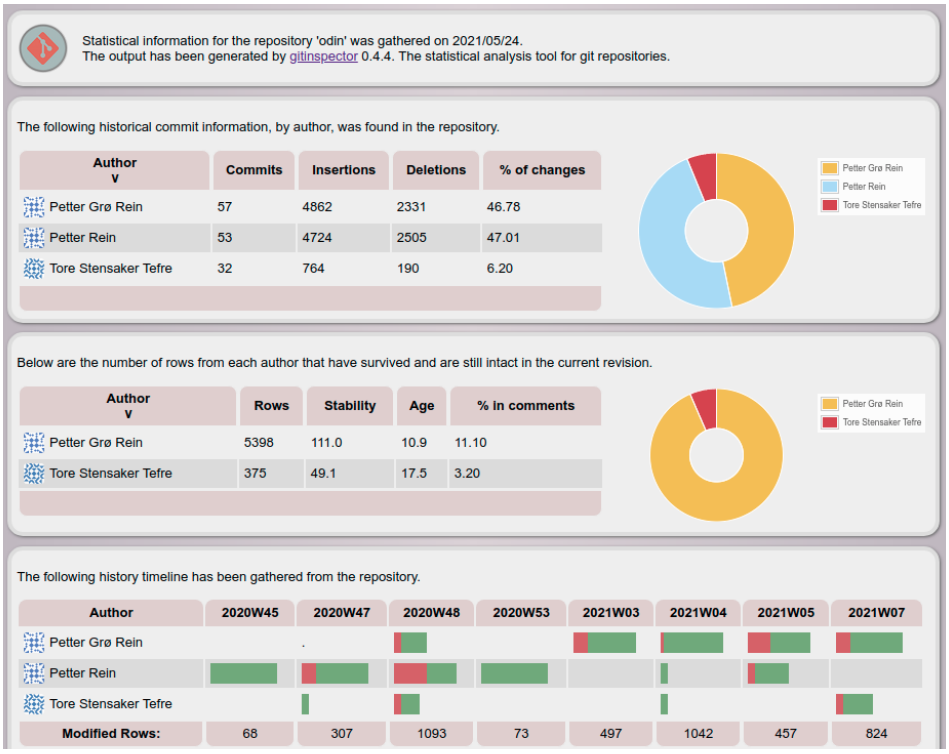
The Classroom team has created an assignment for you to use to teach your students the fundamentals of Git & GitHub. Click here to try.

Figure 6 - GitHub Classroom assignment dashboard

3.2.3 Generic Purpose Git Insights Tools

3.2.3.1 GitInspector – CLI Tool to Harvest Git Data

GitInspector [24] is a command-line interface (CLI) tool to run in a Git repository to get information about the repository at the given time. You only get information about one repository at a time, and you must clone the repository to your computer. The information can be saved and shown as an HTML page. The information/stats that GitInspector has selected to show you can be used as inspiration for what information we wanted to show in our application.



Statistical information for the repository 'odin' was gathered on 2021/05/24.
The output has been generated by gitinspector 0.4.4. The statistical analysis tool for git repositories.

The following historical commit information, by author, was found in the repository.

Author v	Commits	Insertions	Deletions	% of changes
Petter Grø Rein	57	4862	2331	46.78
Petter Rein	53	4724	2505	47.01
Tore Stensaker Tefre	32	764	190	6.20

Below are the number of rows from each author that have survived and are still intact in the current revision.

Author v	Rows	Stability	Age	% in comments
Petter Grø Rein	5398	111.0	10.9	11.10
Tore Stensaker Tefre	375	49.1	17.5	3.20

The following history timeline has been gathered from the repository.

Author	2020W45	2020W47	2020W48	2020W53	2021W03	2021W04	2021W05	2021W07
Petter Grø Rein		.	🟥🟩		🟥🟩	🟥🟩	🟥🟩	🟥🟩
Petter Rein	🟩	🟥🟩	🟥🟩	🟩		🟩	🟥🟩	
Tore Stensaker Tefre		🟩	🟥🟩			🟩		🟥🟩
Modified Rows:	68	307	1093	73	497	1042	457	824

Figure 7 - GitInspector dashboard

3.2.3.2 Git Analyzer – Web Application to Harvest Git Data

Git Analyser is a web application from the University of Wisconsin-Parkside [25], where you could input a Git repository URL, and you would get information about the repository at the given time. You only get information about one group at a time, but they note that supporting retrieving information for a person across an organization or team is planned. The Git repository is "cloned"/downloaded and the information is extracted in a similar matter as GitInspector.

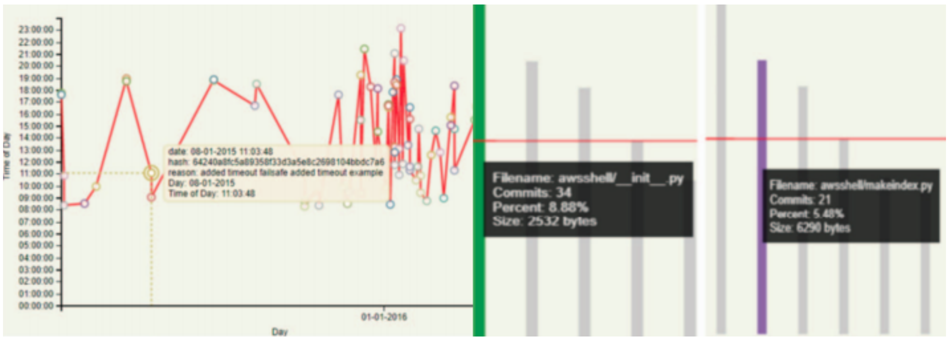
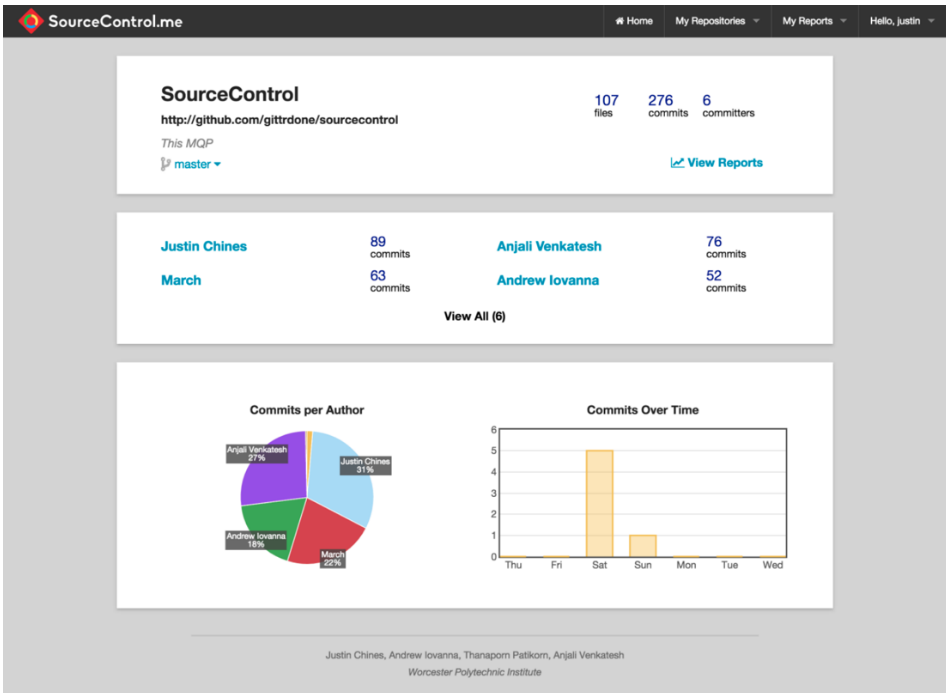


Figure 8 - Git Analyzer charts

3.2.3.3 SourceControl.me – Web Application to Harvest Git Data

SourceControl.me was a web application developed as a bachelor project at Worcester Polytechnic [26], where you could input a Git repository URL, the repository would be "cloned"/downloaded, and you would get information about the repository at the given time. You do not have functionality as importing roster from LMS, create/see information about group repositories, creating template repositories, or managing your group repositories. This web application only facilitated for viewing information about already created groups and did not enable for the management of them in the web application.



SourceControl
http://github.com/gittrdone/sourcecontrol
This MQP
master

107	files
276	commits
6	committers

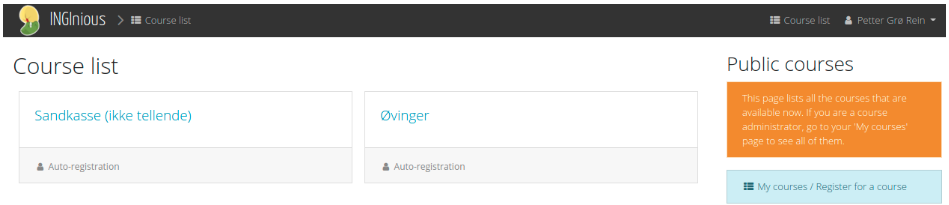
Justin Chines	89 commits	Anjali Venkatesh	76 commits
March	63 commits	Andrew Iovanna	52 commits

Justin Chines, Andrew Iovanna, Thanaporn Patikorn, Anjali Venkatesh
Worcester Polytechnic Institute

Figure 9 - SourceControl.me dashboard

3.2.4 INGInious – Web Application for Automated Evaluation

INGInious [27] is a web application that allows you to securely run automated exercises assessment on deliveries, also providing a pluggable interface with your existing LMS. INGInious does not have the functionality to be integrated into a Git hosting platform and is only for giving automatic feedback on student's deliveries. We believe it is a great platform to collect deliveries and give feedback right away to students. It can be a great idea to use the exercise assessment functions from this web application in other applications. INGInious are used in the course TDT4120 to allow the students to run their code to check if it is working, and they get ranked based on the runtime of their code [28].



This page lists all the courses that are available now. If you are a course administrator, go to your 'My courses' page to see all of them.

Figure 10 – INGInious dashboard

3.3 Git in the Classroom

There are several functions from the Git repository hosting services that can be used to help your course. Angulo and Aktunc [29] mention as an LMS to host course material, collection assignments, and projects.

Git repository hosting services cannot be used as full-fledged LMS but in conjunction with another LMS. They usually lack features as gradebook management, formal assignment submissions, and plagiarism checking. But what they provide is an easy method of storing your course material year to year, keep track of versions of your material, and the option to make the material public so everyone can read it, even people that are not students. If you chose to have the material on a Git repository hosting service, you could get direct feedback in terms of pull requests or issues directed to the material. If you store assignments that contain skeleton code, and you made an error, you can fix this by creating a branch/update on the initial code, and the students can merge the fix into their code that they already have worked on with functionality from Git.

Git repository hosting services also is a great way to collect assignments, especially programming assignments. They provide an easy way to see the contents of everybody's assignments, you can see the constant updates done to the work, and for programming assignments, LA's can give direct feedback on the assignment with a pull request or an issue.

Where using a Git repository hosting services excels is when students must collaborate. Using Git, you get version control and branching, and with using a Git repository hosting service, you can push the local changes to the cloud, and the students' peer can fetch the changes and merge them into their copy.

When setting up the course on a platform, choose one that supports creating hierarchies of organizations/teams/groups. This is because you could add the staff to the top organization, and they will get access to everything below the top organization. Also, if you as a course instructor create the repositories, you can set the read/write permission and make private repositories so groups cannot see and plagiarize other group's work. RepoBee [22] uses hierarchies to mirror your course on GitHub or GitLab, so if you want to use tools like RepoBee, your Git hosting platform much support some hierarchy.

Another advantage you get from using a Git repository hosting services is that most of the platforms support issue tracking, which is a method that lets you create a discussion or asking questions precisely at a point in the code where it can be a problem that needs to be fixed [30]. The students can use issues to track work that needs to be done or bugs in the code. Instructors can use issues to create feedback on an assignment or a piece of code.

A problem with Git is the initial learning curve for students, some GUI tools aim to make it easier to use, but students often find it easier to use the command line. This may be because the GUI shows you all of the functionality for Git, and it can be a bit overwhelming, then it is easier to remember a few easy commands. Later, when you have used Git for a while and understand how it works, you can switch over to the GUI for its easier usability, nicely said by Kelleher: "The Git system was forgiving to novices though the GUI tools would likely be of greater value when knowledge of the system matures for students" [30].

3.4 Displaying Git Data of Groups

When the students use Git in a course, you can harvest a lot of data because of the norm to constantly push work to the cloud, which is being recorded. Not only can you look at the finished artifact from a group, but you can get data during the course that can be used to give feedback to groups. Several data points can be used to do an assessment of work, number of commits, size of commit, code lines added/removed, timing of commits (date when work done), number of pull requests, issues raised, number of discussions on someone's code and how good they follow good Git methodic.

Several papers have been published that look into what data points you can use to do an assessment; Sprint and Conci [30] showed a weak correlation between the grade given to students and how they used Git.

You cannot give a correct grade only based on the data points you can harvest from Git, but you can use some of them as a guide.

Parizi, Spoletini, and Singh [32] created a tool for objectively measure a team member’s contribution using data from a Git repository that results in a report that instructors can use to evaluate a group. They hope that such a tool will encourage students to be more actively engaged by providing meaningful contributions and efforts and ultimately benefit universities to produce high-quality and competent software engineering graduates for the industry.

GitInspector [24] is a tool for harvesting data from a selected group. GitInspector shows data about the percent and number of commits done by contributor, additions/deletions in the code, percent of changes in the code done by the contributor, number of surviving lines created by the contributor, a timeline of additions/deletions, and which files the contributor contributed the most work into.

4 Defining Requirements and Understanding the Problem

An important part of dealing with a challenge is to understand the problem and define how it can be dealt with. Following the research method "Design and creation" this can be seen as the awareness and suggestion steps. To better understand the problem the team identified the possible stakeholders in the problem, and thoroughly interviewed them about their situation and context, and how they are dealing with the problem in the current situation. It was also attempted to identify any ideas or requests that could help the team better understand how the situation could be improved.

Following the initial interviews, the team started on their suggestion, a web application that aims to support the course staff in software development courses and others facing similar challenges. This was done by defining a set of requirements the application would need to fulfil, and by using tools defined in the user-centered design approach and the agile development process, including personas, scenarios, and user stories. Some aspects of software architecture were also considered at this stage, including the identification of functional and non-functional requirements related to the web application, together with technical decisions such as the selection of the technology stack and how the web application should interact with other relevant software such as Git repository hosting services and learning management systems.

4.1 Stakeholders

People interested in the artifact being developed are generally referred to as stakeholders. Based on their connection to the artifact, the stakeholders are often divided into groups that the designers and developers can have in mind during their work.

4.1.1 Stakeholders and their Concerns

Stakeholders generally have various concerns that designers and developers should have in mind when creating software. Many of them can be related to what is usually referred to as software quality attributes. These are defined by Bass et al. as "a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders." [33, p. 63]

We mapped who we believed to be our stakeholders during the start of the project and attempted to map them to their main quality attributes of concern.

As presented in Table 10, end-users are the users that may be using the application. This is typically course coordinators and learning assistants. Developers are the people implementing changes in the software, this group can include both contracted students and course coordinators. The system administrators are the people ensuring that the application is working safely and functioning as planned. In this project, some of these groups may overlap. For example, a course coordinator with knowledge about system development may fill the role as end-users.

Stakeholder group	**Quality attribute of concern**	**Explanation**
End-users	Usability	The application should be easy to learn and use for all users.
	Security	The information in the application should be safely stored and processed.
	Availability	The application should be available when the user needs it.
Developers	Modifiability	The implementation should be possible to change when needed.
System administrators	Interoperability	The system should exchange and store information in a meaningful way.

Table 10 - Software quality attribute concerns for stakeholders

4.2 Initial Expert Interviews

The project was based on a user-centered design approach. Early interviews with the most important stakeholders seemed like a good place to start the process. Two course coordinators were decided to be the most important stakeholders in the early phase, together with the supervisor of the project and two head learning assistants. The participants had a connection with the courses at NTNU using Git in the classroom.

4.2.1 Impressions from interviews

The interviews were lengthy, and not all that was said, was of interest to the project. Therefore, only the blocks of particular interest was included in this section.

4.2.1.1 Course Coordinator 1

The interview was conducted at the NTNU campus with a course coordinator in a software development course at NTNU.

On a question, if the participant has any ideas about what tasks are requiring the most time, the participant stated the following:

- We are two coordinators in this course, and it is the other one that has been creating the groups and such things. I feel that it takes some time, but he [the other coordinator] has automated some of it. What often takes much time after the course is created, is adding single members to a group or removing them, you can call it maintenance of the course.

  I have many requests where I must go to a web address to look at the Git repository of a student, instead of going through a list of all the students, searching for the name, finding it, and having a nice presentation of the Git repository. In our course, we have individual assignments first, then group work. In any case, I need to quickly log in and get an overview of the course before I start looking at the details.

When the possible solutions to these problems were discussed, he expressed the following:

- He [the other course coordinator] has written a script, I guess it is Python-based, but I am not a fan of Python stuff and prefer doing

> things with Java, so I have thought about creating a Maven plugin, reading some kind of configuration file. Anyway, it does not really matter what technology it is based on, if only configuration can be done based on information files about the course. So, if we could check out [using Git] and build it using some kind of a build job, ensuring things [groups, permissions, and code repositories] are being set up, I think it could be a good solution.
>
> There are two types of scenarios. There is one where we are rigging for the whole class when the course starts, and there is one where we are incrementally changing it. An intelligent tool will then use the list of students in the course, add and remove recent changes based on what is necessary. I think that would be a good solution, and then I think, why not use Maven for that? But that is because I already use Maven extensively, and the Java API looks very rich, built around the REST API.

Asked about what this technical solution could do, he expressed the following:

- It could be some kind of job that checks out their [the students] projects, where we generate a report for each repository. Again, this could be solved with Maven, using it to traverse the list of students and groups, check it out, and generate a report to get the overview we want. A web application could also do this kind of job.

  Something we are evaluating in my course is the correct use of Issues, Commits, and Branches, and to get an indication, one could look at closed issues, see if they have referred commits to an issue or something, and see what has been merged into master [the master branch in a Git repository].

Asked about if students often change groups, he answered:

- No, it’s more in the sense that if a group doesn’t work well together, we split that group, and this should ideally happen before any hand-ins are delivered.

The interview ends with the participant stating that he looks forward to seeing the results of the project.

4.2.1.2 Course Coordinator 2

The interview was conducted digitally through a video link with a course coordinator in one of the software development courses at NTNU.

Asked about if students often change groups, he answers:

- No group changes have been made, as we focus on group work and how people work together. We have random groups, and we think it works well. We have a couple of challenges with the fact that not all students in Blackboard end up participating in the course. We haven't got complete control over the issue, but we are letting the student groups manage themselves on GitLab so that they can remove non-participating students or keep them in the group if they want to.

Asked on what kind of functionality he thinks a third-party application to Blackboard and GitLab could contain, he answers:

- Well, the way we are using those systems, we are using them to separate things. Blackboard is a web page where the students access course information and is also used as a grading tool. I have seen that tools providing statistics such as burndown charts exist, which could be interesting to see for student groups in Blackboard so that we could see how the group functions in GitLab. I am just worried that this would require a lot of implementation work, so currently, the alternative is logging into GitLab ourselves.

Asked on whether he thinks some kind of activity-based dashboard could be useful, he answers:

- This [the lack of activity-based dashboard] is a general problem with Blackboard. What I can see when looking at a group in Blackboard is very poorly made. I can see the group members, but I cannot see what they have been working on. So, everything there could be improved, to say it that way. And I think that having some kind of dashboard with links for each group could have been an excellent general function in Blackboard so that we could add some GitLab-links and such on the group pages of Blackboard, which would help a lot.

Asked on whether he thinks it would be useful to have a report that detects and diagnose issues within the groups, he answers:

- I think it would be a plus only in complicated groups. In most cases, we have a deliverable, the students deliver that, and okay, everything is there, and they get their points. But, if I get a complaint that the group is not working, I think it would be good to click on the members in Blackboard and be able to see how many Issues each member has closed and how much he has committed to see some activity. I don't know, I mean, this is also a sensitive thing. What I normally do, is that I have a meeting with

> all the group members, and everybody has the chance to say their thing, and normally the problems get solved like that. It’s impressive to see how little the groups sometimes talk to each other because they assume that the other ones are mean. But when you put them into a group, the problem is solved. But I think that in extreme cases, or in cases where we wonder: OK, what have they really done, because it may not be easy to see out of the deliverable, then it may be helpful. But in general, I think all this information may cause a lot of noise. We have to be careful not to have too much information also.

The interview ends with the participant stating this is a very interesting project.

4.2.1.3 Head Learning Assistant 1

The interview was conducted in a written manner, and all quotations are translated from Norwegian to English. The participant has a background as a head learning assistant in two of the software development courses at NTNU.

What administrative tasks do you have the impression taking the most time in your courses?

- The administrative tasks in Blackboard take the most time. This includes creating assignments, placing members [students and learning assistants] into groups and giving them the correct permissions in the group. A lot of this work involves manual processes requiring many hours of labor. Often the management of groups is delegated to student learning assistants so that one person does not have to do this work for 500 students. Apart from that, there is a lot of administrative work regarding the evaluation of exercises, where we often have to “battle” with Blackboard. The creation of groups in GitLab does also require a considerable amount of time.

Were there any tasks in your courses that caused duplicate work in Blackboard and GitLab?

- The creation of groups is absolutely the largest [task] that have to be done twice, especially if the automatic group creation function of Blackboard cannot be used. GitLab groups must be created manually, requiring a couple of hours. We solve this by adding just one student from each group and ask them to invite the rest of the group themselves.

How much of this work is done by you that potentially could have been delegated if appropriate tools were available?

- Group creation requires many hours, but we manage to delegate it. Other tasks also require us to use tools other than Gitlab and Blackboard. For example, we use Excel to organize which groups to monitor, and to give student learning assistants feedback on how they guided their groups. This could have been integrated into Blackboard, but we find it easier to use Excel.

4.2.1.4 Head Learning Assistant 2

The interview was conducted at the NTNU campus, and all quotations are translated from Norwegian to English. The participant has a background as a head learning assistant in one of the software development courses at NTNU.

What administrative tasks do you have the main responsibility for that requires a lot of time and resources?

- The main administrative task that I do that takes much time in the course is to create the student groups. We ask the students to answer a form and use the answer to match students with a script created by us. The problem is that if students answer incorrectly on the form, then the script won't work. When the script has decided on the groups, we can create the groups on Blackboard and GitLab.

How do you create the groups on Blackboard and GitLab?

- Blackboard has a group creation tool that takes a CSV file from our script and uses the CSV file to create the groups on Blackboard. For GitLab, we give the CSV file to the course coordinator, which uses the file to create the groups on GitLab.

4.3 Aggregated Functionality Requests from Interviews

Based on the data gathered in the above-mentioned interviews and our understanding of the problem, some aggregated needs and feature requests for our application were aggregated, as presented in the following sections.

4.3.1 Dashboard

Not mentioned by the head learning assistants, but mentioned by both course coordinators and our supervisor, is that a tool for getting better insights about the work done by student groups in code repositories is wanted.

Both subjects in our interviews seemed to request a dashboard. The dashboard should, in their opinion, contain a list of the groups in the course, and ideally their progress or latest activities.

The dashboard should also include some sort of link portal. This is mainly based on insights from interview 1 and based on our understanding from interview 2, the feature would also be welcomed by subject 2.

4.3.2 Group Membership Handling

Both the course coordinators and one of the head learning assistants stated that the process of managing groups in group-based project work is one of the tasks requiring the most time and resources. Various levels of automation and delegation regarding this work were reported. However, we have a strong indication that multiple hours are spent managing groups regardless of who performs the task.

4.3.3 Feedback Through Creation of Issues

One of the course coordinators mentioned that the communication with students during practical activities in software development courses could be improved. This was as we understood it connected to the difficulty of giving coding-related feedback through platforms such as Blackboard and e-mail. A proposal was then to provide feedback to the students by creating Issues that would directly appear in the GitLab repositories. We agreed that a feature like this should be possible to implement, and would be a time-saving feature for the course staff.

A reason for wanting this feature was explained to be that it was more in line with how Issues and Pull Requests are used to give feedback on work out in the industry. Using those methods, it is possible to discuss and comment additions or changes directly in the code. As this is how it is done in the industry, the course staff wanted to explore using the same methods for giving feedback to student assignments and make assignments so that the students become familiar with techniques used in the industry.

A problem with this approach is that there might be a need to create the feedback both in an issue or pull request and on Blackboard if it is part of the grading process, creating duplicate work that we want to avoid. So, the proposal here was to perhaps create a system where you can give the feedback and cross-post it to Blackboard and GitLab, but this would require a substantial amount of implementation work. The feature was therefore not prioritized for the minimum viable product.

4.3.4 Automated Tests

Both course coordinators mentioned running automated tests as a feature they would like to have in software working with Blackboard and GitLab. However, this was decided to be a relatively resource-consuming functionality to implement, so the team decided it should not be done in the timeframe of this project.

4.3.5 One-click Deployment of Student Work

Many of the practical activities in software development courses revolve around projects containing some kind of runnable code. During the interviews, it was mentioned that running those projects by clicking a link or button in the group dashboard would be useful. However, like automated tests, this would require integration with a third-party software such as GitPod or complicated infrastructure requiring both time and resources that would require a large number of implementation resources and time that could have been spent on more important features.

4.3.6 Assignment Related Functionality

Assignments are often part of the practical activities in software development courses, and course staff may be interested in how the student groups worked during the assignment. Participants mentioned that they now tag the last Commit pushed to GitLab to create a snapshot that the learning assistants will use when giving feedback on a particular assignment. Together with allowing the course management to configure automatic tagging of assignments on the delivery date, another interview participant mentioned that he wanted a report of the work done during the assignment added to the tag.

4.3.7 Restricting Functionality for Learning Assistants

One of the interview participants was concerned about giving too much power to learning assistants. The participant wanted to do the work himself to ensure it was done correctly. A proposal here was to create a feature for the learning assistants to suggest changes and edits. The course coordinator would be presented with a list of suggestions, having the option to approve or disapprove them. This would result in the course coordinator still controlling everything that would happen, enabling learning assistants to do work when suitable.

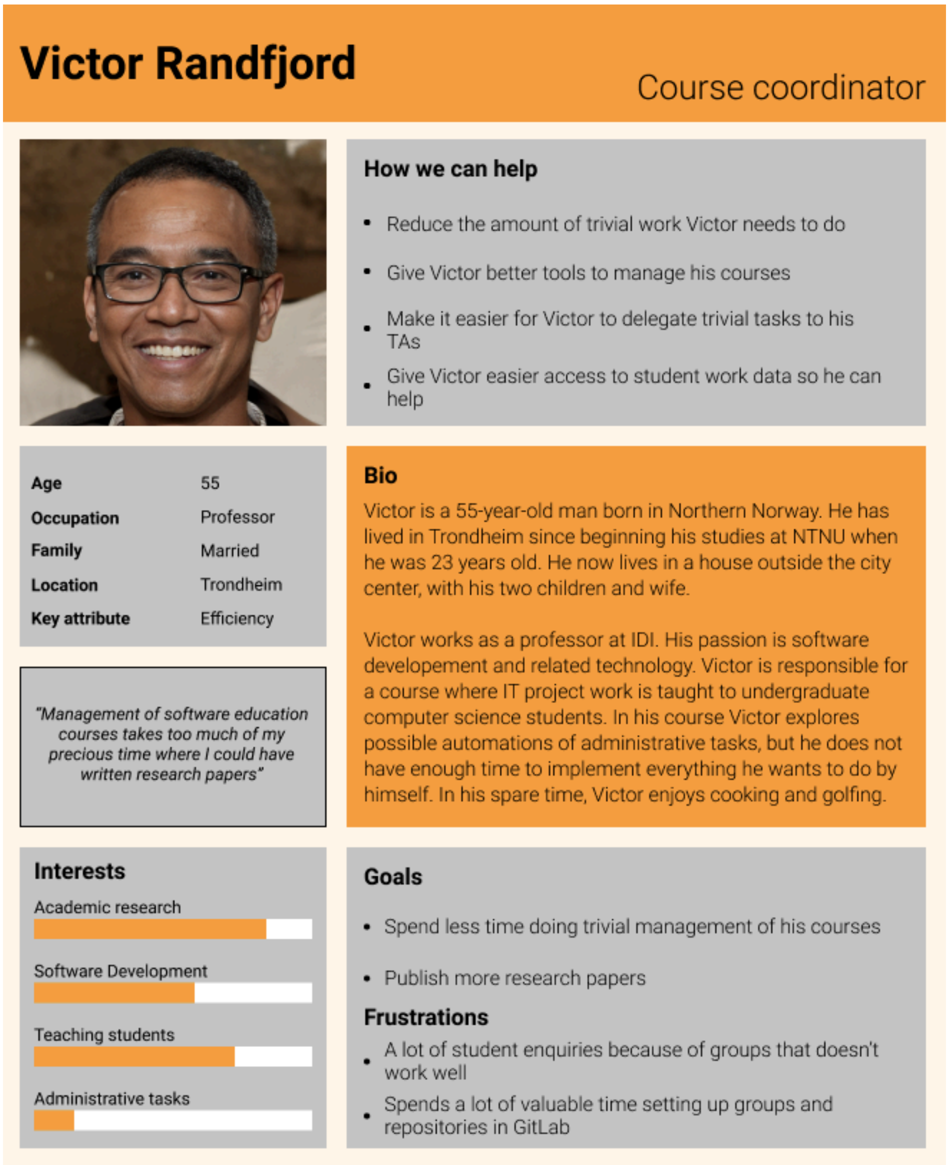
4.4 Personas

As described in section 2.5.2, three personas were created as a part of the user-centered design process. These are inspired by several sources, including the expert interviews conducted at the beginning of the project and the learning hierarchy at NTNU. Personas are detailed descriptions of typical users of the product under development, on which the designers can focus and for which they can design products. [10, p. 403]

The personas in this project are inspired by data gathered from the initial interviews, statements from stakeholders, together with our own impressions of the situation. Personas sometimes include some degree of psychological profiling often related to metrics from the psychological models commonly referred to as "The Big Five" [34] or "The Myers-Briggs Type Indicator" [35]. However, these are not included in our personas. This is based on the assumption that gathering accurate data on this topic requires a fair amount of time, knowledge, and resources, combined with our belief that those attributes of personas are more important when attempting to more actively engage with the potential users, or sell them products or services.

4.4.1 Victor - Course Coordinator

Victor is one of the personas. He is a professor at NTNU and the course coordinator in one of the courses where GitLab and Blackboard are used for software development education at NTNU.



Victor Randfjord

Course coordinator

How we can help

- Reduce the amount of trivial work Victor needs to do
- Give Victor better tools to manage his courses
- Make it easier for Victor to delegate trivial tasks to his TAs
- Give Victor easier access to student work data so he can help

Age	55
Occupation	Professor
Family	Married
Location	Trondheim
Key attribute	Efficiency

"Management of software education courses takes too much of my precious time where I could have written research papers"

Bio

Victor is a 55-year-old man born in Northern Norway. He has lived in Trondheim since beginning his studies at NTNU when he was 23 years old. He now lives in a house outside the city center, with his two children and wife.

Victor works as a professor at IDI. His passion is software developement and related technology. Victor is responsible for a course where IT project work is taught to undergraduate computer science students. In his course Victor explores possible automations of administrative tasks, but he does not have enough time to implement everything he wants to do by himself. In his spare time, Victor enjoys cooking and golfing.

Interests

Goals

- Spend less time doing trivial management of his courses
- Publish more research papers

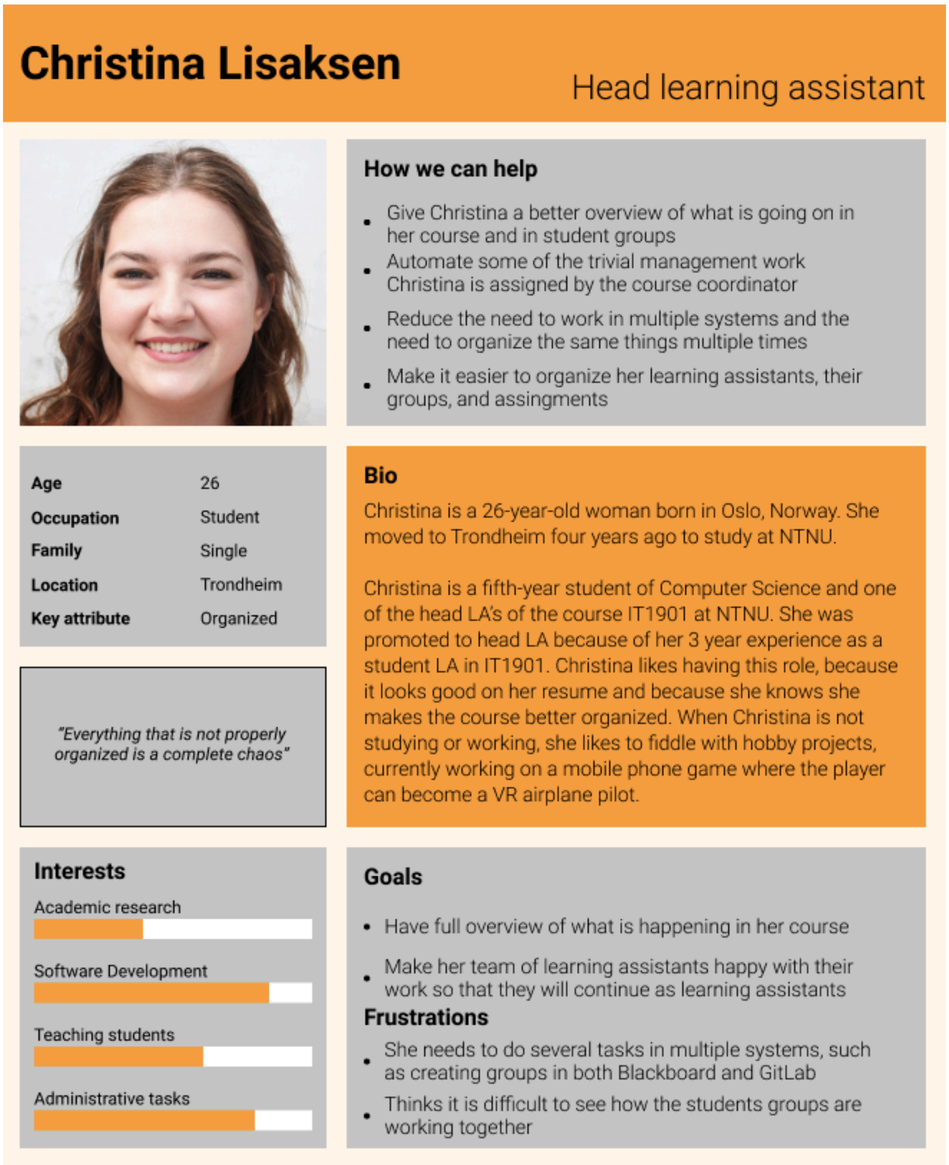
Frustrations

- A lot of student enquiries because of groups that doesn't work well
- Spends a lot of valuable time setting up groups and repositories in GitLab

Figure 11 - Persona card course coordinator

4.4.2 Christina – Head Learning Assistant

Christina is one of the head learning assistants in the course where Victor is the course coordinator. Christina is a fifth-year student of Computer Science and is part-time employed as a head learning assistant.



Christina Lisaksen

Head learning assistant

How we can help

- Give Christina a better overview of what is going on in her course and in student groups
- Automate some of the trivial management work Christina is assigned by the course coordinator
- Reduce the need to work in multiple systems and the need to organize the same things multiple times
- Make it easier to organize her learning assistants, their groups, and assingments

Age	26
Occupation	Student
Family	Single
Location	Trondheim
Key attribute	Organized

"Everything that is not properly organized is a complete chaos"

Bio

Christina is a 26-year-old woman born in Oslo, Norway. She moved to Trondheim four years ago to study at NTNU.

Christina is a fifth-year student of Computer Science and one of the head LA's of the course IT1901 at NTNU. She was promoted to head LA because of her 3 year experience as a student LA in IT1901. Christina likes having this role, because it looks good on her resume and because she knows she makes the course better organized. When Christina is not studying or working, she likes to fiddle with hobby projects, currently working on a mobile phone game where the player can become a VR airplane pilot.

Interests

Goals

- Have full overview of what is happening in her course
- Make her team of learning assistants happy with their work so that they will continue as learning assistants

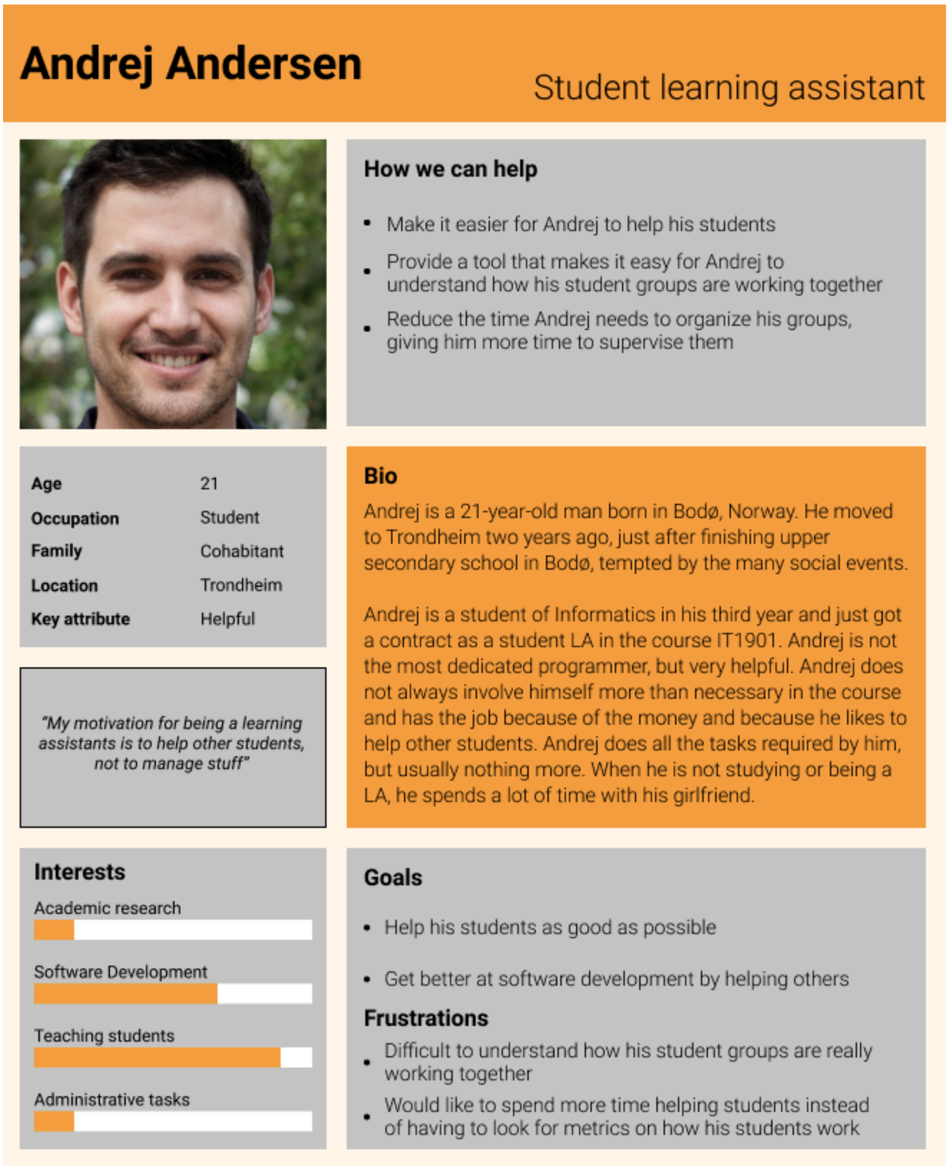
Frustrations

- She needs to do several tasks in multiple systems, such as creating groups in both Blackboard and GitLab
- Thinks it is difficult to see how the students groups are working together

Figure 12 - Persona card head learning assistant

4.4.3 Andrej – Student Learning Assistant

Andrej is one of the many student learning assistants in the same course as Christina and Victor is a part of. Andrej is a third year Informatics student at NTNU, and is part-time employed as a student learning assistant working fewer hours than Christina.



Andrej Andersen

Student learning assistant

How we can help

- Make it easier for Andrej to help his students
- Provide a tool that makes it easy for Andrej to understand how his student groups are working together
- Reduce the time Andrej needs to organize his groups, giving him more time to supervise them

Age	21
Occupation	Student
Family	Cohabitant
Location	Trondheim
Key attribute	Helpful

"My motivation for being a learning assistants is to help other students, not to manage stuff"

Bio

Andrej is a 21-year-old man born in Bodø, Norway. He moved to Trondheim two years ago, just after finishing upper secondary school in Bodø, tempted by the many social events.

Andrej is a student of Informatics in his third year and just got a contract as a student LA in the course IT1901. Andrej is not the most dedicated programmer, but very helpful. Andrej does not always involve himself more than necessary in the course and has the job because of the money and because he likes to help other students. Andrej does all the tasks required by him, but usually nothing more. When he is not studying or being a LA, he spends a lot of time with his girlfriend.

Interests

Goals

- Help his students as good as possible
- Get better at software development by helping others

Frustrations

- Difficult to understand how his student groups are really working together
- Would like to spend more time helping students instead of having to look for metrics on how his students work

Figure 13 - Persona card student learning assistant

4.5 Scenarios

Scenarios were created based on the defined personas in section 4.4 to facilitate the creation of user stories at a later stage of the user-centered design approach. Therefore, the scenarios are mapped to the personas included in section 4.4.

4.5.1 Scenario for Course Coordinators

It is a sunny day in Trondheim in August, and Victor has just returned to his office after a well-deserved vacation. His course IT1901 is scheduled to start with the return of the second-year Informatics and Computer Science students. The students have registered for the course in the web application StudentWeb and have therefore been added to the course room in Blackboard.

Victor wants to have the course setup ready before the lectures commence so that the students can start working on their practical activities as soon as Victor has introduced them in the second lecture of the year. For this to be possible, Victor needs to divide the students into groups in both Blackboard and GitLab, which previously was a time-consuming task that needed multiple days to fully complete. So, instead, Victor logs into the new web application designed and implemented by two of his master students. Here, he authenticates using the same authentication method as he uses for all other services at NTNU, selects his course for the correct term, enters details about the GitLab instance, and enters the application's dashboard.

The dashboard then notifies Victor that there are new students in his course, based on their existence in the Blackboard course room, and that the new students should be divided into groups. The application then asks Victor how many students he would like to have in each group and then creates a suggestion of groups, that Victor can either accept or edit to his preferences. Victor thinks the suggestion seems fine and accepts it, resulting in the groups being created in Blackboard with the students correctly assigned. The groups are then visible in the application, and Victor is asked if he wants to replicate the groups in GitLab with code repositories. Victor wants the application to fix this for him, as doing it manually would require a substantial amount of work, and he clicks the create group button.

Victor then confirms that the groups and repositories look fine, before he creates the first assignment of the course. The he can see that everything has completed successfully, making him ready to introduce the practical activities in the second lecture of the course as planned.

4.5.2 Scenario for Head Learning Assistants

It is one of the many rainy evenings during the autumn in Trondheim, and Christina has decided to catch up with some of her tasks in IT1901 after studying for some hours.

The course is well underway, with the students having just delivered their work for assignment 1. This creates a couple of tasks for Christina. One of them is to reassign group members based on feedback from the student learning assistants, a task where multiple tasks had to be carried out both in Blackboard and GitLab. Now, Christina enters the group functionality of the new application for managing practical activities in software development courses. Here, she rearranges some of the students in the groups that her student learning assistants have reported are not working together very well and asks the application to make sure the changes are synchronized across Blackboard and GitLab.

After Christina has confirmed that the changes have been carried out by looking at the group list again, she continues with her other tasks, such as evaluating groups that her other learning assistants have challenges evaluating themselves, before taking the rest of the evening off.

4.5.3 Scenario for Student Learning Assistants

It is an early Monday morning, and Andrej is preparing to meet his students after spending a weekend in his hometown Bodø. Andrej has been assigned responsibility for six of the groups in IT1901. Therefore, he is arranging meetings with the groups every other week, where Andrej and the student groups discuss how the students in the groups work together and if they are facing any challenges or problems, either related to their work or the group itself.

Even though Andrej prefers to work as little as possible, he thinks meeting the groups without being prepared is very stressful. He is afraid that being unprepared may leave him unable to help the groups with their challenges in a good way. Andrej was last year a student LA in another course at

NTNU where they were not using the new tool for course management in software development courses. From that course, he remembers that checking how the groups were working required him to locate and analyze data in the various GitLab repositories by himself and comparing the groups with the impression he had of other groups at the same time.

In IT1901, however, Andrej has access to the new tool, and by entering the application, he is presented with a nice dashboard containing the student groups he is responsible for. To get the information he needs, he selects one group and is then presented with details of the group's work, together with some charts that help Andrej quickly understand how the group is working. Andrej can also compare the group with the stats of other groups by simply clicking a button. Now, Andrej has gathered all the data he needs to get a good feeling on how the group works together, before meeting the groups to compare his impressions with the group's own impressions on how things are working out.

4.6 User Stories

User stories were created based on the project proposal from our supervisor, personas, scenarios, and insights from the interviews earlier in the process. The art of defining user stories has developed over the years but was in 2009 described by Cohn as something that "… describes functionality that will be valuable to either a user or purchaser of a system or software." [36, p. 4]

In this project, the user stories were regarded as the basis of the software development carried out while also describing the users' needs. After being defined, the user stories were inserted into an issue board in the GitLab project used by the developers, allowing the software development progress to be connected to the user stories.

A difficult part of most agile software development projects is selecting what should be implemented first. This is often handled by sorting the user stories of the project. A vast number of factors can be considered when prioritizing user stories, such as the importance of the user stories for the end-users, user stories depending on the existence of each other, and the required resources to complete functionality for a user story.

It was considered creating user stories strictly based on the format "As a [insert course role], I would like to [task].", but this was considered to make room for confusion as administrative tasks are being distributed in various ways in the different courses, meaning that people with different roles may perform the same tasks.

The user stories implemented are included in the sections describing iteration 1 to iteration 3.

4.7 Wireframes and Prototypes

Based on the user stories selected for implementation, we wanted to visualize how the solution would look like in a prototype. Often, this process tends to start with the creation of low-fidelity wireframes. The wireframes should contain the most essential element of a design and the content.

However, based on the circumstances with covid-19 resulting in a lot of remote work, it was decided that paper prototypes could be challenging to work with. Therefore, we decided to attempt wireframing using Figma, a web-based prototyping tool.

Wireframing in Figma comes with a few advantages and disadvantages. The most significant advantage in this project was how simple it was to use Figma to collaborate over design choices in real-time while working separately from home.

The most prominent disadvantages connected to wireframing in Figma include increased time from idea to wireframe, and increased complexity while creating wireframes. However, it was believed that because working from home was a necessity in periods, the advantages would be more significant than the disadvantages.

4.8 Architecture Requirements

Based on the initial interviews with the stakeholders in the project and our mapping of stakeholders and their perceived concerns about software quality attributes, tables containing architectural requirements [33, pp. 63-78] were worked out.

The main quality attribute selected for the project was Usability, with the secondary quality attribute as Maintainability.

Usability was selected because there would be a strong focus on how the user sees and uses the application. The research strategy chosen was design and creation, as described in section 2.1, focusing on user-centered design, as described in section 2.5.

Maintainability was selected as a quality attribute because it expressed concern that students' projects often did not get maintained and updated after the initial project was finished. Also, under maintainability, you have modifiability as an attribute, and a desired feature was that the application could be fitted to the individual course, but also other entities that might have a use for a tool for information and feedback from Git hosting platform would use it but not have the same environment as NTNU.

4.8.1 Functional Requirements

The functional requirements define the system´s capabilities, e.g., what the system should do and how it should behave. Thus, the requirements directly impacted the application architecture, and the initial functional requirements are included in Table 11.

The requirements were translated to User Stories, which were used in user-testing. After the iteration they were implemented to check if we had satisfied the requirements.

ID	**Functional Requirement**
F1	View data from GitLab/BlackBoard.
F2	Create data in GitLab/Blackboard.
F3	View complex data as a visualization.
F4	Give access to everybody involved in the course.
F5	Run background task: aggregate data for later use, scheduled tasks.
F6	Run tests against student assignments. (Check for files, the content of files, run tests from the code)
F7	Run assignments in a secure environment.
F8	Authentication with Role-based access.

Table 11 – Functional Requirements for architecture

4.8.2 Non-Functional Requirements

The non-functional requirements will impact the application as a whole and can be used to judge the application and if the decided architecture is complete. The initial non-functional requirements are included in Table 12.

To check if we had satisfied the non-functional requirements of the application, we held expert interviews at the end of the project to get insights from experts in the field of software development.

ID	Non-Functional Requirement
NF1	Possible to access from everywhere. (On the bus, at home, at school)
NF2	Easy to host in a production environment.
NF3	User-friendly, easy to use, require little training to use.
NF4	Accessible for everybody, color blindness, etc.
NF5	Easy to set up and run for developers.
NF6	Can be used on the phone and/or the computer.
NF7	Maintainable for future development.
NF8	Support other LMS than Blackboard
NF9	Support other Git hosting platforms than GitLab.

Table 12 - Non-Functional Requirements for architecture

4.9 Deciding on Architecture

Before deciding on our technical architecture, it was essential to determine what kind of artifact would meet the largest number of architectural requirements and be possible to create with the time and skills available in the project. Listed in Table 13 are the considered client solution for the application.

4.9.1 Including a Backend in the Architecture

Because the functionality you get with having a backend is desired to meet functional requirement F4-F8 and the non-functional requirement NF1. It was decided to have a backend in the architecture.

There are several advantages to having a backend. The most important for us, and why we choose to have a backend: it can hold secrets, aggregate data before sending it to the client, schedule tasks, single entrypoint for a client to get data, and central storage for users.

Combined, these features mean that we can authenticate towards our backend from a client and not the different systems we want to integrate. Then the backend can connect to the other systems, collect the data needed, transform data into a structure we want, and send it as a single payload to the frontend.

Without a backend, you cannot sync data, so you cannot switch devices on the fly, and everybody cannot see your local changes. If you want to run a background task, you must have the client constantly running, and it will be hard to set up a secure environment for student assignments.

You could also have greater performance if you host your backend close to the Git repository hosting service instance because of the vast amount of requests that are made. Every one of the requests could be slow down or failed and will delay everything, it is best if the backend and instance are on the same internal network for the best performance. This is also why we want a backend. You can get slower performance when doing all the calls from the client, so you do all of them on the backend, aggregate the data into a single payload and send it to the frontend.

4.9.2 Selection of Client Solution and Possible Clients

Nr	**Solution**
1	Web application
2	Mobile application
3	Desktop application
4	Extension for an existing service.

Table 13 - Possible client solutions for our artifact

Based on the architectural requirements identified and the possible solutions, we saw the following connections:

- A web application will meet every functional requirement if set up correctly. And will meet all of the non-functional requirements but will probably require some training to use and require some more work to host and maintain. The disadvantages of a web application are that you are constrained to the browser, you will not have access to all of the device's features, you are limited in performance, and differences in web browsers can cause challenges. But the advantages of running in a browser are that you are disconnected from the underlying operating system, often easier development, and time to market is faster. A web application can be made responsive, and the application can be used on mobile devices and stationary devices with different screen sizes.

- A mobile application will meet every functional requirement if set up correctly. But you will have a more challenging time to meet every non-functional requirement. Using a mobile application on your computer is possible, but that is not something you want to do. Also, it will have the same concerns as a web application with the need for training, hosting, and maintenance. Maintenance is a big concern with a mobile application, and you may have to update the application each time the mobile operating system updates. In the worst-case scenario, everything breaks. There are also different mobile operating systems with their own ecosystems, if you want to support most of the users you will have to develop for more than one system.

- A desktop application will have a hard time meeting requirement NF6, with only being made for desktop. You will also be constrained to the operating system you choose to develop for. And now, with the emergence of ARM processors in desktop, you have another processor architecture your application must support.

- An extension will meet all the functional requirements but will not meet NF8 or NF9. So, you will be stuck with the service you are extending. A goal of the project is that it can be used elsewhere than NTNU, and it is not desirable to limit which systems it can be integrated with.

The best solution looked like it was to make an extension or addition to an existing solution. For use at NTNU, it could be possible to extend Blackboard to integrate with GitLab. E.g., in Blackboard, when you create the groups for the course, you could also have the option to create the groups on Gitlab. This would have made it possible to see the group's Git data in Blackboard. With this solution, you do not have to add a new service and run it on a server, increasing the solution's maintainability. The user does not have to get familiar with a new system and thus improve the solution’s usability. It may be easier to get internal data from the service rather than open data extraction methods. The downside with an extension or addition is that you would have been stuck with the initial choices for LMS and Git repository hosting service. If you choose to change one of them later, you would maybe have to change significant parts of the extension. It also requires some knowledge of the system to extend and may not be the easiest to develop for, and it can be harder to make big changes. Because we wanted the solution to be independent of the LMS and Git repository hosting services the application is to be integrated towards, we choose not to explore and make an extension.

The second-best solution looked like it was to make a web application, as it will fit most of the requirements. A web application can be made agnostic of what LMS and Git hosting platform you develop for. But you would have to maintain a new service on a server, and the users must become familiar with a new service. Another advantage in our experience is that it can be easier for new developers to understand and maintain if you choose the right technologies. And you would have greater freedom in making changes you may want to do in the future.

We went with developing a web application because of the points mentioned above, and also because we both had experience with developing web applications and knew we could make a prototype faster with that.

4.10Architecture and Integrations

GitLab was used as the Git repository hosting service because IDI has its own instance running. Blackboard was used as the LMS we integrated toward. The GitLab instance is used for the courses IT1901, TDT4140, and many more, and Blackboard is used for every course at NTNU, so we had some real data to work with. We weren't allowed at the start to use the production instance of Blackboard and only got access to the test instance during development. It is possible in the future to switch to the production instance. If it is desirable at a later point to use the production instance, then the application needs to get approved by NTNU IT, change the URL and key for the LMS in the settings, and the application should be ready to use.

4.10.1 Architecture

The decided architecture was a web application made by Next.js. Next.js provides frontend and backend functionality in one package. Our application then talks with the services wanted to integrate towards, in our setting Dataporten, Blackboard, and GitLab.

An overview of the architecture can be viewed in Figure 14. The software used to build the architecture is listed in Table 14.

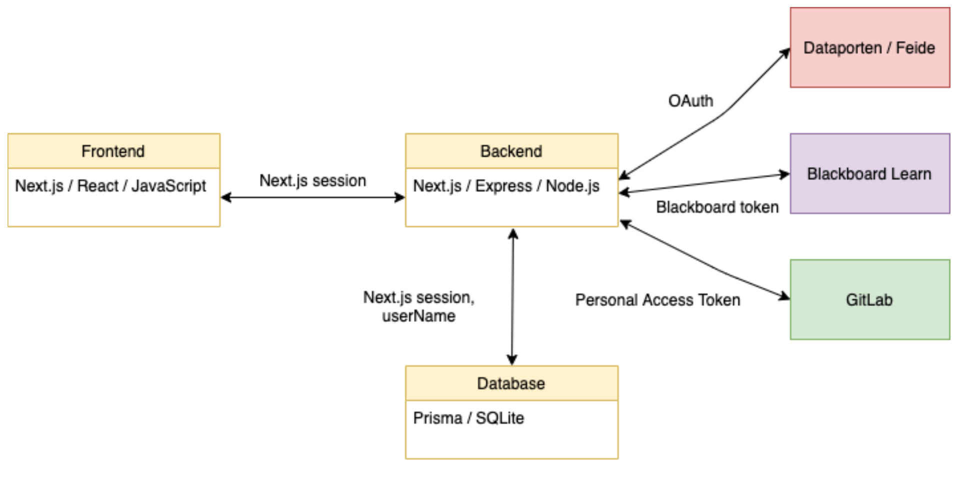


Figure 14 - Architecture diagram

Role	**Software**
Frontend	Next.js - JavaScript
Backend	Next.js - JavaScript - Node
ORM / Database	Prisma - JavaScript - Node - SQLite

Table 14 - Software used in the technical stack

4.10.2 Selection of Frameworks and Libraries

In this section, we are discussing our selection of framework. Of course, selecting the best framework that fits the requirements is essential, but you must also factor in the project's time and the developers' experiences with different frameworks.

4.10.2.1 Frontend and Backend Technology Stack

When selecting a technology stack for a web application, you may have unlimited choices, where each of the alternatives have some advantages and disadvantages. The easiest option is to choose something you have experience with, but that may not be the best choice overall for what you are doing. But what you can do then, is look at every framework and library you have experience with and see which of them will fit the requirements best. The frameworks and libraries for web applications that we among us have experience with are listed in Table 15.

Framework/library	Based on
React	JavaScript / TypeScript
Django	Python
.NET MVC	C#
Vue	JavaScript / TypeScript
Java Spring	Java
Express	JavaScript / TypeScript

Table 15 - Frameworks and libraries the team have experience with

The frameworks and libraries we knew existed and found in our research about possible frameworks and libraries to use in this project are listed in Table 16.

Framework/library	Based on
Svelte	JavaScript / TypeScript
Angular	TypeScript
Flask	Python
Meteor	JavaScript
Ruby on Rails	Ruby
Phoenix Framework	Elixir

Table 16 - Frameworks and libraries the team knew about

All the above-listed frameworks/libraries would fit the requirements for the product to some degree, and we had to find out which of them would fit them the best. We chose to focus on which of them would make it easier for people to develop and use, thus making the product easier to maintain and modify according to our quality attributes. The three most prominent and most popular frontend framework/libraries for web development are React, Vue, and Angular, as seen from the number of stars on GitHub as a metric. All of them support the use of "Components" that are reusable pieces of code that you can easily modify or extend and will improve our modifiability. We gave the edge to React as it is taught at NTNU in a course, and at NTNU, you will have a higher chance of finding somebody with knowledge about React and increasing maintainability.

4.10.2.2 Next.js & Express

We initially went with React as the frontend and an Express backend. This way, the same programming language was used, and with that, it becomes a bit easier to maintain. But when we were investigating how to do Authentication, we got a bit stuck and asked for help, and we were pointed towards a package for Next.js that would fit our needs perfectly. The package provided the Authentication we wanted and supported the use of other providers than Dataporten. So, if somebody wants to use something different than Dataporten, they will have the opportunity to do that. The switch to using Next.js instead was relatively easy as most of it is pretty similar to what we already had, we just had to make some configuration changes, move some files, and combine the frontend and backend into one. Next.js is just a React development framework that does server-side rendering of React pages, so it is still React but with some added functionality.

Next.js includes an Express backend and will export its own endpoints together with your pages if wanted. You can choose not to use it, but it is easier to have everything collected in one place. You will also get the advantage of reusing code in the backend and the frontend, so all the previous code you had to have duplicates of you can now store in a single place.

4.10.2.3 Prisma

Prisma was chosen because you don't have to write SQL statements in your code with an ORM, and you can use the already chosen programming language, in our case, JavaScript. Because Prisma handles the conversion between JavaScript and SQL, you can easily switch the database system behind it. This means that you can have a local SQLite database that requires no work to set up when you are developing, and you can use something like MySQL in production.

We chose Prisma over another JavaScript ORM because Prisma has TypeScript support, which is helpful if somebody wants to use TypeScript later. The process of defining schemes to map tables and columns was easy to understand and great to work with.

Other solutions are TypeORM, Mongoose, and Sequelize. The downside with them is that you have to define models that are mapped to tables and columns and can cause object-relation impedance mismatch. With Mongoose, you are also stuck with MongoDB as the database and cannot choose which database you want.

4.10.2.4 HighCharts

When looking for visualization libraries, we wanted something easy to set up, had excellent documentation, had good accessibility, and would give us the visualizations we thought we needed. HighCharts meet all these needs, they have pre-made charts that you can use, every chart has a demo on their site with a CodePen link, they have an accessibility module that you can add, and when we were looking through the premade chart, we found the chart we wanted plus some we had not thought of.

HighCharts is also developed by a Norwegian company and used by a lot of Norwegians.

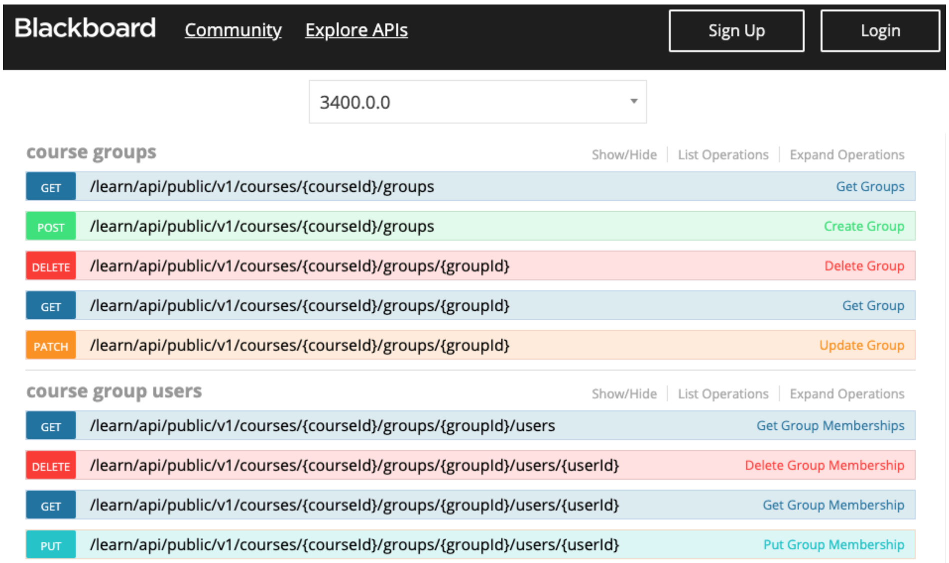
Other solutions are D3, Visx, and Recharts. The disadvantage with using D3 is that you have to create your own charts or copy somebody else's. It does not come with premade charts for you, but you will have complete control over the charts when you create them yourself. Visx and Recharts build upon D3 and provide you with chart components that can be combined into charts. Visx and Recharts offer you some pre-made components but still control over the charts. If more control is wanted than what you get with HighCharts, the chart logic in the application must exist in its own components so that the work needed to change it out is lessened.

4.10.3 Blackboard API Documentation

Blackboard has developed its own "API Dev Portal" [37]. The portal makes it possible to browse the various API endpoints available from Blackboard, see what parameters they require, what data they provide, and the data structure. You can also register as a developer in the portal, create applications, and request access to Blackboard instances.

The various API endpoints are versioned to have a mix of v1, v2, v3 endpoints. Several times during development, we used the wrong version and then looked at the site for the correct one. We think a better solution for Blackboard is to upgrade every endpoint to the latest version even if the endpoint does not change.

We found which endpoints we had to use to get the data needed for our artifact from the developer portal. We created a module in our project that is responsible for fetching the data we needed to get. The primary data is which courses a user is a member of, the students in the courses, the groups in the courses, and the members of the groups in the courses.



course groups

GET	/learn/api/public/v1/courses/{courseId}/groups	Get Groups
POST	/learn/api/public/v1/courses/{courseId}/groups	Create Group
DELETE	/learn/api/public/v1/courses/{courseId}/groups/{groupId}	Delete Group
GET	/learn/api/public/v1/courses/{courseId}/groups/{groupId}	Get Group
PATCH	/learn/api/public/v1/courses/{courseId}/groups/{groupId}	Update Group

course group users

GET	/learn/api/public/v1/courses/{courseId}/groups/{groupId}/users	Get Group Memberships
DELETE	/learn/api/public/v1/courses/{courseId}/groups/{groupId}/users/{userId}	Delete Group Membership
GET	/learn/api/public/v1/courses/{courseId}/groups/{groupId}/users/{userId}	Get Group Membership
PUT	/learn/api/public/v1/courses/{courseId}/groups/{groupId}/users/{userId}	Put Group Membership

Figure 15 - List of Blackboard API endpoints

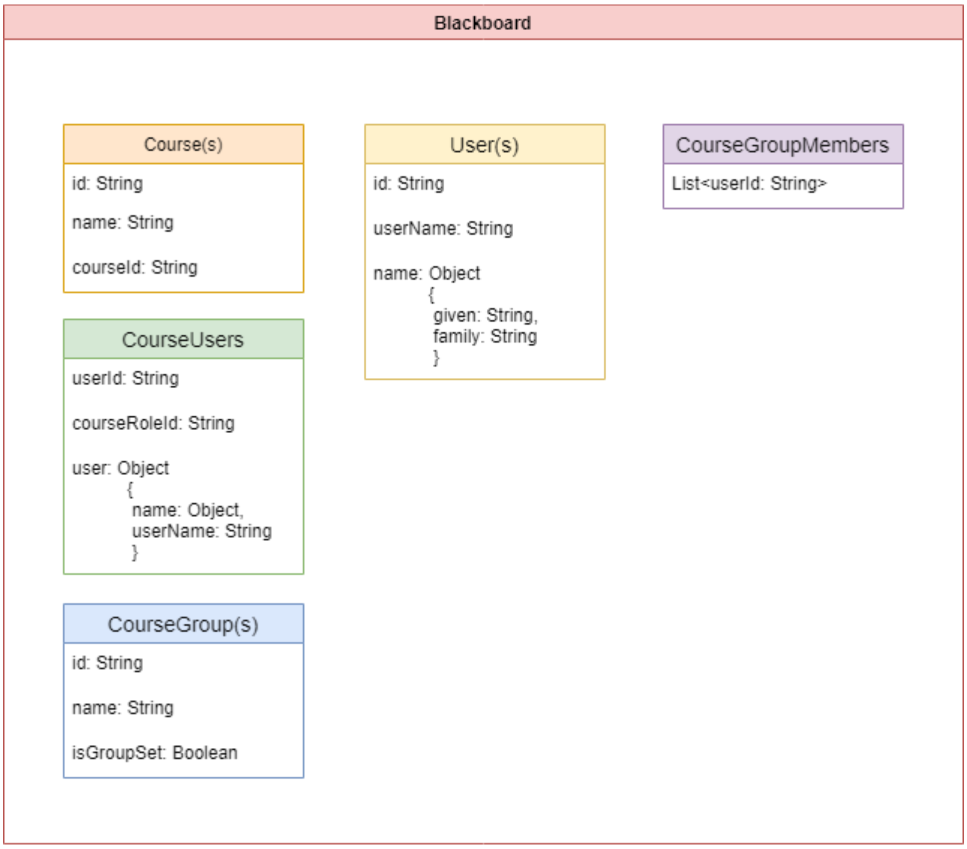


Figure 16 - Data used from Blackboard

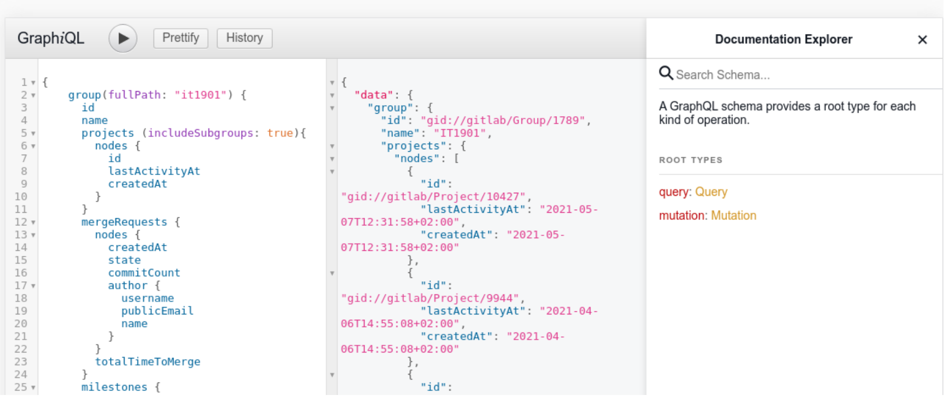
4.10.4 GitLab API Documentation

Compared to the Blackboard documentation discussed in section 4.10.3, GitLab is a bit different with their documentation. They have created a website that lists the resources with a webpage per resource [38]. They also have different versions of their API, but a big difference is that v4 is REST and v5 is GraphQL. They currently maintain both, but when they have fully implemented GraphQL and moved everything over from the REST endpoints, they will only maintain v5. When we developed the artifact, we used a mix of both as you want to use GraphQL because of the advantages you get from fetching a tree of data, but as v5 was not fully implemented, we had to get some data from v4.

When we were looking at documentation for v4, we used their webpages, but for v5, we found it easier to use the built-in GraphiQL explorer [39].

As with the Blackboard documentation, the GitLab documentation gives you: a list of every endpoint/resource, what parameters they require, what data they provide, and the data structure.

The data we wanted to use from GitLab was the groups in a course, members of the groups, milestones, issues, projects, and merge requests of groups, and branches, files, commits, and wikis in a project. In the figures below, you can see that some of the data is fetched over the v4 REST API and some over the v5 GraphQL API, then we later merge them. The REST models have their own endpoints, but the GraphQL data is fetched as a tree from the GraphQL endpoint.



```
{
    group(fullPath: "it1901") {
      id
      name
      projects (includeSubgroups: true){
        nodes {
          id
          lastActivityAt
          createdAt
        }
      }
      mergeRequests {
        nodes {
          createdAt
          state
          commitCount
          author {
            username
            publicEmail
            name
          }
        }
        totalTimeToMerge
      }
      milestones {
```

```
{
  "data": {
    "group": {
      "id": "gid://gitlab/Group/1789",
      "name": "IT1901",
      "projects": {
        "nodes": [
          {
            "id":
"gid://gitlab/Project/10427",
            "lastActivityAt": "2021-05-
07T12:31:58+02:00",
            "createdAt": "2021-05-
07T12:31:58+02:00"
          },
          {
            "id":
"gid://gitlab/Project/9944",
            "lastActivityAt": "2021-04-
06T14:55:08+02:00",
            "createdAt": "2021-04-
06T14:55:08+02:00"
          },
          {
            "id":
```

A GraphQL schema provides a root type for each kind of operation.

ROOT TYPES

query: Query

mutation: Mutation

Figure 17 - GraphiQL client made available by GitLab CE

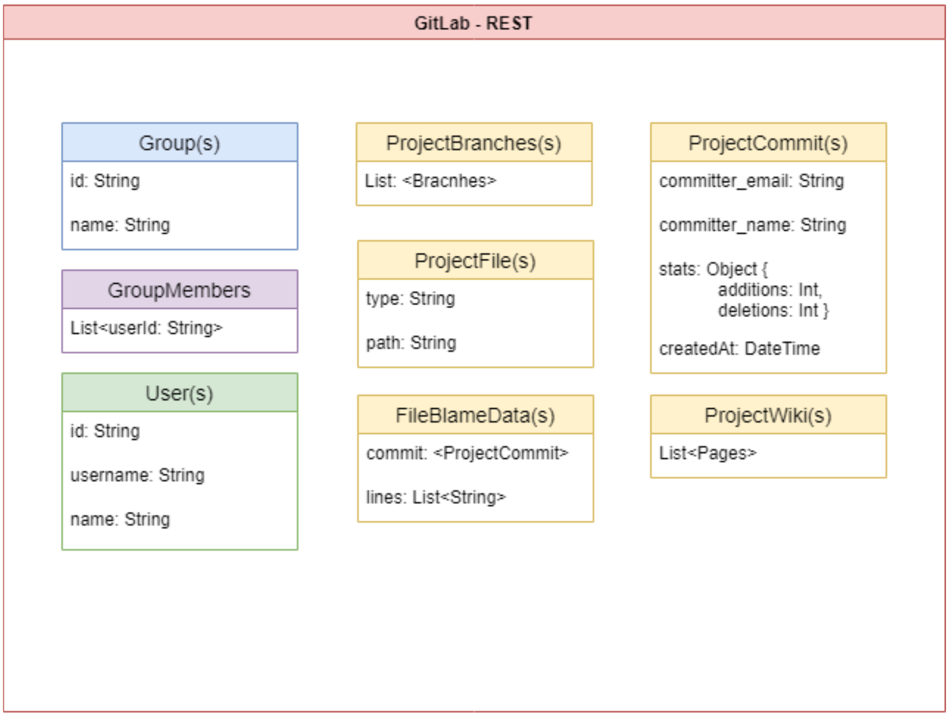


Figure 18 - Data used from GitLab REST

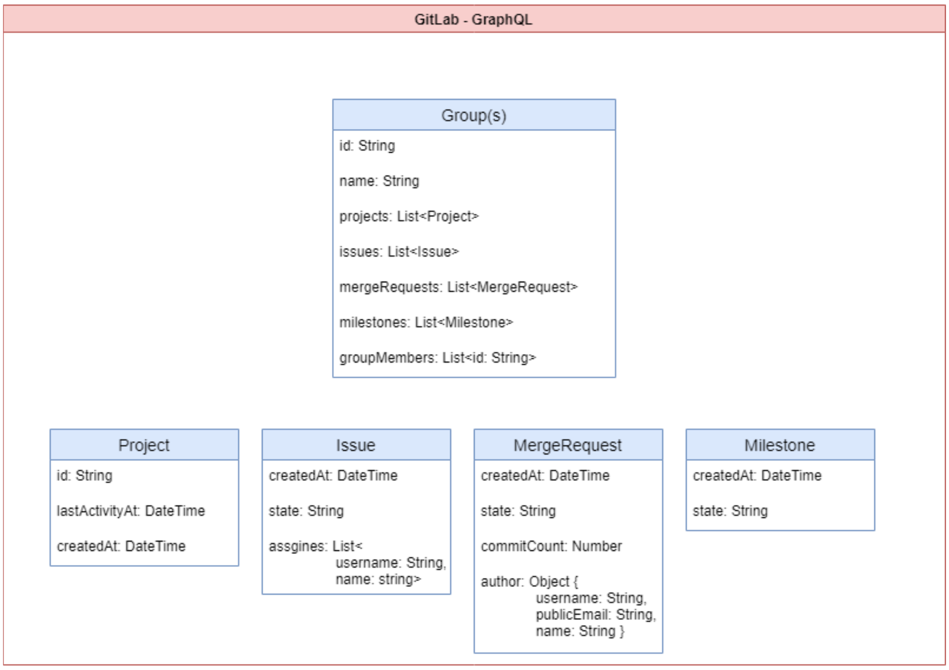


Figure 19 - Data used from GitLab GraphQL

4.10.5 Mapping Models and Data

As the design and flow of the web application prototype are designed to be agnostic of the platform used, we made it so that all the functions we have do not have any coupling with how Blackboard or GitLab work. Created models of how we want the data to look from our LMS and Git hosting platform. Using these models, we created functions for Blackboard and GitLab to retrieve necessary data. All the functions regarding retrieving data from Blackboard and GitLab were placed into their own files and folders. If you want to use GitHub instead of GitLab, you could copy our GitLab folder, make necessary changes and switch the function link used in our application.

If there were any inconsistencies between how we wanted the data, we would restructure it into the model's structure. GitLab for example provides the field "userName" with the key "username", so before returning the data, we remap "username" into "userName". Blackboard provides "name" as an object with the keys "given" and "family", but GitLab provides the string "name" which is given and family combined. We decided that we want "name" as an object, so we had to remap the data from GitLab. Doing this makes supporting something other than Blackboard and GitLab easier. You could look at our models and our implemented functions to create your own for the LMS and Git hosting platform you chose.

We went for using JavaScript as our programming language, JavaScript doesn't have types and doesn't care what the properties of your data are. In hindsight, we should have used TypeScript, which gives you types and requires you to type your custom objects. Now, as things are, we have to make additional documentation of the models/types and how we want the data structure, but with TypeScript, the functions and the data are self-describing with types.

There are probably some requirements for the system you want to integrate into our web application. Now the LMS must in some way have the possibility to provide you with: courses you are a member of, users in your course, groups in your course, members in the groups. The Git hosting platform must have the possibility to provide you with: a hierarchy of parent repository/project/group and sub repository/project/group, the members of the repository/project/group, and data about the code in the repository.

We investigated if any other Git hosting platform other than GitLab meets these requirements: GitHub does not have any way, but GitHub Teams and GitHub Classroom does. Bitbucket has a solution for making sub-groups and giving them access to a repository. So Bitbucket meets the requirements. Azure also seems to support some form of hierarchy and will then be possible to use. So, compatibility with our apps looks good.

For the LMS, we assumed most of them already meet the requirements we have.

4.10.6 GitLab API and Wiki Data

We wanted to get some data about the wiki you can create per project, but GitLab provides no endpoint to retrieve data about the wikis. The wikis on GitLab are stored as projects, and you can use the command `git clone` to download the files, the same you would have done for an ordinary project if you wanted to download it. You would then think you can use the endpoints for projects to retrieve data, but you cannot, and GitLab has possibly marked them as special in some way. The solution may be to download the files and use commands from Git to get data. The disadvantage here is that it can take some time if the wiki is extensive, and we would now have two methods to get data we would have to maintain: over the endpoints and directly download the files.

4.10.7 Authentication

The NextJS authentication package lets you create a secure session on the backend that you also have access to in the frontend. When creating this session, you can call other data points, e.g., Blackboard, to get extra information about the user that you can store in the session. You can use this additional information to check what role a user has in the course, and from that, decide what function the user has access to on the backend and frontend.

The access key you get from Feide/Dataporten cannot be used towards Blackboard or GitLab, so you must get different keys for them.

4.10.7.1 Blackboard

For Blackboard, we got our application registered as an application that retrieves data from the test instance of Blackboard NTNU has. This means we got a set of id/keys that we can exchange for an access key. We can then use this access key to get data from Blackboard. We can then make a request on behalf of a user.

Feide/Dataporten does not use the same key for a user as Blackboard or GitLab, so in our solution, we do: take the userName of the user from Feide/Dataporten, ask Blackboard for users with the same userName, take the result and look for a user with matching userName, take information about the user found (courses, role in the courses), store the information from Blackboard in the session. So, when we retrieve data from Blackboard on behalf of the user, we can access the session on the Backend and check if the user is allowed to access the data on Blackboard using the role the user has in the course.

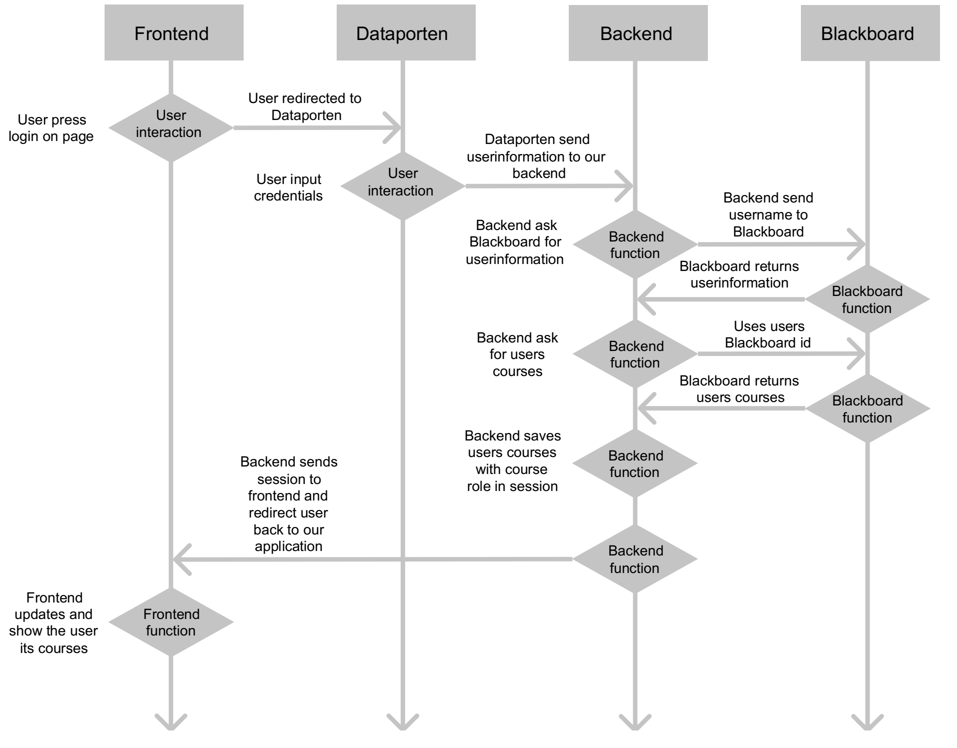

Frontend
Dataporten
Backend
Blackboard
User press login on page
User interaction
User redirected to Dataporten
User input credentials
User interaction
Dataporten send userinformation to our backend
Backend ask Blackboard for userinformation
Backend function
Backend send username to Blackboard
Blackboard returns userinformation
Blackboard function
Backend ask for users courses
Backend function
Uses users Blackboard id
Blackboard returns users courses
Blackboard function
Backend saves users courses with course role in session
Backend function
Backend sends session to frontend and redirect user back to our application
Backend function
Frontend updates and show the user its courses
Frontend function

Figure 20 - Login flow towards Blackboard

4.10.7.2 GitLab

We couldn't get our application connected to Gitlab with the same technique we used for Blackboard. Our solution here was, that when a user logged into a course in our application for the first time, we asked them for a Personal Access Token (PAT) from GitLab. This token you can use to retrieve data from GitLab as if you were the user. We save this token in our database linked with the user's userName, not the most secure method, but it works for our minimum viable application.

Personal Access Token is not considered a password, and it is highly recommended that you create a token per application you use with an expiration date. If you expose your token to a malicious third party, you can delete the token and create a new one. The token is not in any way connected to your password.

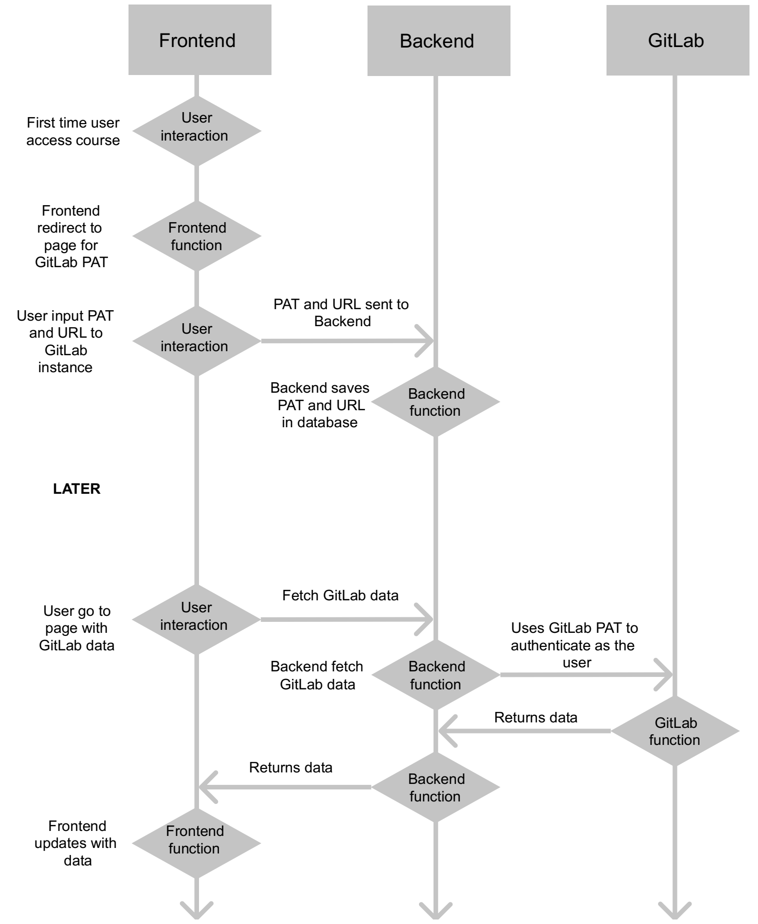

Frontend
Backend
GitLab
First time user access course
User interaction
Frontend redirect to page for GitLab PAT
Frontend function
User input PAT and URL to GitLab instance
User interaction
PAT and URL sent to Backend
Backend saves PAT and URL in database
Backend function
LATER
User go to page with GitLab data
User interaction
Fetch GitLab data
Backend fetch GitLab data
Backend function
Uses GitLab PAT to authenticate as the user
Returns data
GitLab function
Returns data
Backend function
Frontend updates with data
Frontend function

Figure 21 - Authentication flow GitLab

4.10.8 Data Storage

We decided to try to save as little data as possible in our application and instead update the sources Blackboard and GitLab. Blackboard is considered the primary source of data and the truth, but we show the differences with the data from GitLab, so you can decide which of them you want or a bit of both when you update something stored in both.

We wanted this because you can still use the functions in Blackboard and GitLab to update their local data if you're going to do that. This means that you won't become dependent on our application to make changes.

E.g., if we had stored the members of a group in our application, a LA that wants to update the members of a group cannot do it on Blackboard. But when storing all the data on Blackboard or GitLab, you can use our application to edit the members, and we will update the data on Blackboard or GitLab, or you can use Blackboard or GitLab to do the edit if wanted.

The only information we store now is the connection between a course on Blackboard and the main repository for a course on the Gitlab instance and the personal access token for a user to communicate with GitLab.

4.10.9 Sequence Diagram of User Stories

Two diagrams of the flow that happens in two user stories are presented in this section. These are the flows for when a user wants to look at the group list in a course and the group's data, and also when a user wants to check the state of groups and members and possibly update the mentioned state.

4.10.9.1 Flow for Retrieving Group list and Data

Figure 22 shows the flow for retrieving the group list in a course and the corresponding data about the group for GitLab.

The flow starts with the user accessing the page that shows the list of groups and data about the group from GitLab. The loading of the page triggers a call to our backend to fetch the data to be shown. The backend gets the request and retrieves the data required to access GitLab. When the backend has gotten the personal access token (PAT) and the URL of the GitLab instance, it triggers the function to get the list of groups. The function is from our GitLab module. The GitLab function executes, and the list of groups is retrieved. The backend will then retrieve data about the individual groups, again a function from our GitLab module is used. When the data about individual groups is retrieved, our backend aggregates the data and sends it to the frontend to update and show the user the list of groups.

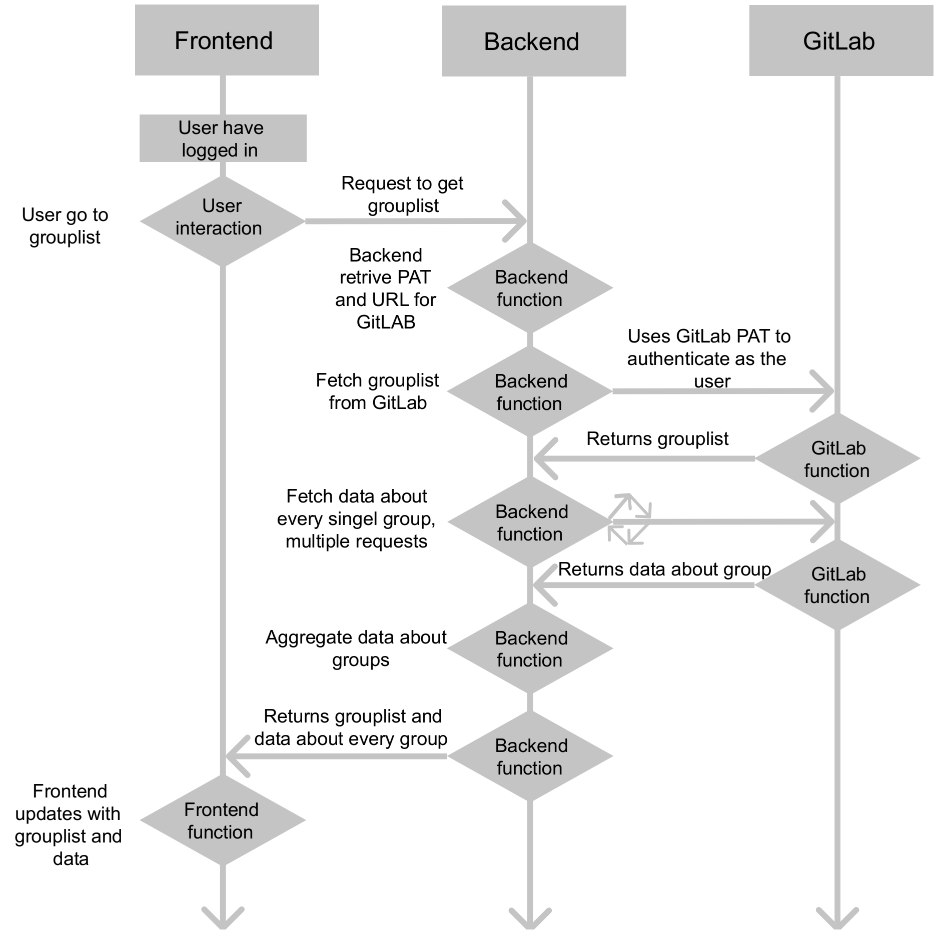

Frontend
Backend
GitLab
User have logged in
User go to grouplist
User interaction
Request to get grouplist
Backend retrive PAT and URL for GitLAB
Backend function
Uses GitLab PAT to authenticate as the user
Fetch grouplist from GitLab
Backend function
Returns grouplist
GitLab function
Fetch data about every singel group, multiple requests
Backend function
Returns data about group
GitLab function
Aggregate data about groups
Backend function
Returns grouplist and data about every group
Backend function
Frontend updates with grouplist and data
Frontend function

Figure 22 - Flow of retrieving group list and data

4.10.9.2 Flow for Checking State of Groups and Update the State

Figure 23 shows the flow for checking the state and difference of groups and members on Gitlab and Blackboard. When the state and difference are displayed to the user, the user can make changes and sync towards Blackboard and GitLab.

The flow is started with the user accessing the page that shows the state of groups and the potential difference of groups in Blackboard and GitLab. A request is triggered to fetch the state of groups from our frontend to our backend. Our backend then retrieves necessary data to access Blackboard and Gitlab, the personal access token (PAT) and the URL of the GitLab instance, and the access token from Blackboard. The backend then uses our Blackboard module functions to retrieve the groups and students on Blackboard and our GitLab module function to retrieve the list of groups on GitLab. When it has gotten the groups and students, it will check if any students are not in a group and calculate the difference between the groups on Blackboard and GitLab. The state of groups is returned to our frontend and shown to the user.

After the state is shown, the user can change the groups and choose to sync the changes to Blackboard and GitLab. Differences between the initial state from the backend and the new state made by the user are calculated, and necessary calls to our backend are made. The backend will then forward the request to Blackboard and GitLab, which will save their local state, and now the state should be the same in both.

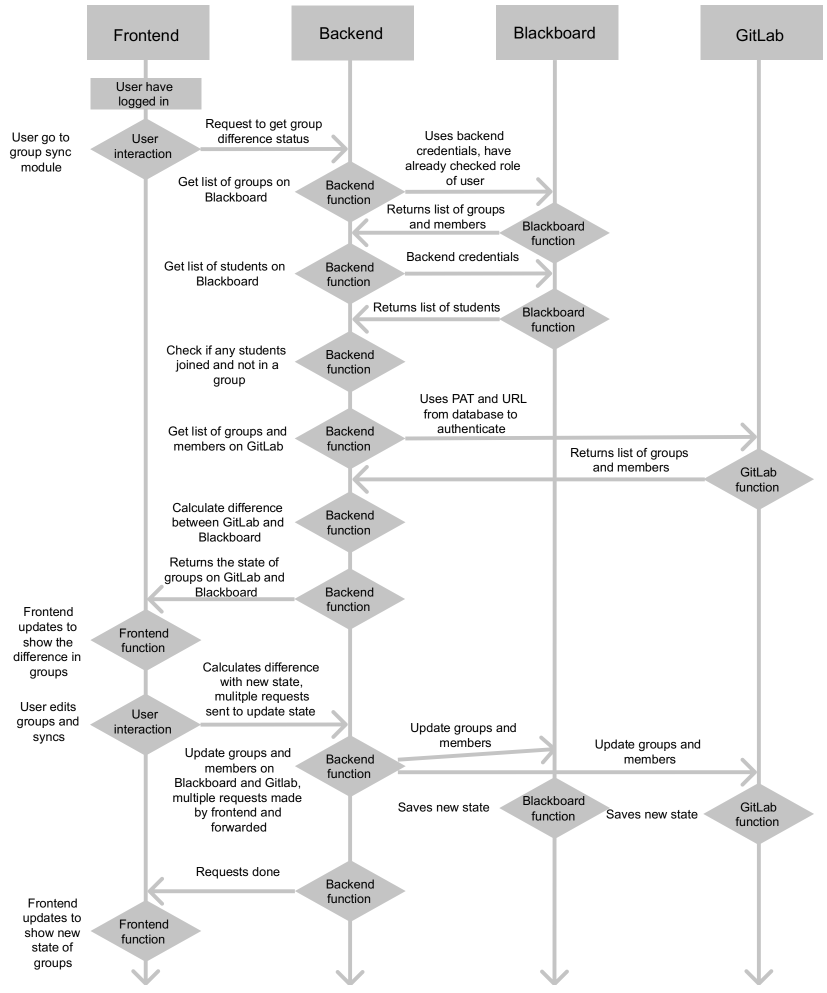

Frontend
Backend
Blackboard
GitLab
User have logged in
User go to group sync module
User interaction
Request to get group difference status
Uses backend credentials, have already checked role of user
Get list of groups on Blackboard
Backend function
Returns list of groups and members
Blackboard function
Backend credentials
Get list of students on Blackboard
Backend function
Returns list of students
Blackboard function
Check if any students joined and not in a group
Backend function
Uses PAT and URL from database to authenticate
Get list of groups and members on GitLab
Backend function
Returns list of groups and members
GitLab function
Calculate difference between GitLab and Blackboard
Backend function
Returns the state of groups on GitLab and Blackboard
Backend function
Frontend updates to show the difference in groups
Frontend function
Calculates difference with new state, mulitple requests sent to update state
User edits groups and syncs
User interaction
Update groups and members
Update groups and members
Update groups and members on Blackboard and Gitlab, multiple requests made by frontend and forwarded
Backend function
Saves new state
Blackboard function
Saves new state
GitLab function
Requests done
Backend function
Frontend updates to show new state of groups
Frontend function

Figure 23 - Flow for checking groups state and update state

4.10.10 Structure of Pages and Components in Next.js

Next.js handles creating pages by making every file in the folder “pages” a page. The file path and the name of the file pages become the url the page is served on. The page located in “pages/homepage.js” will be served on “/homepage”, and the page “pages/schools/ntnu.js” will be served on “/schools/ntnu”.

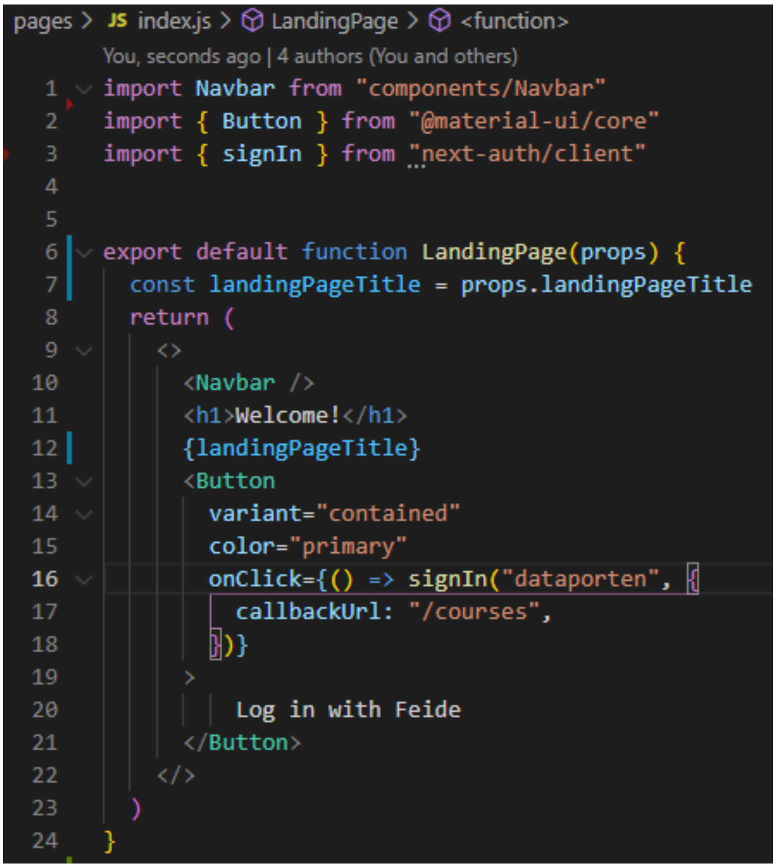
There is also a folder called “api” in “pages”, every function in a file here can be used when serverside renders a page or exported as a REST endpoint.

Every page is made up of “Components” that together create the page. The components can be building Next.js components, third-party components, or self-created components. Components are reusable bits of code stored in a folder called “components” so that they can be used in multiple places and easier to decouple from the pages.

If you want to change the contents on a page, you will edit the components in the page file, and if you're going to change a component, you edit the component file. If you want to make new components, you can create a component from scratch or copy and edit one.

Components can take “props” as inputs and the component will dynamically change/update depending on the props inputted. E.g., the component card that shows stats about groups can show other stats if you provide other props than selected. This makes it easy to extend and modify components if wanted. Props can either come from a parent component or Next.js’ serverside render pages. You can inject props when building the page. Next.js provides three functions you can include in your page that offers the functionality of injecting props to a page. “getServerSideProps” is one of the three functions and is used to inject props when the page is rendered on the server. Here you can safely run backend code and fetch the initial data needed for the page.

Figure 24 shows the code for the landing page component. The file path is “pages/index.js” because “index” is special in Next.js. This page will be served on the root of the application “/”. The name of the component is “LandingPage” and consists of the self-made component “Navbar”, “Button” from Material-UI, and the function signIn from Next.js. It can use the prop “landingPageTitle” to render something on the page dynamically. Because this is not TypeScript, it can be anything.



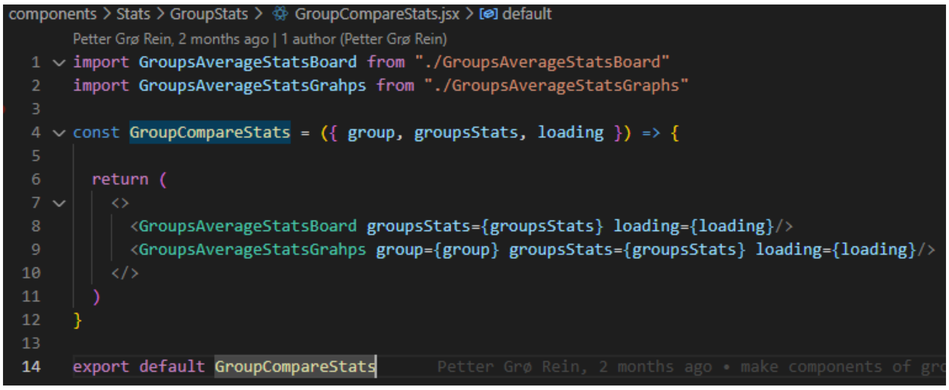
pages > JS index.js > LandingPage > <function>
You, seconds ago | 4 authors (You and others)

```jsx
import Navbar from "components/Navbar"
import { Button } from "@material-ui/core"
import { signIn } from "next-auth/client"


export default function LandingPage(props) {
  const landingPageTitle = props.landingPageTitle
  return (
    <>
      <Navbar />
      <h1>Welcome!</h1>
      {landingPageTitle}
      <Button
        variant="contained"
        color="primary"
        onClick={() => signIn("dataporten", {
          callbackUrl: "/courses",
        })}
      >
        Log in with Feide
      </Button>
    </>
  )
}
```

Figure 24 - Code for landing page

Figure 25 shows the code for the group compare stats component. This component consists of two other components. If you want to change this component, you can add a new component or edit/copy one of the already existing components.



components > Stats > GroupStats > GroupCompareStats.jsx > default
Petter Grø Rein, 2 months ago | 1 author (Petter Grø Rein)
Petter Grø Rein, 2 months ago • make components of grc

```jsx
import GroupsAverageStatsBoard from "./GroupsAverageStatsBoard"
import GroupsAverageStatsGrahps from "./GroupsAverageStatsGraphs"

const GroupCompareStats = ({ group, groupsStats, loading }) => {

  return (
    <>
      <GroupsAverageStatsBoard groupsStats={groupsStats} loading={loading}/>
      <GroupsAverageStatsGrahps group={group} groupsStats={groupsStats} loading={loading}/>
    </>
  )
}

export default GroupCompareStats
```

Figure 25 - Code for Group Compare Stats Component

5 Iterations and Changes

The implementation of the project evolved around an iterative user-centered design and development process. Four iterations were completed, iteration 0 focusing on a Figma prototype, while iteration 1, 2, and 3 focused on the actual implementation of the application. The events of all the iterations are visualized in Figure 26, as well as in the first four sections of the current chapter.

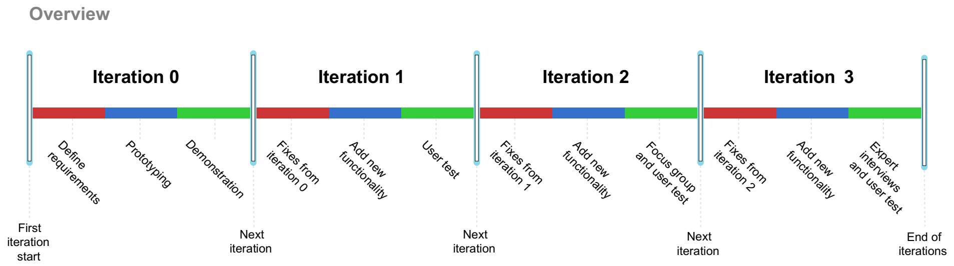


Figure 26 - Timeline of all iterations

5.1 Iteration 0 – The Figma Prototype

In what was considered iteration 0 of the iterative design process of the application and as a part of the user-centered design approach, we decided to prototype the application. While wireframes on paper would have been the fastest and probably most efficient way to start prototyping, it was decided that prototyping in Figma would be beneficial due to the prevailing circumstances making collaboration on paper challenging. The prototypes were created in November 2020, with the aim of getting feedback in December and making a functional prototype during January.

Due to a strict work from home policy in Trondheim, traditional user testing of the prototype was not an option, and it was therefore decided that we replaced user testing of the Figma prototype with asking our primary stakeholder to click through the prototype in Figma and provide feedback through the comment section of the software. We later discussed the feedback and other thoughts in a video meeting.

The main events of Iteration 0 are visualized in Figure 27, and described in the sections of this chapter.

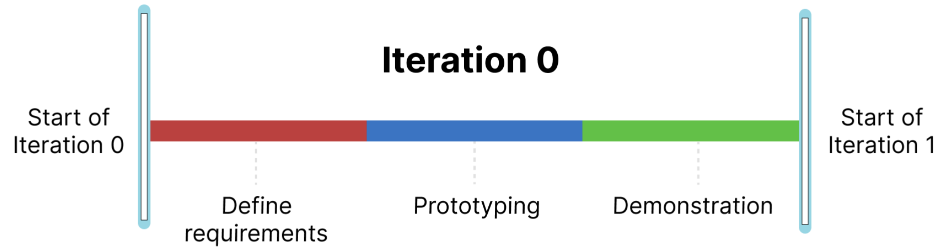


Figure 27 - Timeline of Iteration 0

5.1.1 Defining Requirements

Before being able to prototype an application it was necessary to define what to include in the prototype. This was done based on the work in chapter 4.

One of the most important requirements was that the application should be accessible on both mobile devices and larger devices, the prototype was designed using an approach sometimes referred to as "mobile first". In this project, this means that the prototype views were initially designed to fit a frame that resembles a mobile phone with a viewport having more height than width, commonly referred to as portrait orientation. Another important requirement in the prototyping process was that the application should be designed according to the design guidelines defined in section 2.5.5, to ensure a high level of usability.

5.1.2 Prototyping

The prototype was created using the online tool Figma, where both team members could collaborate in real-time despite the work-from-home policies enforced at that time.

While prototyping the team had requirements from chapter 4 in mind, while also following the design guidelines from section 2.5.5.

The prototype views can be seen in appendix B of this thesis, and a few of them will also be discussed in section 5.1.3.

5.1.3 Feedback from Primary Stakeholder

After creating the Figma prototype, it was decided that it should be shown to one of our stakeholders in a demonstration, so that we at an early stage of the process could receive feedback, so the team could avoid spending a lot of time and resources on developing application functionality that would later be removed or reworked.

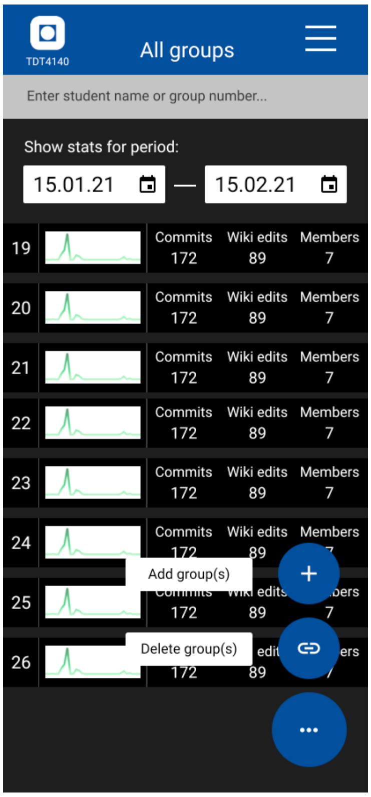
The findings related to the demonstration with the primary stakeholder can be found in Table 17.

ID	Feedback	Comment
0.1	The Floating Action Button seems unnecessary.	The team agreed that the FAB was unnecessary at this stage. Could be introduced at a later stage.
0.2	Some pages have too many elements visible at the same time.	The team agreed that some of the pages and views contained too much information, and some of it will be moved to another view or removed in later iterations.
0.3	The text seems to be a bit too small in some places.	After looking at the prototype on a mobile device, the team agreed that the text size should be increased for better readability. This needs to be seen in relation with feedback finding 0.2.
0.4	Unsure why there is a separate view only containing one button "log in with Feide"	The reasons why the view with the "log in through Feide button" exists is to simplify the addition of other authentication methods at a later stage, and to have the possibility to provide information to users before redirecting them to their authentication provider.
0.5	The prototype does not contain a lot of visualizations.	Visualization requires a lot of time and resources to prototype, but the team increased focus on visualizations after receiving this feedback.
0.6	Functionality for communicating with groups could remove focus from main issues.	The team agreed with this prioritization, so functionality for communicating with groups was postponed into later iterations, as this could be done in other ways without being critical for the usefulness of this application.

Table 17 - Findings from Figma prototype demonstration

With relation to feedback ID 0.1, our stakeholder stated that the FAB could be considered confusing, due to the application having at least two places where seldomly used options could be hidden away. He therefore proposed not implementing the FAB until it was necessary.

As shown in Figure 28, navigation in this prototype is handled by three mechanisms. The Floating Action Button (FAB), the navigation drawer and traditional buttons.



Show stats for period:

15.01.21 — 15.02.21

	Commits	Wiki edits	Members
19	172	89	7
20	172	89	7
21	172	89	7
22	172	89	7
23	172	89	7
24	172	89	7
25	172	89	7
26	172	89	7

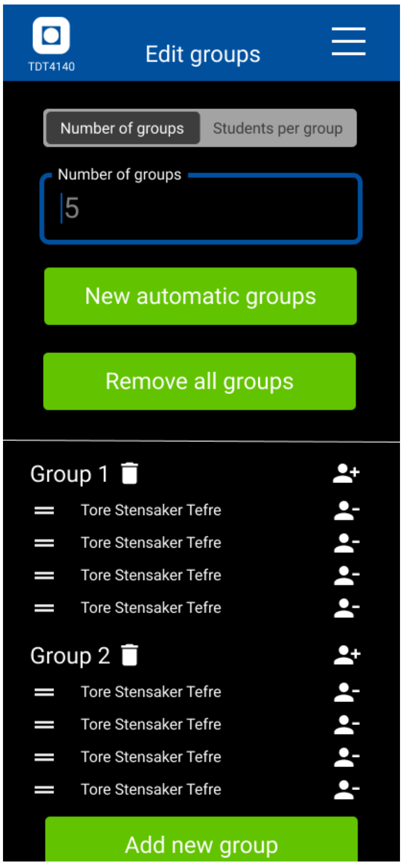
Figure 28 - Floating Action Button (FAB) screenshot (Mobile)

The navigation drawer is based on the navigation drawer from Material Design but is in our prototype placed in the upper right corner of the application, opposite to the traditional upper left corner placement. The rationale for this decision was the idea that a user might want to return to the dashboard, and that this possibility should always be present. The idea is to have a logo that the user can click on in the upper left corner, to return to the dashboard at any time. In line with our design guidelines, where the importance of consistency is discussed, and that this is a very common approach. Additionally, we believed that the upper right corner would be more accessible for right-handed users on mobile devices.

The prototype also features the Floating Action Button, shortened FAB. Similar to the navigation drawer, the FAB is a part of Google's Material Design project. As per the creators [40], the FAB "performs the primary, or most common, action on a screen." While some argues that the FAB isn't necessarily always the best solution [41], the FAB was seen as beneficial in this application. In this prototype the FAB gives the users access to the most common actions that can be done with the content on the page the user has visited.

FAB was discontinued as it was no longer seen as necessary as proposed by the stakeholder, and also because the navigation drawer in the upper right corner could offer much of the same functionality.

Related to feedback ID 0.2 and 0.3, the page for editing groups is one of those where there are too many elements and too small text. The page is shown in Figure 29. Measures here could include splitting the functionality into two different pages, or make some of the content collapsable. Changing the layout of the page will also be considered.



Number of groups: 5

Group 1
Tore Stensaker Tefre
Tore Stensaker Tefre
Tore Stensaker Tefre
Tore Stensaker Tefre

Group 2
Tore Stensaker Tefre
Tore Stensaker Tefre
Tore Stensaker Tefre
Tore Stensaker Tefre

Figure 29 - Create groups proposal with edit option (Mobile)

5.2 Iteration 1

Iteration 1 was based on the prototype created in Figma, which to some degree was replicated as the first iteration of the application, adjusted in line with the feedback the team received before beginning development work.

Iteration 1 included the initialization and setup of the application, as well as the implementation of necessary functionality related to connecting the web application with software such as GitLab, Blackboard, and authentication through Feide user accounts.

The main events of Iteration 1 are visualized in Figure 30.

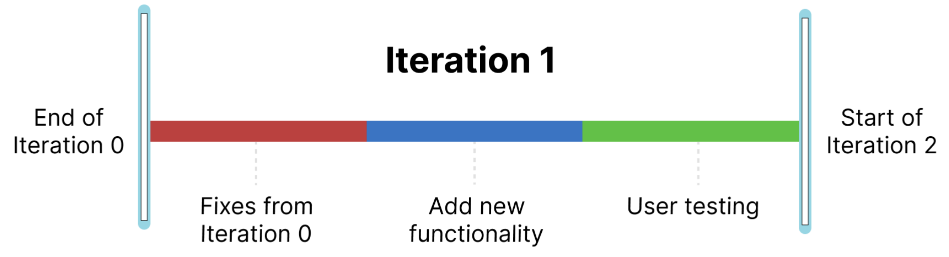


Figure 30 - Timeline of Iteration 1

5.2.1 Improvements from Figma Prototype

Based on feedback and impression from the evaluation of the Figma prototype, changes from the Figma prototype were implemented as described in this section.

One of the changes was related to the floatable action button (FAB). While the FAB was considered a good idea in the prototype, it became clear that it would be unnecessary, at least in the first iterations. It was therefore removed as discussed in section 5.1.3.

Another change is that when creating the connection between a course on Blackboard and a GitLab instance the user is now asked for a personal access token (PAT) together with the URL to the GitLab CE instance of choice. The token is used to authenticate the user on the GitLab instance from the URL field. This makes it possible for the person that creates the connection to choose which instance the person will use and not being limited to an instance decided by the maintainers of the application.

5.2.2 New functionality

The new functionality of Iteration 1 is presented in Table 18.

ID	User story
1	I would like to log in using my existing Feide user account.
2	I would like to create a connection to my selected GitLab CE instance, using a personal access token.
3	I would like to see a list of courses based on the courses I am registered as a lecturer in Blackboard.
4	I would like to see a dashboard for a course.
5	I would like to create groups based on Blackboard generated CSV files.
6	I would like to create groups based on a given number of groups or a given number of students in a group.

Table 18 - User stories implemented in Iteration 1

5.2.3 User Testing

The purpose of arranging and user test after Iteration 1 was to identify usability issues at an early stage of the development, such as features not being easy to use, misleading text and labels, and see if the proposed menu system works as intended.

The tasks were designed to test the usability of the functionality developed since starting the development of the application prototype. All tasks in the user testing are based on the assumption that the test participant is the head learning assistant in the course _57_1. The tasks are presented in Table 19, while the findings from the user test are presented in Table 20.

#	Task	Focus areas
1	I would like to log in to the application, using my already existing Feide credentials from NTNU.	Authentication
2	I would like to create a course in the application, and have it connected to courses in Blackboard and GitLab.	Initial setup
3	I would like to create student groups in the application, and have them created in GitLab and Blackboard automatically.	Group creation

Table 19 – Tasks for user test after Iteration 1

ID	Feedback	Comment
1.0	It would be useful to see statuses and synchronization statuses of the students and groups in Blackboard and GitLab.	The team agrees, and functionality for this will be prioritized in the next iteration. This functionality will help the users to confirm that all groups are correctly synchronized and help them with correct actions if not.
1.1	The application behaved slowly and waiting for data was common.	Due to the usage of REST APIs, the number of requests became quite high, and a rework of many of those requests to GraphQL would reduce loading time.
1.2	Some of the data expected did not load.	The team observed this issue as it happened, and the main reason for the issue was rate limitations imposed by the GitLab instance we were testing against. The team believes this issue will be removed by changing some of the requests done by the application from the REST API to the GraphQL API.

Table 20 - Findings from user tests after Iteration 1

While evaluating the prototype after Iteration 1, the need for a group differentiation module appeared as described in feedback ID 1.0. The problem here was to handle the situations where the groups appearing in Blackboard and GitLab were not matching. Due to a number of reasons explained in the technical chapter, it was decided that we wanted to store as little data as possible in our systems. This meant that looking at previous group data was not an option, and that we would need to solve this challenge using a different strategy.

An option to solve this problem was to consume APIs containing logs of actions taken in the source systems regarding adding and deletion of users from a group, but none of the APIs available was considered to be useful for our scenario.

As most users of the web application would have extensive knowledge of Git and related software, it was decided to take inspiration from Git. While using Git’s so-called branches of code to compare code blocks from different branches to see it they are mismatching.

During one of the user tests the participant experienced rate limitations with the GitLab APIs, which impacted the loading of several elements, and resulted in data not being visible in the application. This is mentioned in feedback ID 1.1. Some of the reasons for this is that some of the data fetching was happening through the GitLab REST API, requiring a large amount of requests to fetch all necessary data. This can also be related to feedback 1.2, where some testers reported that felt the application was behaving slowly.

5.3 Iteration 2

Iteration 2 focused on implementing the necessary pages and functionality for group administration. The main events of Iteration 2 are visualized in Figure 31.

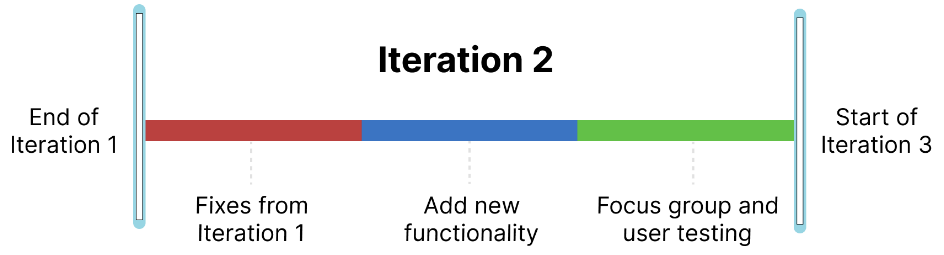


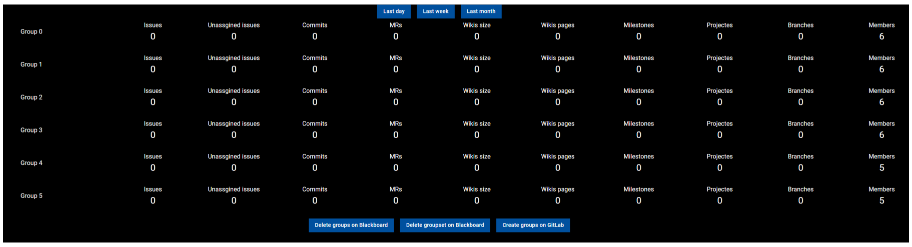
Figure 31 - Timeline of Iteration 2

5.3.1 Improvements from Iteration 1

A number of changes and improvements were done based on feedback from the user tests following Iteration 1. The most important changes are discussed in this section.

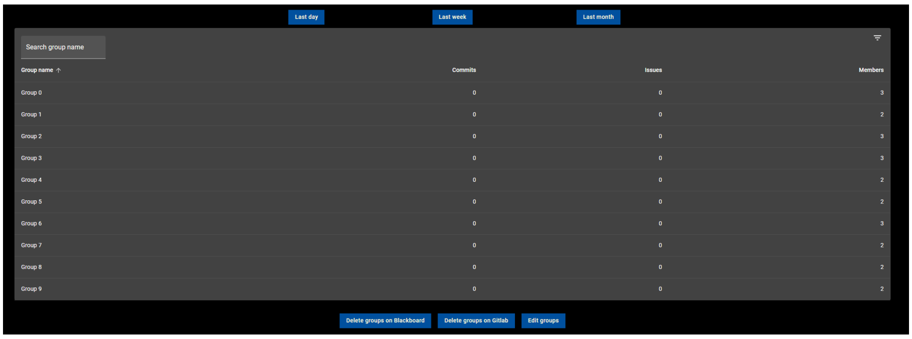
5.3.1.1 Changed layout for group list

Here we changed into using the Material-UI default list, enabling sorting and selection of metrics, which was challenging to implement using the initial custom tiles. The changes are shown in Figure 32 and Figure 33. The after screenshot shows fewer stats, but with the filter button in the upper right corner it is possible to enable and disable additional stats, with three different stats enabled at the time of screenshotting.



	Issues	Unassgined issues	Commits	MRs	Wikis size	Wikis pages	Milestones	Projectes	Branches	Members
Group 0	0	0	0	0	0	0	0	0	0	6
Group 1	0	0	0	0	0	0	0	0	0	6
Group 2	0	0	0	0	0	0	0	0	0	6
Group 3	0	0	0	0	0	0	0	0	0	6
Group 4	0	0	0	0	0	0	0	0	0	5
Group 5	0	0	0	0	0	0	0	0	0	5

Figure 32 - Group list in desktop before changes



Group name ↑	Commits	Issues	Members
Group 0	0	0	3
Group 1	0	0	2
Group 2	0	0	3
Group 3	0	0	3
Group 4	0	0	2
Group 5	0	0	2
Group 6	0	0	3
Group 7	0	0	2
Group 8	0	0	2
Group 9	0	0	2

Figure 33 - Group list in desktop after changes

5.3.1.2 Reengineered data fetching

As discussed in feedback ID 1.1 and 1.2, the application behaved slowly and sometimes failed to fetch the data requested by the user. This was reworked in Iteration 2, with two main changes done to address those issues.

First and foremost, several of the requests were changed from REST API requests into GraphQL API requests. This allowed the team to greatly reduce the number of API requests needed, which had two large benefits. Most importantly, the rate limits imposed by the GitLab CE instance were not important anymore, but the new data fetching also required a lot less time to fetch the required data.

The time spent loading data was also reduced greatly with the introduction of request caching. Caching meant that data that had already been requested by the application was not necessary to fetch before a given time had passed. This greatly reduced the time needed before being able to display data the user had just seen, while also reducing the number of requests initiated from the application.

5.3.1.3 Created group difference page

During the user test, as addressed in feedback ID 1.0, one of the participants expressed that keeping groups synchronized in both of the source systems would be challenging. The team therefore proposed a view where group sync status could be seen, and implemented this in iteration 2. This view is shown in Figure 30.

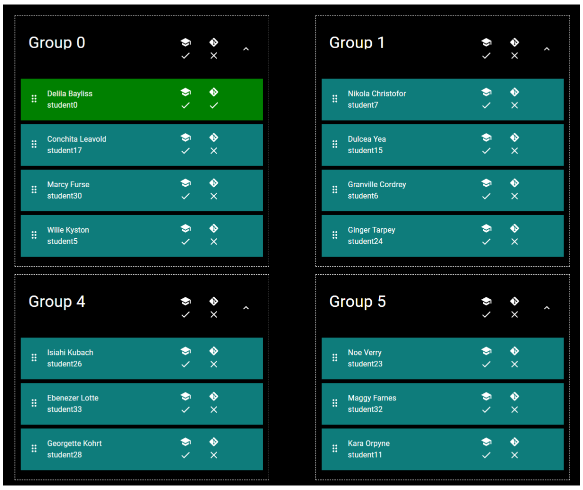

Group 0 ✓ ✕
Delila Bayliss student0 ✓ ✓
Conchita Leavold student17 ✓ ✕
Marcy Furse student30 ✓ ✕
Wilie Kyston student5 ✓ ✕
Group 1 ✓ ✕
Nikola Christofor student7 ✓ ✕
Dulcea Yea student15 ✓ ✕
Granville Cordrey student6 ✓ ✕
Ginger Tarpey student24 ✓ ✕
Group 4 ✓ ✕
Isiahi Kubach student26 ✓ ✕
Ebenezer Lotte student33 ✓ ✕
Georgette Kohrt student28 ✓ ✕
Group 5 ✓ ✕
Noe Verry student23 ✓ ✕
Maggy Farnes student32 ✓ ✕
Kara Orpyne student11 ✓ ✕

Figure 34 - Group sync view

5.3.2 New Functionality

The new functionality of Iteration 2 is presented in Table 21.

ID	User story
5	I would like to see a list of all students in a course.
6	I would like to see a list of all student groups in a course.
7	I would like to see a list of all course staff in a course.
8	I would like to see a list of all groups created in Blackboard.
9	I would like to create GitLab groups and projects automatically based on the groups in the application.
10	I want to sort/filter the list of groups based on the stats shown
11	I want do compare the groups state on Blackboard with GitLab

Table 21 - User stories implemented in Iteration 2

5.3.3 User Testing

The purpose of user testing Iteration 2 was mainly to identify usability issues in the group sync module, and to get feedback on the navigation flow of the application as functionality was added.

The tasks were created based on new functionality since Iteration 1, where the previous user tests were conducted. All tasks in the user testing are based on the assumption that the test participant is the head learning assistant in the course _57_1, and are presented in Table 22.

ID	Task	Focus areas
1	Check if the groups are correctly synchronized to GitLab and Blackboard and take action to fix any issues you may encounter.	Group sync module
2	Check if any students are not in a group and if any, assign them to a suitable group.	Group sync module
3	Based on the groups already existing in the application, I would like to create matching groups and project repositories in GitLab.	Group creation in GitLab
4	Sort all groups in your course based on a metric that is not displayed in the application by default.	New group list

Table 22 – Tasks for user tests after iteration 2

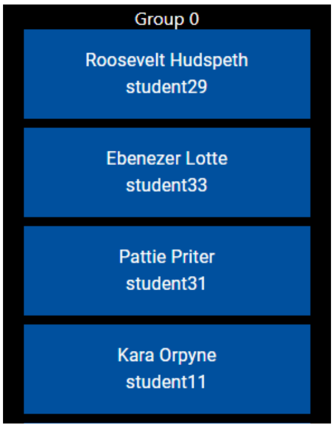
The findings from the user tests after Iteration 2 are presented in Table 23, while some of them are also discussed in detail later in this section.

ID	Feedback	Comment
2.0	Drag and drop functionality was not affordable enough.	The team agrees, and measures will be taken in the next iteration.
2.1	The text on the button for starting group synchronization was considered unprofessional stating "Click me to sync".	The team agrees that the text is too long, and the text will be replaced with a more professional phrase.

2.2	Some participants were unsure what the icons in the group status page indicated.	The team agrees that this is an issue. However, we believe that the implementation is not the problem, but that the user in a better way should be informed about what the icons mean.
2.3	During the user testing of iteration 2, the test leader noticed that some of the testers developed a habit of counting the number of students in the groups, leading the team to believe that a count should be shown.	While chatting after the user test was completed, the test leader proposed that some sort of counter could be implemented, showing at the top bars of the group boxes. The participants agreed, and it is therefore being implemented.
2.4	Lack of feedback after pressing the sync button in group view.	The team agrees that the feedback should be improved. The issue is discussed thoroughly at the end of this section.
2.5	Participants were surprised when the group showing students without a group disappeared when all students had been assigned to a group.	This problem could have been solved in different ways, but the most obvious solution would be to let the box stay present in the layout even when empty.
2.6	Some participants are requesting a box for students not really participating in the practical activities of the course	To avoid duplicate storage of data, students not participating in the group work should be placed in a separate box. To save this information, the group could either be synced to Blackboard as one single group containing all the students not participating in the group work, or they could be synced to individual groups in a common group set, if they should not be able to see each other.

Table 23 - Findings from user tests after Iteration 2

One of the feedbacks to be discussed is feedback ID 2.0, where one of the test subjects reported that when presented with functionality requiring the use of drag and drop it was not obvious at first that drag and drop was the required interaction needed to move users in the group sync module. The functionality in question can be seen in Figure 35, and was improved in iteration 3.



Group 0

Roosevelt Hudspeth
student29

Ebenezer Lotte
student33

Pattie Priter
student31

Kara Orpyne
student11

Figure 35 – Drag and drop with limited affordance

Another feedback we received, identified as feedback ID 2.2, was that during some of the tasks requiring use of the group sync module, one of the test subjects expressed a belief that the icon that had been chosen to reflect the synchronization status to Blackboard was a representation of whether the student was class registered and / or exam registered in the FS source system. The participant believed this was the icon for class registration, which is a status coming from FS. After researching our possibilities to get this data, the team decided to contact our contact at NTNU IT regarding this matter, and he confirmed that we needed to go through FS to get information about exam registrations. The team decided that this use case could be considered an edge case and combined with the large efforts such an integration would require, it was decided it was not a priority.

Mentioned in feedback ID 2.4 was the issue that while performing task 4, all testers reacted to the lack of feedback after clicking the synchronization button at the bottom of the group sync module. A possible solution to this usability issue could have been to display some sort of message after the operations initialized by the button click event was finished. Another solution that was proposed was to redirect the user away from the group sync module, assuming that the user was finished working on the tasks that were possible to do in that environment. However, we could not be sure of that, and the mentioned approach could have been suitable to confuse some users. That led us into the third possible solution which was to update the data on the page in such a way that the user easily understands that their changes have been carried out.

Based on previous experiences, it was decided that a combination of the first and the last proposed solution could be implemented. This means that the user will be informed on how the changes are carried out both explicitly through a toast notification and implicitly because the changes are reflected in the page where the icons and colors representing student status in the groups changes.

5.3.4 Demonstration with Primary Stakeholder

During Iteration 2, the team had a demonstration with the primary stakeholder, where the plans for Iteration 3 and some early functionality that was going to be implemented during Iteration 3 was discussed. The stakeholder expressed some insights and functionality wishes, which are listed in this section.

5.3.4.1 See unassigned issues

In many software development projects the use of unassigned issues represents development tasks that are currently not being worked on. A large number of unassigned issues may indicate different things, depending on the situation. For example, a large number of unassigned issues in the beginning of a new project may indicate that the project team plans the work of the team well in advance, while a large number of unassigned issues after a project deadline may indicate that not all possible features or necessary project work was completed.

Based on the feedback and reasoning, data about unassigned issues was included in the application during iteration 3.

5.3.4.2 Distribution of work

According to many of the people that were interviewed in the initial phase of this project, work distribution amongst the members of the groups is a common field of interest. The distribution of work can be considered based on a number of variables, and some of them have been specifically mentioned by stakeholders in this project, including the distribution of commits, code additions, code deletions, issues, code reviews and merge request creations.

Based on the feedback from our supervisor in this round, basic functionality for this concern was selected as a priority and implemented during iteration 3. The implemented solution includes a modal window where the data values of interest can be filtered to ensure that the users are able to focus on their data of interest.

5.3.4.3 Issues with no title, description, or discussion

During the demo our supervisor mentioned that he wanted to see whether the issues created by the groups were correctly formatted and containing necessary elements, including issue title, description and discussion. Regarding the title, it is at the time of writing not possible to create GitLab issues without a title, a behavior shared by fellow version control software GitHub.

Regarding the description and the discussion, providing meaningful insights about the group work based on these parameters is believed to be difficult due to a number of reasons. Among these is the hypothesis that the sole length of the merge request description does not tell much about the quality of the merge request and the work that lies behind it. Some merge requests may only require a single sentence of text to be easily understood, while some other merge requests may require numerous paragraphs of text to be understood. It is of course possible to check if the description is not empty, but this check could easily be circumvented by adding some characters.

Regarding checking for discussion posts on merge requests, our hypothesis is that while the number of discussion posts may make it possible to understand more about how the group works, it may not be a very good indication of the quality of the group work. The count of discussion posts may be affected by automated posts related to automatic pipelines, and the number of posts is not necessarily connected to the quality of the discussion. Discussion may also happen outside of the discussion functionality in GitLab, for example directly between the involved persons.

5.3.4.4 Check if code reviews are performed by the code author

When introducing new code and features into a software development project, a common procedure is to examine the code before it gets merged into the production code, a process typically called a code review. Code reviews are typically facilitated through the merge request functionality in GitLab, and stats related to merge requests can therefore indicate if and how code reviews are performed in the groups.

Specifically mentioned in the feedback was the information about whether or not the same person that opened the merge request or wrote the code in the merge request was the same person performing the code review. This can be indicated through GitLab data, by looking at the author of the merge request and the person set as the reviewer of the merge request. However, there may be edge cases where the developer may not create the merge request, and the reviewer has to create it instead. Checking if the author of the merge request and the reviewer is the same person will in that case indicate that the author also is the reviewer, not in line with the intentions of the check.

It was decided that this functionality was not to be implemented because making sure that the data was valid was decided to require too many resources compared to the potential benefit of the functionality.

5.3.4.5 Contrast between colors and text

Our stakeholder expressed uncertainness that the contrast between the text colors and the interaction elements and background colors were not ideal. This potential issue has been thoroughly discussed and covered in section 5.4.1.

5.3.4.6 Explanation of colours in group sync module

In the group sync module, a color system was implemented, aiming to make it easier for the users to understand what actions required attention. Multiple user testers commented that these colors should have been explained, as the meaning of each color was not obvious.

A possible solution to improve usability here was to introduce some sort of explanation of what the different colors represent.

5.4 Iteration 3

Iteration 3 was about visualizing activity in the groups, and presenting the data that we had fetched out of the source systems. The main events of Iteration 3 are visualized in Figure 36.

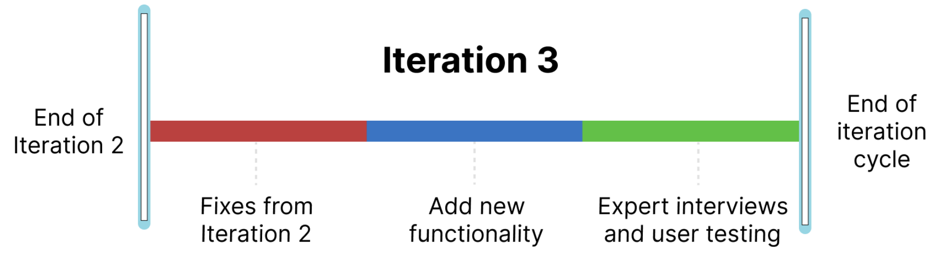


Figure 36 - Timeline of Iteration 3

5.4.1 Improvements from Iteration 2

Based on the feedback from the user tests and the focus group a number of improvements were considered in Iteration 3.

5.4.1.1 Missing counter in top of group module box

During the user testing of iteration 2, the test leader noticed that some of the testers developed a habit of counting the number of students in the groups. While chatting after the user test was completed, the test leader proposed that some sort of counter could be implemented, showing at the top bars of the group boxes.

This was implemented in iteration 3, as shown in Figure 37.

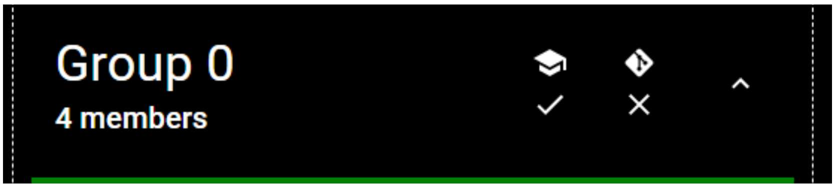


Figure 37 - Added count of members

5.4.1.2 Contrast between background and text

Our stakeholder raised concern about the contrasts between text, interaction elements and backgrounds, as identified in section 5.3.4.5.

Inspired by the default NTNU color scheme, many of the elements in the application uses a combination of the blue color from the NTNU logo (#00509e), and plain white (#FFFFFF). Readability of text is a matter of relative luminance, as explained in WCAG 2.1. [15] To avoid the issue, we tried replacing the plain white with cornsilk, an off-white color, and plain black to see if they proved more readable.

As color choices and readability is related to accessibility and universal design, the Web Content Accessibility Guidelines, commonly referred to as WCAG, is worth mentioning. At the time of writing, WCAG 2.2 is still considered a Working Draft, leaving WCAG 2.1 as the current guidelines, published as a W3C recommendation in June 2018. WCAG 2.1 lists as success criteria 1.4.3 that text and images of text should at a minimum have a contrast ratio of at least 4,5 : 1, while that to reach level AAA text and images of text should have a contrast ratio of at least 7 : 1. This contrast ratio can be calculated using a formula defined in WCAG 2.1 as (L1 + 0.05) / (L2 + 0.05), where

- L1 is the relative luminance of the lighter of the colors, and
- L2 is the relative luminance of the darker of the colors.

These calculations have been done with a selection of colors from our palette, and the results can be found in the table below.

Background color	**Foreground color**	**Contrast ratio**	**Result**
#00509e (NTNU blue)	#FFFFFF (white)	7,95 : 1	AAA Pass
#00509e (NTNU blue)	#FFF8DC (cornsilk)	7,46 : 1	AAA Pass
#00509e (NTNU blue)	#FF0000 (yellow)	7,40 : 1	AAA Pass
#00509e (NTNU blue)	#FF0000 (red)	1,99 : 1	Fail both AA and AAA
#00509e (NTNU blue)	#000000 (black)	2,64 : 1	Fail both AA and AAA

Table 24 - Contrast ratios for selected text colors

Based on these calculations there is not a very large difference in contrast ratios between the different white variants. Due to the perceived readability observed by our stakeholder and the team, it was decided that cornsilk (#FFF8DC) should be used as the white text color on the NTNU blue (#00509e) backgrounds.

5.4.1.3 Improve sync button

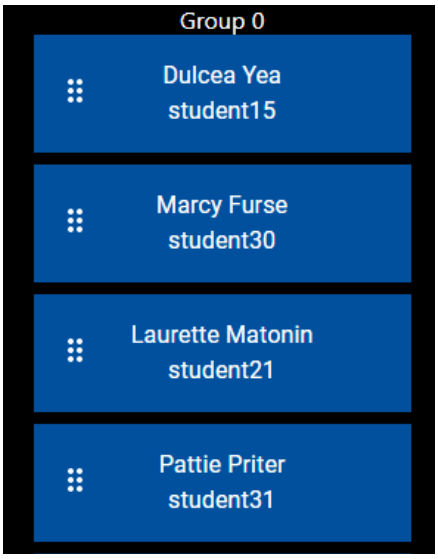
In the group sync module the button on the bottom of the page said "Click me to sync". One of the users reacted to the text on this button and mentioned that it sounded a bit unprofessional. The team agreed on this, and the text was replaced.

5.4.1.4 Improve feedback when changing and loading pages

Implemented skeleton components with a wave loading animation that is shown when pages and data loads, which gets replaced with the real component when the data have finished loading.

5.4.1.5 Increase affordance in drag and drop in group creation and group sync module

During user testing, we received feedback that it was not obvious that some elements were possible to drag and drop to move, suggesting that the affordance of that functionality was a problem. To combat this usability issue, we introduced an icon on those elements that are consistent with how many other modern web applications are designed. The result can be seen in Figure 38.


Group 0

⠿ Dulcea Yea
student15

⠿ Marcy Furse
student30

⠿ Laurette Matonin
student21

⠿ Pattie Priter
student31

Figure 38 – Drag and drop after increasing affordance

5.4.2 New Functionality

The new functionality in Iteration 3 is presented in Table 25, and focuses mainly on improved data visualization, especially in the group pages.

ID	User story
12	I would like to see data such as commits, issues and branches for a single group.
13	I would like to see data such as commits, issues, and branches for all groups in a listed view.
14	I would like to see data for a group visualized using various charts.
15	I would like to be able to see the average data for all groups.
16	I would like to compare the group I am looking at with all the other groups using charts.

Table 25 - User stories implemented in Iteration 3

5.4.3 User Testing

As part of the user-centered design process user tests were conducted with learning assistants with the purpose of identifying any usability issues and room for improvement after implementing new functionality in iteration 2 and 3.

The tasks for the user tests after Iteration 3 were created based on new functionality added after iteration 2, where the previous user tests were conducted.

All tasks in the user test are based on the assumption that the test participant is the head learning assistant in the course _57_1. The tasks for the user test are presented in Table 26, and the findings are presented and briefly discussed in Table 27.

#	Task	Focus areas
1	You want to view a list of all the groups, and view listed data about the commits, issues, and numbers of files for every group.	Group list, especially filtering options
2	You want to identify the group with the largest number of issues, so that you can ask them if their project is going as planned, and that they are using GitLab correctly.	Dashboard or group list, can be solved multiple ways
3	You are suspecting that someone is committing files in the code repositories that should not be there, and therefore want to identify the four groups having the highest number of files in your course, so that you at a later time can check what files they have committed.	Group list, sorting
4	You are not sure if the groups are synchronized between Blackboard and GitLab. Therefore, you want to check if all group members are added to the correct groups in Blackboard and GitLab.	Edit groups functionality
5	You want to determine what student in group 45 has had the most commits the previous year.	Group dashboard, pie charts or student list
6	You want to see how group 45 were performing during the second period of the course, between the 1st of October 2020 and the 2nd of November 2020. The group has complained that one of the members had few contributions in this period, and you want to verify this.	Group dashboard, date picker and charts
7	You are unsure how group 45 have performed compared to other groups. Therefore, you want to see how group gr2045 are doing in the whole work period from the 1st of August 2020 to the 15th of December 2020, when looking at issues, commits, and merge requests.	Group dashboard, comparisons functionality and charts
8	You have been informed by one of the student learning assistants that one group member in group 49 had a lot of commits at one given date. You want to find out what date this happened and who committed, so that you at a later time can investigate the commits and guide the committer if the work could be improved.	Group dashboard, charts for distribution of commits and area chart for commits

Table 26 - Tasks for user test after Iteration 3

ID	Feedback	Comment
3.0	Date format was found to be confusing for Norwegian users.	Lack of configuration resulted in dates being shown in the format month/day/year, but this should be easily fixable.
3.1	Button for "Edit groups" is hidden when the list of groups is long. This results that the buttons of the page will not be visible on the screen if the user does not scroll downwards.	When large courses are displayed, the list of groups grows long. This issue could be mitigated by moving the three buttons at the top of the page to somewhere visible by default when the page with group lists load. The floating action button proposed in the Figma prototype could have been useful in this situation.
3.2	Options for filtering is hidden by default and requires a click of the filter icon to be shown	This could be improved by either showing all filtering options, or by providing a better label for the filter icon, but our testers did not really agree on what would be the preferred approach.
3.3	User choices in the application should be remembered when navigating to a new page or view.	The team mainly agrees on this, and it was planned for a later approach. However, there might be situations where the user may expect that some fields are reset, so this feature should be user tested properly.
3.4	Some of the defaults in the date pickers were not very sensible.	This was mainly an issue when the data was quite old
3.5	Constraints on date pickers are not very sensible	As dates in the future and very far past are allowed, the team agrees that constraints could be imposed allowing for a better user experience.

Table 27 - Findings from user tests in Iteration 3

6 Evaluation

One of the five steps in the design and creation research strategy is evaluation, where the worth of the developed web application and deviations from expectations are considered.

As the application developed in this project is part of a user-centered design process with multiple iterations, formative and summative evaluation from all of these iterations are included in this section.

The application has been thoroughly evaluated in all phases of the development process, using three rounds of user tests, a focus group, and a round of expert interviews.

While the user tests have primarily been used to identify usability issues, the focus group and the expert interviews focused on the usefulness of the application within software education courses at NTNU. The expert interviews were successful and confirmed our indications that the application could augment the usefulness of code repository hosting services in the education of software development at NTNU. While there obviously is room for further development of the application and its functionality, due to it being a minimum viable product only developed in three iterations, based on the evaluation we believe that it could have created value if it had been used today.

The various forms of evaluation are described throughout this chapter, and additional details can be found in the various cross-referenced sections.

6.1 Demonstration with Primary Stakeholder

An important aspect of solving a problem using a user-centered process is early involvement of the stakeholders in the project. Therefore, we decided that our main stakeholder should be invited to demonstrations of new functionality whenever a need for it was observed, and feedback was required for the project to go in a direction that would prove useful for the stakeholders. This was especially important after designing the prototype in Figma so that we could avoid spending valuable development time on functionality that would not be useful for the stakeholders.

6.1.1 Findings

Demonstrations with the main stakeholders were conducted throughout the project when the team needed feedback on important decisions for the project. After completing iteration 0, the Figma prototype, we invited our main stakeholder to a demonstration to see if our initial thoughts complied with the needs of stakeholders. After completing iteration 2, we invited the same main stakeholder to see if the functionality so far in the project was aligned with expectations.

At approximately the same time as our main stakeholder was invited to a demonstration after the iteration, the focus group of the process was conducted. The participants of this focus group were four learning assistants, ensuring that opinions from a broader group of stakeholders were taken into account while the development was ongoing. The focus group can be read about in section 6.3.

When the application in our belief had reached the expected level of a minimum viable product (MVP), we had the final expert interviews which included demonstrations of the application as part of the evaluation process. The expert interviews can be read about in section 6.4.

The detailed findings from the demonstrations with the stakeholder can be found in section 5.1.3 and section 5.3.4 about iterations.

6.2 User tests

The purpose of arranging user tests was to identify usability issues at an early stage of the development, such as features not being easy to use, misleading text and labels, and to see if the features were working as intended.

As mentioned in section 4.8.1, the user tests were also part of the evaluation of the functional architectural requirements of the application.

6.2.1 Findings

User tests became an essential part of our user-centered design approach. While a large part of the motivation for conducting user tests was to identify usability issues, it also became a good arena for potential end-users to discuss the functionality and how the application could become the most useful for them.

The detailed findings from the user tests can be found in section 5.2.3, 5.3.3, and 5.4.3.

6.3 Focus Group

A focus group was conducted towards the end of iteration 2. At this point of the project, a focus group can be seen as both a formative and summative form of evaluation. Formative because there is still some room for changes, and summative because a lot of the design and development work already is carried out.

As most of the data gathering methods used in this research have been the use of interviews and user tests, often conducted digitally, we decided that we should also perform a focus group interview while the covid situation allowed it. This can be seen as a way to approach "methodological triangulation" [10, p. 264], described as the act of gathering data through the employment of different data gathering techniques.

Due to many of the interviews being focused on professors and course coordinators, it was decided that the focus group should be populated with learning assistants from various positions in the hierarchy.

The focus group was successful and provided a lot of feedback to the project. As the time set aside for software development in this project was limited, not all ideas and requests could be included in the application, but many of them represent ideas for projects in the future.

Furthermore, the focus group suggested that our application could be useful both for the management of courses, but also be of great use during practical activities in group-based software engineering courses at NTNU.

As planned in the program presented in section 2.3.2.4, the events in the focus group can largely be seen as divided into two main parts, where the main findings and interesting aspects of it are included in section 6.3.1 and section 6.3.2.

6.3.1 Experience Based Discussion

The participants of the focus group discussed their roles, tasks, and pain points from their courses. The process of setting up and customizing courses and their respective course platforms were mentioned several times as time-consuming tasks, and reportedly a lot of this work is delegated from professors to teaching assistants. Some courses may also have their own routines, where teaching staff and sometimes academic staff work out how their course should be managed, and sometimes they create necessary tools and scripts to be able to work with their routines. One of the participants mentioned the use of smart views in Blackboard, where the head learning assistants of the course had created functionality that to some degree corresponded with the teaching assistant dashboard that was part of the Figma prototype of this project. Time-consuming tasks such as adapting to steep learning curves and helping out students with tech-related issues were also mentioned in this part of the focus group, but this was deemed unrelated to the research question and focus of this project.

The participants also mentioned that there is a form of hierarchy within the courses, where teaching assistants are in the lower parts of the hierarchy while scientific staff and professors are in the upper parts of the hierarchy. Between the courses, there seem to be differences in how autonomous the various actors in the hierarchy are allowed to work.

There also seems to be various ways to discover cases where group dynamics and learning outcome may be subpar, and none of the courses being represented in this focus group seem to only rely on stats from Git and coding exercises to determine if the dynamics within a group seem to be working as intended. Instead, a combination of human interaction and data from the deliverables and Git seems to be the preferred way to evaluate group dynamics and the learning outcome. However, there seems to be a potential for giving teaching assistants access to better tools for monitoring student work, as a concern for the participants is that they are not able to detect all cases of subpar group dynamics and learning outcome and that the teaching assistant sometimes feels that they are involved too late in the process to intervene and improve the situation.

6.3.2 Application Related Discussion

Part 2 of the focus group started with a demonstration of the application this project revolves around. Initially, the plan was to separate the demonstration and the discussion, but the participants were eager to discuss the functionality included in the demonstration and to propose added functionality and design twists, so part 2 became a session with continuous feedback.

The participants seemed to like the application as we presented it to them, and some of them mentioned that it absolutely would be of interest to use it in their courses. This can be confirmed by quotes they expressed during part 2 of the focus group. Participant 1 was quoted that "With minor added functionality, this view could potentially save me 40 hours of work each year". Participant 2 was also quoted that "After seeing the potential of this application, I dream about making Git an important part of my course so that I can easily follow up on my students and help them with their issues."

The participants also expressed some dissatisfaction on how Blackboard caters to their needs as teaching assistants, and a clear point of frustration amongst the focus group participants was related to how Blackboard handles situations where students join or leave the course later than the start date of the course. According to focus group participants, these situations make it hard to organize student groups and complicate the assignments of teaching assistants to both individual students and student groups.

6.3.3 Takeaways for This Project

Based on the information gathered from the focus group, our hypothesis that the project may be able to empower teaching staff in software development courses, is strengthened.

The focus group was arranged due to a number of reasons, one being that the team seeked knowledge about how teaching assistants supervised student groups in software development courses. After the focus group, we understand that the teaching assistants focus is on group dynamics and learning outcome and uses a combination of metrics from the code repositories together with human interaction to understand how well the student groups are working. Based on statements from the focus group, we understood that counting commits and issues in a code repository does not tell the full truth about the work distribution within a student group. However, we believe that by providing metrics about the code repositories in student groups to the teaching assistants, we empower them to better understand their groups, and commits, issues, and merge requests seem to be the most interesting metrics for them. This is important to know before finishing the application this project evolves around, as the most interesting metrics is likely to be the most used data, and it would therefore be helpful for the users if those metrics are in focus in the application by default.

6.3.4 Takeaways for Future Work

Some of the things that were discussed during the focus group session have great potential as possibilities for future work after this master thesis. This includes mentions of a web portal where students can give feedback on the perceived group dynamic, so that teaching assistants can become aware of potential issues as early as possible. This functionality could also potentially be seen in combination with student facing web portals related to this project, where metrics, stats, and feedback from the course staff may be available. The implementation of this student facing portal was not prioritized in this project, but feedback suggests that there may be a potential for such a project in the future.

Depending on the final implementation of dashboards for student groups and teaching assistants in this project, a potential task for future work could have been automatic creation of smart views in Blackboard, but at the time of writing in April 2021, the Blackboard API seems to lack functionality to handle this task. The smart views in Blackboard are used to allocate groups that a learning assistant is responsible for and give the learning assistant a view where he could give feedback to his groups.

During the focus group the potential for adapting the application in this project to better support individual programming assignments or deliverables was also mentioned. As of now, the application is primarily designed for software development courses with groups, but it should not be too hard to adapt most of the functionality to solve needs related to individual deliverables.

The possibility of automated grading was also mentioned during the focus group. However, we were aware that other teams were working specifically with this issue, so it was not considered of particular interest for this project. An attempt to exchange experiences and information with one of the teams working on automated grading was made half-way through this project, but we then learned that the automated grading project had come to a halt.

6.4 Final Expert Interviews

Towards the end of the user-centered design process, the evaluation pivoted from formative evaluation towards summative evaluation.

As we started the design and creation process interviewing course coordinators about their needs and wishes, we decided the finished application should also be evaluated by people in this group.

To get varied feedback, we decided that some of the original participants in the project should be invited to the evaluation, while some new participants also should be invited, to look at the project from a new and perhaps more general perspective. In the expert interviews, one of the course coordinators from the initial interviews were invited, while two new participants with experience from course management in software development courses with group-based practical activities also were invited.

The purpose of doing summative evaluation is to see if the project brings any value to possible users of the application, and to identify any contributions to the state of the art, and we decided that the best way to do this under the current circumstances at the time of writing, is through expert interviews. By doing interviews with experts that are potential end-users as well, we were able to identify if they also saw potential for use of the application in their courses. As mentioned in section 4.8.2, the expert interviews were also aimed at evaluating the non-functional architectural requirements.

The findings of the final expert interviews can be summarized as a confirmation that the minimum viable product (MVP) of the application evolves around can be considered useful at the time of evaluation. The team also confirmed that the application could be useful related to practical activities in software development courses at NTNU, and hopefully in other relatable contexts.

The single group dashboard proved useful, and all participants thought the charts were innovative and useful, but the group list seems to have unrealized potential. To elaborate on the room for improvement based on the final expert interviews the experts had a couple of suggestions, most of them related to a desire of added possibilities for customization of the web application, something the team had planned for later iterations.

Increasing the level of customization could be done in several ways, but the among the ones that were discussed was that the dashboard could be made customizable based on modules that each user could enable or disable themselves. This could also include adding or removing specific data from tables and lists, and to highlight certain data that are of particular interest for the individual user. One of the experts also suggested that a separate settings page could be created, where it could be possible to edit the appearance and the page views of the web application resulting in a very flexible user interface. However, the team is concerned that such a level of customization being possible also could make it possible to create confusing user interfaces, and that such options would require a lot of resources to implement and maintain.

6.4.1 Expert interview 1

The first interview was held with a course coordinator at IDI, with long experience from group-based learning activities within software engineering related education. He has also contributed to the publishing of papers within project-based learning in IT education and was therefore considered a very good expert to interview in the evaluation for this project.

As planned, the interview started with some introduction questions about the background of the expert, before we had a brief introduction of our project. After the project had been introduced, we moved on to the demonstration where we discussed all major features directly after they were presented. This made sure that the participant could remember what he had seen and ask the presenter to clarify any uncertainness about the functionality.

Participant 1 expressed that the project could prove useful and was very clear that the project ideally should be open-sourced and developed into a production-ready application so that it could be used in the education at NTNU.

The participant expressed disappointment in how Blackboard handles the creation of groups, and expressed that this functionality could absolutely be useful.

Regarding the list of groups, the participant mentioned that while the data could be useful, he had some ideas about how it could be better displayed.

6.4.2 Expert Interview 2

The second interview was held with a PhD candidate at IDI, who has been one of the learning assistants in several courses at NTNU, including courses containing project-based software engineering education in groups. He also has valuable experience from design and implementation of web and mobile applications, and was therefore considered a good expert for this project.

The interview was conducted as planned, starting with questions about the background of the expert, before moving on to a demonstration of the application.

The participant stated towards the end of the interview that he liked to look into the code repositories of his students, something he had previously done manually and unstructured, and that this application could structure and simplify this process for him. He also told us that the functionality for creating groups could be of use, but that the course contains very different types of projects, and that the functionality for creating groups in the code repository hosting services was not applicable for that course.

Towards the end, the participant stated that the project could be useful in the education of software engineering at NTNU.

6.4.3 Expert Interview 3

The third interview was conducted with one of the participants from the initial interviews of the project, with background as a course coordinator in one of the software development courses at NTNU. As the background of the participant already had been addressed earlier in the study, no questions were asked before the commencement of application prototype demonstration.

During the demonstration, we asked the participant questions about the usability of each feature, while concluding the expert interviews with general questions about the perceived usefulness of the application prototype in software development courses with practical activities and possible deviations from expectations.

When presenting the possible filters for the group list, the participant stated that he saw a potential for query-based filtering. This could include both time-based filtering and threshold based filtering on specific metrics.

The participant also proposed looking at filtering as something related to generic business intelligence, where integrating or allowing export to a software such as Tableau could provide a lot of functionality without requiring a lot of implementation work.

When showcasing the streamgraphs visualizing commit data, the expert stated that it could be a bit more nuanced if the size of the commits were visualized in some way. There we immediately proposed including a streamgraph based on the dates where additions of code lines were added. The expert was positive to this proposal, but was not sure if additions were the only appropriate metric, and proposed that maybe deletions also should be included. We also discussed introducing weighted additions, where for example the addition of code lines could be considered as more work than the addition of text lines in the documentation. Introducing a blackout list of files that don't count was also considered, but the selection of files could be quite complicated, and implementation of functionality where course staff could select their own files was not possible with regards to the time constraints of this project. In any way, the implementation of something like this should probably be based on additional research not within the scope of this project.

Asked on the deviations from the expectations since he first heard about the project he concluded:

- My impression is that you have surprised me in a positive way. I have no idea what inner qualities that may be included in your software, but you have both understood the requirements for such a project, and the application seems both well elaborated and relatively rich in functionality. I am sure that this application will be useful this upcoming autumn as it is, even though there will always be requests for added functionality.
- This is a very complicated field, but I think you have done it correctly from the start. This is the first iteration, even though you have had several technical sprints, but this is the first iteration where you have validated your research with more than one course coordinator. As you do not have too many choices on behalf of your users, I think you have started in a sensible way.

During the demonstration, the expert was also impressed by the performance of the application, stating the following:

- "Are you fetching [Git data] in real time or have this been downloaded in advance? I am surprised it loads so fast."

This is related to the work done in iteration 2, where we received feedback on the slowness of the application. As a response caching was implemented, and API calls were streamlined to reduce loading time of data in the application, where this statement could be a signal that the measurements were successful.

“Do you think that this application could be useful for managing courses where both Blackboard and GitLab are in use, and if so, how?”

- I think you have been thinking very reasonably when selecting to go for a solution with a single source of truth. It is very nice that you are not forced into continuing using the app if you start using it. I agree with the structure you have implemented for creating groups and repositories in Git.
- I think it looks very promising. I have been thinking a lot on how promising software becomes something more than promising, meaning that they are put into use and maintained. In that regard, you have given us a new guinea pig that we can experiment with, because this cannot be dependent on you, it must depend on what you deliver to us in a good way.

“To what degree do you think that this application can help you with visualizing student work during practical activities in software education courses, for example in your course?

- This is related to both the general approach to visualization of student work and the exact metrics selected for visualization. I think that you have made good choices both with the general structure and the charts, both the pie charts and the compare functionality. However, I think there is still work ahead, to find the good indicators and metrics that give us the best possibilities to guide students. This is not criticism towards your project, because you are working with a set of requirements, and this is an iterative process. We will probably see very quickly when using the application in real courses, that we need to change some of the metrics used in the application, where it is important that the application is maintainable and can adapt to changing requirements in the courses.

“Do you think that this application can do a good job at visualizing the work of students in practical activities in software development courses?

- I think it would be a bit more nuanced if the size of commits were represented.

“To conclude, do you think that this application overall could augment the usefulness of using code repository hosting platforms in relation with practical activities in software development courses?”

- Well, this is a tricky question. On one side, it is a given that if this tool is available, course coordinators may push their students into using a fixed set of technologies, which is not a problem for the course coordinator, but
- There is very much ahead of this related to this project, both practically and in terms of research. My perspective is both on my course, but also on the education throughout the Computer Science studies at NTNU, so I'm also looking at how courses and exam situations could be rigged for better gathering of quality metrics for example from GitLab. This could for example be detecting bad coding practices, which would help us give improved feedback to the students.

The expert later made a point that maybe the students should be able to view their own data, either through the application or by using a plug-in for the VS Code editor many of the students use, which is something we have considered, but decided not to do in this project. This was partly due to time constraints, but also because of the challenges it would have presented, both from a design and implementation perspective, but also from a learning perspective.

The expert also pointed out that this software should be called an MVP and not a prototype if the point is that the application should be developed further, and agreed that the different projects at the institute could be better coordinated so that possibilities for better utilizing synergies could be improved.

7 Results and Discussion

This section presents the summarized results of this project, together with a critical discussion on how the results were achieved, what can be considered the project's contribution, and reflection and debate on how the project and the application could have been improved.

The results of the project can be seen as the achievement of the goals defined throughout the project, as well as how well the two research questions have been answered. The evaluation has shown that the application designed and developed as part of the project seems useful for course staff at NTNU, indicating that it also could be useful for similar situations in other contexts.

The research questions of the study have also been answered through the web application, all the way from the process of designing and implementing it to the expert evaluation. The mentioned evaluation has, together with other observations, made the team fairly confident that the web application both improves the process of managing software development courses where both learning management systems and Git repository services are involved, as stated in RQ1, while also helping course staff better supervise and guide their students and student groups by utilizing data in Git repositories.

The section also includes a description of the contribution made by this project, and critical reflection on how the project could have been improved or done in a different manner, related to a number of topics.

7.1 Results

This section includes the project’s results related to achieving goals set in the early phases of the project and the research questions the project has evolved around. There are reasons why the results of this project may not be entirely valid and reliable, and these are thoroughly discussed in section 7.3.

7.1.1 Achievement of Goals

Focusing on the quality attributes usability, and maintainability, this project has evolved into the design and implementation of a modern web application that helps course coordinators and learning assistants in software education courses at NTNU by simplifying some aspects of the course management and by providing a tool that helps the course staff follow up on their students, which allows them to give better feedback and can facilitate a better learning environment.

We have achieved insight with the work and development of the web application on how to highlight the needs of course management in courses where Git is used. Based on the feedback we have received through the evaluation of the project, we have strong indications that the minimum viable product of the application designed and implemented in this project will be helpful in software education courses at NTNU and meet some of their needs to provide a better learning experience in their courses.

7.1.2 Research Question 1

Research question 1 was "How can the process of managing software development courses where both learning management systems and Git repository services are involved be improved?"

The team have observed strong indications that the project has succeeded at creating an application that improves the process of managing software development courses where both learning management systems such as Blackboard and Git repository hosting services such as GitLab is used. This observation is based on the answers from the final expert interviews, feedback from the focus group, and impressions from the user tests and demonstrations. The sentiment is based on our impression that the functionality in the application developed improves the process, which the sayings from the experts evaluating the application in our belief indicate.

The functionality in our application that contributes to improving the process of managing software development courses include:

- The guidance of creating a structure for courses on a Git hosting platform
- The possibility to create students group extracted from Blackboard on GitLab automatically with appropriate roles and permissions
- The functionality to see the state of students group on Blackboard and GitLab and compare the two states.

7.1.3 Research Question 2

Research question 2 was «How can course staff in software development courses be helped to better supervise and guide their students and student groups by utilizing data in Git repositories?”

Based on the answers from the final expert interviews, feedback from the focus group, and impressions from user tests and demonstrations, we are fairly sure that we have succeeded at creating a tool that facilitates improved feedback from the course staff to students. Even though the metrics used in the application at the time of writing may not be perfect, their presence still seems to be an improvement of the current situation. The tool should not be used for grading by itself, but provides the course coordinators and learning assistants with another angle to look at how the students work, which could improve the feedback given to students by course staff, and as such, could increase the learning outcome of participating in practical activities in software development courses.

The functionality in the application that help course staff with the process of supervising and guiding students include:

- A new dashboard with information about all of your student groups, which improves the feedback you can give and the opportunity for course management to identify groups that need a bit of extra help.
- A new dashboard with information about a single student group, which improves the feedback you can give to the group during the course and at the end.

7.2 Contribution of the Project

Research studies are often focused on contributing to the field and context they are taking place in. This project does, in large part, take place within the area of improving the teaching of software development in higher education.

The contribution of this project could primarily be split into two parts, the findings of the insight work that have been a large part of the project and the application that has been designed and developed using the conclusions of the insight work.

Details of the insights gathered while working on this project are included in this thesis and will be published through NTNU Open and hopefully is of use for other researchers exploring related challenges and problems.

The application that has been designed and developed in this project will be open sourced using the license "GNU General Public License v3.0 or later" [42]. This will ensure that whoever wants to use or contribute to the work done in this project will be able to do so while also ensuring that all improvements will be available for others who may be able to make use of those improvements, rules we think are fair and in line with the NTNU slogan "Knowledge for a better world" [43]. Details related to the open-sourcing of the project can be found in section 7.5.2.1.

The project has also broken technical ground, by, according to an engineer at NTNU IT, being the first third-party outside NTNU IT to develop software that integrated with the NTNU instance of Blackboard through the available REST API.

The source-code and documentation of the source-code can be found on GitHub [44].

7.3 Validity of the Results

Validity is an important concern in most research projects, and while a lot of the choices done throughout the project have been discussed previously in the thesis, our main concerns for the validity of the project and its findings is discussed in this section.

The term validity is often divided into two main parts, namely internal validity and external validity.

Good internal validity is described by Oates to be present in an experiment "…if the measurements you obtain are indeed due to your manipulations of the independent variable, and not to any other factors." [9, p. 131]

One of the main threats identified for the internal validity in this project could be that the needs or behavior of the participants in the project changed throughout the project, for example if they participate in multiple stages of the project. To counter this validity threat, the team has attempted to involve new possible stakeholders at most stages of the process, while still involving some of the original stakeholders. Another threat to the internal validity of the study is what Oates describes as "reactivity and experimenter effects", briefly summarized that people change their behavior as a reaction to being tested or interviewed. This could potentially impact the observations or statements gathered in the project, but as the participants all have an academic background, there is a belief that the participants are aware of the issue themselves, and attempt not to be influenced by these effects, avoiding the potential bias that can be occur from such effects.

Good external validity is described by Oates to be present in an experiment "…if your results are not unique to a particular set of circumstances but are generalizable, that is, the same results can be predicted for subsequent occasions and in other situations." [9, p. 132]

A very important threat for the external validity of this study is that the circumstances at other institutions than NTNU most likely will be different than those present at NTNU. However, this validity threat would be very different to mitigate without having a much larger study. Oates mentions that relying too much on special types of participants in a study could be a threat for the external validity of a project, but the team has decided this risk is almost unavoidable in this project, as the potential user group is a very narrow and small group of persons. Related to this, is the potential threat of inviting non-representative participants to the study, but the possible sample group are again very limited, as further described in section 7.3.2.

7.3.1 Evaluation in a Real-world Scenario

This master thesis project was conducted over a little less than a year. Within this time span, a lot of insight work was needed to be carried out before attempting to solve the selected problems of the project. Related to this is that courses at NTNU either commence in August and finish in December or commences during January and typically end in May. This meant that since our project started in September, designing and developing a valuable version of what ended up being our application was needed to be completed before January, for it to be tested in a real-world scenario in the lifespan of this study. This was not possible, and the application has, because of those timing issues, not been tested in a real-world scenario with the data of an ongoing course. This may have an impact on the validity and reliability of the results in the study, but the team have been testing and evaluating the web application with people that are frequently involved in real software development courses in higher education, removing what is believed to be some of the concerns related to this issue.

7.3.2 Selection of Participants

This project evolves around research questions aimed at course staff in software development courses where GitLab and Blackboard are used to manage practical activities, forming a limited pool of possible participants in the project. Even with few potential participants, we have strived to keep the representation of participants as distributed as possible within the sample group. Time and resources were natural constraints while identifying possible participants for the project. The covid situation also made it difficult to get in touch and schedule interviews and other events with possible participants, related to work-from-home policies and travel restrictions.

This project evolves around research questions aimed at course staff in software development courses where GitLab and Blackboard are used to manage practical activities, forming a limited pool of possible participants in the project. This, together with challenges related to the covid situation during this project, limited potential participants.

The number of participants has also been discussed in a study by Jakob Nielsen [45]. The findings indicate that testing as often as possible with a small number of participants is quite efficient when trying to identify usability issues. This interpretation has also been supported in a subsequent article published by Nielsen Norman Group, stating that "Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford." [46]

7.3.3 Consequenses of Qualitative Approach

Our research has been conducted with a qualitative approach that is subjective and could be influenced by many factors. Additional methodological triangulation could have improved the situation by increasing the reliability of the results, and this could have been done by conducting a survey based on the questions presented in the quantitative approach System Usability Score (SUS) [12], where the participants evaluate the usability of an artifact.

7.3.4 Digital Expert Interviews

A large part of our final expert interviews was the demonstration. Due to our NSD application not including us reporting the need for video recording of participants in our study, and the need to record the sound of the expert interviews to avoid taking notes that would impact the flow of the expert interviews, we were not able to see the facial expressions of our experts while demonstrating our application. This might have reduced the accuracy of our interpretation of how the experts evaluated our work, but we are fairly sure that the experts could be trusted with their spoken feedback, especially when taking their way of speaking and expressing themselves into account. Furthermore, the experts should not have any motivation to influence our results positively or negatively, apart from the natural human reflexes of being friendly and helpful, which the experts participating in our study evidently have.

7.3.5 Having a Combined Supervisor and Primary Stakeholder

Our supervisor was involved in the project both as a traditional master thesis project supervisor and in a stakeholder-like role as he would be one of the potential end-users of the resulting artifact the project would produce.

Meetings were arranged every week in hectic periods of the project, while semi-weekly meetings were planned in not-so-hectic periods of the project. The meetings sessions were primarily arranged using Microsoft Teams, primarily due to work-from-home policies being enforced over long extended periods throughout the length of the project. In these meetings, our supervisor also introduced us to interesting research and similar solutions that were of interest to this projectOur supervisor also introduced us to interesting research and similar solutions that were of interest to this project in these meetings.

The supervisor also helped us point out possible improvements in the thesis the last month before the delivery date.

Independence is an integral part of working with a research project, and it was important to establish when the supervisor acted as a supervisor and when he acted as a customer. This was done by attempting to have separate meetings for supervisor discussions and stakeholder discussions. When discussing the project digitally, we also tried to keep supervisor discussions and stakeholder discussions in individual chat posts.

To avoid our supervisor deciding too much on the project, a relatively large number of other possible end-users were invited to the project. This included both people from the same course as our supervisor was coordinating and course staff from other courses. This enabled us to confirm that the requirements and needs were valid for software engineering courses in general.

7.3.6 Brief Literature Review

Conducting a thorough literature review as part of this project would have required large amounts of the time and resources available in the project. The review was therefore decided to primarily focus on the State of the Art related to the design and creation the project evolves around, while also including a brief literature review connected to our research questions. While there is a chance that this decision leads to the researchers missing out on literature that could have impacted the project and the decisions made throughout, we still believe this was the right decision as it allowed us to explore more of the possibilities related to the design and creation phase of the project. It is also worth noting that literature was consulted throughout the project if it could be seen as useful.

7.3.7 Timing of Focus Group

Focus group interviews are often performed early in the product lifecycle, so ideally, this should probably have taken part before the beginning of product development. On the other hand, the product after two or three iterations could still be considered as early in the lifecycle, and the focus group interview as such took place in what could be regarded as an early stage of the product development lifecycle.

An essential benefit of conducting the focus group after the first two iterations were that it was possible to split the session into multiple sections, where we first could gather the initial thoughts from the participants while continuing to gathering their thoughts about the application after two iterations, and lastly gather the ideas of the panel regarding the future work of the project.

7.4 Collaboration with Other Projects

While working on our project and talking to different stakeholders in the process, the team was made aware about two other projects that could have been seen as related to this project. Those are mentioned in this section, together with an assessment of how some kind of collaboration could have taken place.

7.4.1 Group Activity Monitoring

One of the experts interviewed during the evaluation phase of the project already collaborated with a master student working in a field very related to this project.

We think that if we were aware of the project, we could have collaborated and facilitated more significant contributions to the field in both projects. However, we were informed of the project too late to have any meaningful collaboration.

7.4.2 Project Related to Automated Grading

In contrast to the other project mentioned, we were informed about the existence of another project at IDI related to automated grading, a topic that was discussed when deciding the scope of this project.

However, the student working with this project abandoned the project after the initial research phase, and the team did not succeed at contacting the student to exchange experiences and knowledge, so there was unfortunately no room for further collaboration with project.

7.5 Application Usage After This Project

Despite the application having been designed and developed to possibly be used outside the context it was created within, there is a possibility that the application might face challenges if used or developed outside of the original context at NTNU. There are certain things to consider in these situations, and some of them are discussed in this section.

7.5.1 At NTNU

During the final expert interview with expert 3, the expert raised a general concern that software developed by students need to be used and maintained before it can be considered helpful in authentic contexts.

In this project, maintainability has been selected as the secondary quality attribute we focused on, only following usability, as described in chapter 4.

7.5.1.1 Hosting and DevOps

Because we went for creating a web application, somebody must set up a new environment it can be hosted in. First, it must be decided who will host it, NTNU IT, IDI Technical Staff, or the course coordinator wanting to use the application. The course coordinator is probably happy to do it, but the IT people may see it as a hassle. The best solution we could have done was to ask the IT people what environment and technological stack they already were using and make the same choice, but that may not have been the best for our application. One team member works as a technical staff at IDI and has experience with their technology environment, knowing that the staff at IDI are familiar with the selected technological stack for the application. And to lower the hassle even more, the team focused on making good documentation of the application and on how the application could be set up and configured.

7.5.1.2 Further Development

We hope that others show interest in the project and are willing to use some of their time to develop the application further. During the evaluation interviews there was an interest in making a paper and possibly creating a foundation platform with this project as a possible starting point for later master thesis projects. By selecting maintainability as our second quality attribute and the choices we made, we are sure that we have created a possible project for others to develop further.

The project will be hosted at the GitLab instance that IDI is running and made public for everyone to access, so everyone willing and interested can contribute to the project.

7.5.1.3 Elements Needed Before Application is Production Ready

Before the application could be considered production-ready, a few things need to be considered. The most important of those is to get the application approved by NTNU IT and get credentials for the production instance of Blackboard used at NTNU. It has only been tested with the test instance, with NTNU IT having been involved. However, unknown differences and changes may be discovered when using the production instance, so testing it on the production instance before production is encouraged.

Another change needed to be done is to implement a system for roles in the application. The application right now allows everybody that logins have access to everything, and you should restrict or reduce access for students and others that do not participate in the course.

7.5.2 Outside NTNU

7.5.2.1 Open sourcing

The use of the application in other contexts than in the software development courses IT1901 and TDT4140 at NTNU depends mainly on usefulness, interest, and availability. While factors such as usefulness and interest outside of the context the project was created within are outside the scope of this project, the availability of the application for future use can be influenced.

An important detail regarding the availability of software is the licensing terms. Some software uses commercial licenses, requiring payment for usage, while the opposite would be using other licenses making the software in question considered open source.

While Hars and Ou suggest multiple factors for making software projects open-source, naming both internal and external motivation sources [47], none of them applies precisely to the primary motivation of open-sourcing this project. We believe that if this project can be useful in the education of software development at NTNU, it can also be useful in other contexts.

With this belief, making the application available for others would require one of mainly three strategies: open sourcing, the second being commercialization, and the third would be not sharing the software, only the information needed about it. While open sourcing does not require anything other than a specified license document, it is considered good practice to document the software properly so that other software developers can easily contribute and make use of the software made available. On the other hand, commercialization brings a whole new set of challenges and was not considered in this project. The third option, not making the software available for others, does not require any work but also reduces the project's contribution outside of NTNU, not being in line with the NTNU slogan "Knowledge for a better world" [43].

While open sourcing does not necessarily increase the quality attributes within the NTNU context it may increase them for general use. More eyes and settings on the project will improve the project's usability. If the project is open sourced, having high maintainability is essential to get new developers on board.

The project was designed and developed with the thought that it shall be possible to use the application in other settings than NTNU. It should be possible to extend and modify parts of the project to meet the requirements of different settings. And we, as the initial developers, see that the best way for this project to improve after the thesis is done is to publish the code and this thesis for others to work on later and come with new ideas, experiences, and requirements.

7.5.2.2 Technical challenges

The most obvious challenge with importing this application to a context outside NTNU is the technical issues that can arise from using other technology environments than the environment at NTNU. While the application has been made tailored to NTNU requirements, others may have other technical requirements, meaning that the application should be flexible enough that adapting to new requirements is a possible task.

Our context was Blackboard, GitLab, and Dataporten, but others may want to use Moodle instead of Blackboard. This is why we chose to create models of the data we use from Blackboard so others can look at the models and map them to their choice of LMS. They can then look at the isolated code for Blackboard and adapt it for their LMS. The same applies to GitLab and Dataporten. A functionality in the future may be a modular system that accepts modules of code for Blackboard and other LMS, meaning that it would be possible to select what module that is suitable for a specific context while configuring the application.

Another hurdle is the hosting and deployment of the application. The application is developed with a certain DevOps pipeline in mind, and if other deployment configuration is required, some extra work could be required. Our solution to lessen this possible work is to provide reasonable documentation and make it easy for others to host without additional research. It is also possible and easy to create a configuration for Docker [48]. Docker is a service to deliver software in isolated containers, making it possible for others to host the application without thinking about the environment the application needs.

7.5.3 Convertion to TypeScript

The team decided to use plain JavaScript for this project as it was believed that the overhead of using TypeScript would have been greater than the advantages. In hindsight the project should have been written in TypeScript from the start, a reason being that the team underestimated how complex the project would become. Now, as things are, you must make additional documentation of the models, types and the data structure, but with TypeScript it would have been self-documenting.

All the software used in the project was purposely chosen to support TypeScript if it was later decided to convert over, because of this it should be possible and not require much work.

7.5.4 Functionality to be Finished

In some software development projects, all functionality is not necessarily completed according to the original progress plan. In this project, the pages for viewing all students, a student, and which groups a learning assistant is responsible for, was not finished and should be developed, but the pages were not considered critical to implement before using the application in production. The pages implemented was seen as more important because of feedback the team got, and this was the reason why the team decided not to prioritize to finish those three pages.

Prototypes of the pages can be found in Appendix B, and a starting point for them is implemented in the application.

8 Conclusion and Further Work

This chapter contains both the conclusion of our work, together with some thoughts and suggestions about further work that can be derived from this master thesis project.

The first part summarizes how it has been to work on this project and study, while presenting some of the challenges and how the team has coped with them throughout the study.

The second part presents some ideas for further work based on this master thesis project, alongside some thoughts from the team on the potential of the ideas for future research.

8.1 Conclusion

The web application that became the result of this master thesis project has not yet been put into production and providing a final conclusion is therefore challenging. However, an evaluation of the web application designed and implemented as a large part of the study is included in chapter 6, and we see strong signals that our work has been valuable, and provides a good foundation for improving the usefulness of Git repository hosting services for practical activities related to software development courses.

While the covid situation has impacted the possibilities of the project throughout its lifespan, the team is happy with the results and the achievement of its goals. Of course, as discussed in chapter 7, there will always be room for improvement and other ways to do things, but overall, the team is satisfied with the project and the results of the study. This is also something which have been indicated by other stakeholders in the project, such as the supervisor of the study and interview participants, as described in chapter 6.

The web application that has become the center of the project has been designed and developed using a user-centered design approach, and an agile development process. During these processes the team has used a broad variation of techniques and methodologies learned through their years at NTNU, while also learning a lot working on the project and the study. The web application itself is based on very modern technology, while also having usability and maintainability in focus as explained in the quality attributes that was focused on in chapter 4, and the team is very thankful for having the opportunity to conduct a master thesis study that hopefully has a real impact on the future of the teaching of software development, either on NTNU or maybe in another context.

8.2 Further Research

The web application has not been used in a real-world scenario because of covid-related delays, so there should be potential for both a further evaluation of the usefulness, and for further development of the web application and the idea. The web application may also be expanded to cover a new set of requirements.

All of the suggestions in the section should be able to connect to the motivation of this project, and can maybe be seen as some kind of continuation of the project, while some of the suggestions would require a separate project to be initiated.

8.2.1 Automatic Grading

Automated grading has been a field of interest for several of the people we have interviewed throughout this project. Due to the complex nature of the problem and the need for customization of grading, we decided not to pursue this branch of the problem.

8.2.2 Automatic Group Creation

Many of the people we interviewed mentioned that group creation could be automated. Group dynamics could potentially have a significant impact on how students learn. Based on our interviews, we think there is a demand for automatic group creation.

8.2.3 Integration with Other Applications

The application described in this thesis is based on the NTNU instance of Blackboard Learn and the usage of GitLab Community Edition. However, many of the other institutions in Norwegian higher education use the learning management system Canvas after an agreement was signed in 2016 [49], requiring a different set of integrations. As of March 2021, Canvas offered APIs based on REST and GraphQL technology, so it should be possible to integrate the web application with the Canvas software. Other institutions may also have a different set of requirements for a similar solution.

8.2.4 Data Visualization

The application fetches a lot of data and only visualizes a selection of it. As the technical platform is very modern and flexible, there may be a potential for expanding the use of the aggregated data and display it in new and creative ways.

8.2.5 Learning Assistant Management

During the focus group, one of the head learning assistants mentioned that some of his time was spent following up on student learning assistants that had not finished their tasks, delaying other processes related to their task.

A possible solution for this challenge could be adding data to the dashboard for course coordinators and learning assistants, so that the task of getting an overview of tasks needed to be done in the course becomes easier.

8.2.6 Deploy Student Applications

Several of the people interviewed have expressed that they wanted to run software developed by students as part of practical activities in software development courses too see how it is functioning and not only view the Git stats for a group. In the current situation, LA’s must now clone the project and run them on their own devices and systems. This is a hassle and can be dangerous if some students accidentally or on purpose make malicious code.

A solution mentioned by one of the interviews was to integrate towards GitPod as GitPod provides automated environments to run code in. The group list page in our application can have a list of assignments or projects in a group and a link that takes you to GitPod and starts up an environment for the application.

There can also be looked into using other services than GitPod or implementing another system for NTNU for running code securely, but this is a big task and should probably be a separate project to look into. The team thinks that a project for this could have a large potential, but also require a lot of work with the technology we are aware of at the time of writing.

References

[1] NTNU, "ntnu.edu - TDT4140," [Online]. Available: https://www.ntnu.edu/studies/courses/TDT4140. [Accessed 4 May 2021].

[2] NTNU, "ntnu.edu - IT1901," [Online]. Available: https://www.ntnu.edu/studies/courses/IT1901. [Accessed 4 May 2021].

[3] M. Lorås, "From teaching assistants to learning assistants – lessons learned from learning assistant training at Excited," in *Vol. 4 No. 1 (2020): Proceedings fra Læringsfestivalen 2020*, Trondheim, 2020.

[4] "Blackboard Learn," [Online]. Available: https://www.blackboard.com/teaching-learning/learning-management/blackboard-learn. [Accessed 4 June 2021].

[5] M.-A. Letnes, A. E. Killerud and I. G. Kalfoss, "Studenter og underviseres forventning og erfaring med bruk av læringsplattformen Blackboard ved NTNU," *Nordisk Tidsskrift for Utdanning Og Praksis, Vol. 15 Nr. 1,* 2021.

[6] NTNU, "innsida.ntnu.no - E-læring," [Online]. Available: https://innsida.ntnu.no/wiki/-/wiki/Norsk/E-l%C3%A6ring. [Accessed 4 May 2021].

[7] "Git," [Online]. Available: https://git-scm.com/. [Accessed 4 June 2021].

[8] "GitLab," [Online]. Available: https://about.gitlab.com/. [Accessed 4 June 2021].

[9] B. J. Oates, Researching Information Systems and Computing, London: SAGE Publications, 2006.

[10] H. Sharp, J. Preece and Y. Rogers, Interaction Design: Beyond Human-Computer Interaction, Fifth Edition, Indianapolis, IN, USA: John Wiley & Sons, Inc., 2019.

[11] J. Brooke, "SUS - A quick and dirty usability scale," in *Usability Evaluation in Industry*, London, UK, Taylor & Francis Ltd, 1996, pp. 189-194.

[12] J. R. Lewis, "The System Usability Scale: Past, Present, and Future," *International Journal of Human–Computer Interaction,* no. 34:7, pp. 577-590, 2018.

[13] International Organization for Standardization, "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems," 2019. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en. [Accessed 8 April 2021].

[14] International Organization for Standardization, "Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts," 2018. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en. [Accessed 8 April 2021].

[15] W3C, "Web Content Accessibility Guidelines (WCAG) 2.1," 11 June 2018. [Online]. Available: https://www.w3.org/TR/WCAG21/. [Accessed 18 May 2021].

[16] R. Molich and J. Nielsen, "Improving a human-computer dialogue," *Communications of the ACM,* pp. 338-348, March 1990.

[17] D. Norman, The Design of Everyday Things: Revised and Expanded Edition, Basic Books, 2013.

[18] "Agile software development," [Online]. Available: https://en.wikipedia.org/wiki/Agile_software_development. [Accessed 4 June 2021].

[19] H. Kniberg, Scrum and XP from the Trenches, C4Media, 2015.

[20] K. Beck, Extreme Programming Explained, Embrace Change, Addison-Wesley, 1999.

[21] NTNU, "Data Storage Guide," [Online]. Available: https://innsida.ntnu.no/wiki/-/wiki/English/Data+storage+guide. [Accessed 8 September 2020].

[22] S. Larsén and R. Glassey, "RepoBee: Developing Tool Support for Courses using Git/GitHub," in *ITiCSE '19: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 2019.

[23] GitHub, Inc., [Online]. Available: https://docs.github.com/en/education/manage-coursework-with-github-classroom. [Accessed 25 May 2021].

[24] A. Waldenberg, "GitHub.com," [Online]. Available: https://github.com/ejwa/gitinspector. [Accessed 25 May 2021].

[25] J. Brekke, T. Guilfoyle, V. Huarota, Y. Berezkin and Z. Altahat, "ieeexplore.ieee.org," [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9208267&tag=1. [Accessed 25 May 2021].

[26] J. Chines, A. Iovanna, T. Patikom and A. Venkatesh. [Online]. Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-032215-091925/unrestricted/MQP_Formatted-GFP_Final_Final.pdf. [Accessed 25 May 2021].

[27] G. Derval, A. Gego, P. Reinbold, B. Frantzen and P. Van Roy, "Automatic grading of programming exercises in a," 2015.

[28] M. L. Hetland, "algdat.idi.ntnu.no," [Online]. Available: https://algdat.idi.ntnu.no/info/2018/11/09/ny-ovingstjener.html. [Accessed 25 May 2021].

[29] M. A. Angulo and O. Aktunc, "repositories.lib.utexas.edu," 4 April 2018. [Online]. Available: https://repositories.lib.utexas.edu/bitstream/handle/2152/79922/using-github-as-a-teaching-tool-for-programming-courses.pdf?sequence=2&isAllowed=y. [Accessed 25 May 2021].

[30] J. Kelleher, "ieeexplore.ieee.org," [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6916568&tag=1. [Accessed 25 May 2021].

[31] G. Sprint and J. Conci, "Mining GitHub Classroom Commit Behavior in Elective and Introductory Computer Science

Courses∗," *The Journal of Computing Sciences in Colleges,* pp. 76-84, 2019.

[32] R. M. Parizi, P. Spoletini and A. Singh, "ieeexplore.ieee.org," [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8658983. [Accessed 25 May 2021].

[33] L. Bass, P. Clements and R. Kazman, Software Architecture in Practice, 3rd Edition, Addison-Wesley.

[34] M. R. Barrick and M. K. Mount, "The Big Five Personality Dimensions and Job Performance: a meta-analysis," *Personnel Psychology,* vol. 44, no. 1, pp. 1-26, March 1991.

[35] I. Briggs Meyers and M. H. McCaulley, Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator, Consulting Psychologists Press, 1985.

[36] M. Cohn, User Stories Applied for Agile Software Development, Boston, MA, USA: Pearson Education, Inc., 2009.

[37] "Blackboard REST API Documentation," Blackboard, [Online]. Available: https://developer.blackboard.com/portal/displayApi. [Accessed 5 June 2021].

[38] "GitLab API Docs," GitLab, [Online]. Available: https://docs.gitlab.com/ee/api/README.html. [Accessed 5 June 2021].

[39] "GitLab GraphiQL," GitLab, [Online]. Available: https://docs.gitlab.com/ee/development/api_graphql_styleguide.html#graphiql. [Accessed 5 June 2021].

[40] "Absolutely FAB," Google, [Online]. Available: https://design.google/library/floating-action-button/. [Accessed 5 June 2021].

[41] T. Y. Siang, "Material Design: Why the Floating Action Button is bad UX design," 1 June 2015. [Online]. Available: https://medium.com/tech-in-asia/material-design-why-the-floating-action-button-is-bad-ux-design-acd5b32c5ef. [Accessed 5 June 2021].

[42] Free Software Foundation, "GNU General Public License v3.0 or later," 29 June 2007. [Online]. Available:

https://spdx.org/licenses/GPL-3.0-or-later.html. [Accessed 5 June 2021].

[43] NTNU, "NTNU strategy 2018-2025," [Online]. Available: https://www.ntnu.edu/strategy. [Accessed 19 May 2021].

[44] P. G. Rein and T. S. Tefre, "Source-code of Odin," [Online]. Available: https://github.com/GitForEdu/odin.

[45] 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*, 1993.

[46] J. Nielsen, "Why You Only Need to Test with 5 Users," 18 March 2000. [Online]. Available: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/. [Accessed 3 June 2021].

[47] A. Hars and S. Ou, "Working for Free? Motivations for Participating in Open-Source Projects," *International Journal of Electronic Commerce,* no. 6:3, pp. 25-39, 2002.

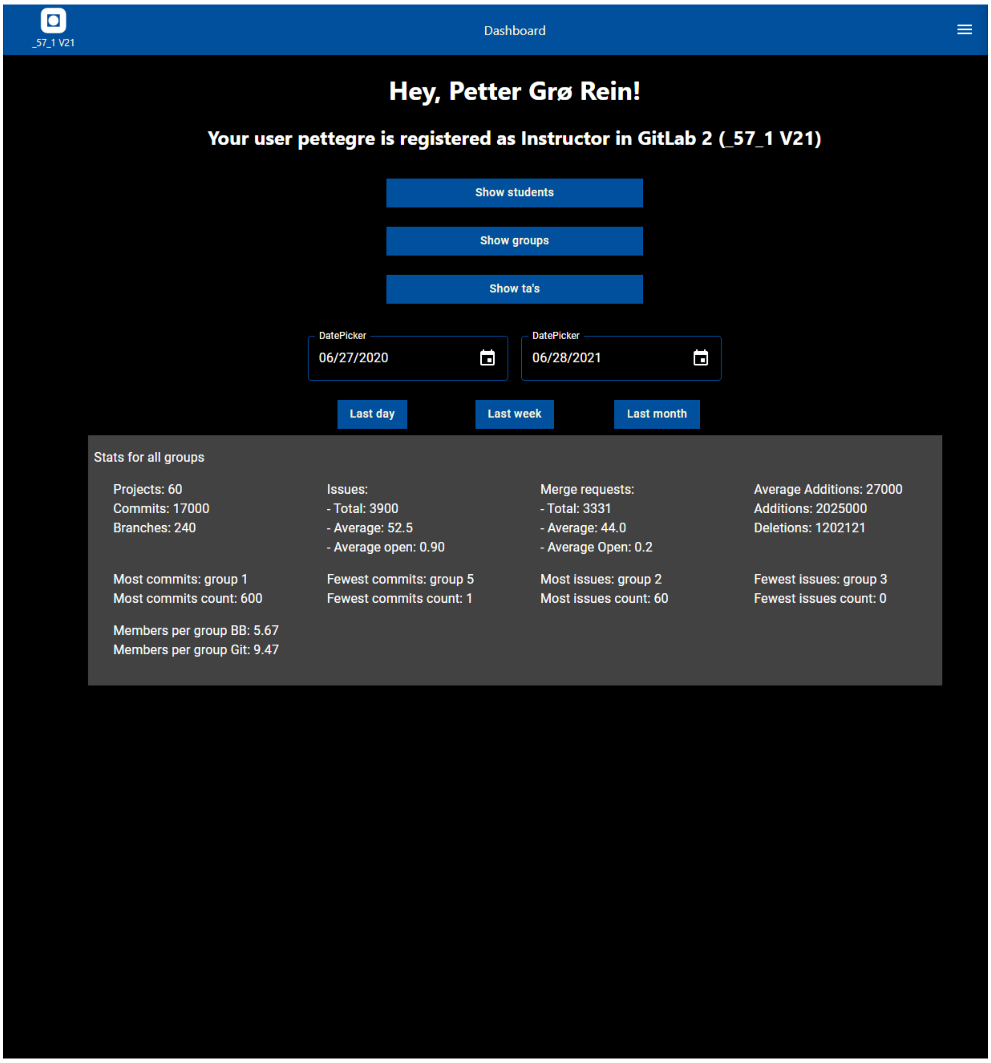
[48] "Docker," Docker, Inc., [Online]. Available: https://www.docker.com/. [Accessed 5 June 2021].

[49] K. Furberg, "Universitetsavisa," 9 December 2016. [Online]. Available: https://www.universitetsavisa.no/student/canvas-valgt-til-felles-plattform-for-e-laering-for-20-institusjoner/134280. [Accessed 27 April 2021].

Appendices

Appendix A – Desktop Screenshots of Pages in the Application

1. Dashboard Page



Hey, Petter Grø Rein!

Your user pettegre is registered as Instructor in GitLab 2 (_57_1 V21)

DatePicker: 06/27/2020

DatePicker: 06/28/2021

Stats for all groups

Projects: 60
Commits: 17000
Branches: 240

Issues:
- Total: 3900
- Average: 52.5
- Average open: 0.90

Merge requests:
- Total: 3331
- Average: 44.0
- Average Open: 0.2

Average Additions: 27000
Additions: 2025000
Deletions: 1202121

Most commits: group 1
Most commits count: 600

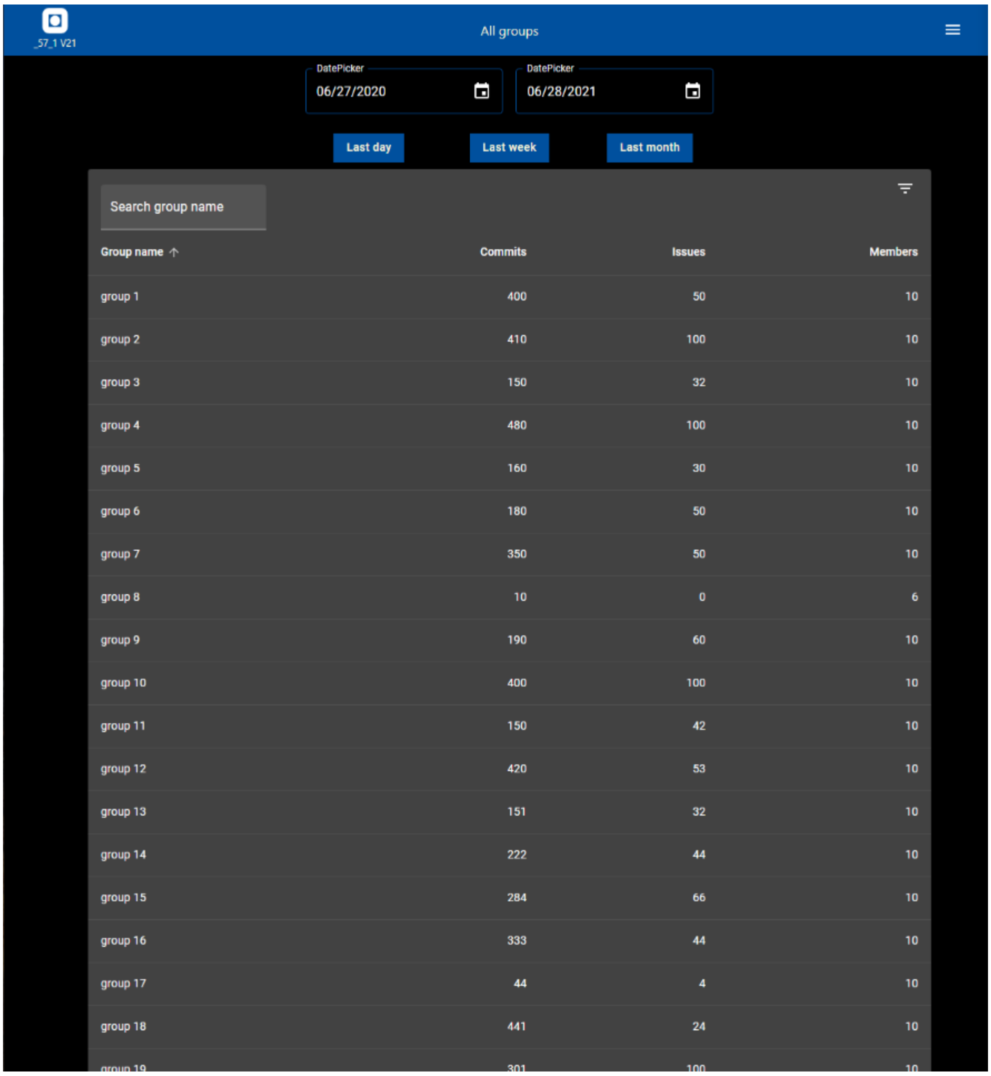
Fewest commits: group 5
Fewest commits count: 1

Most issues: group 2
Most issues count: 60

Fewest issues: group 3
Fewest issues count: 0

Members per group BB: 5.67
Members per group Git: 9.47

2. Group List Page

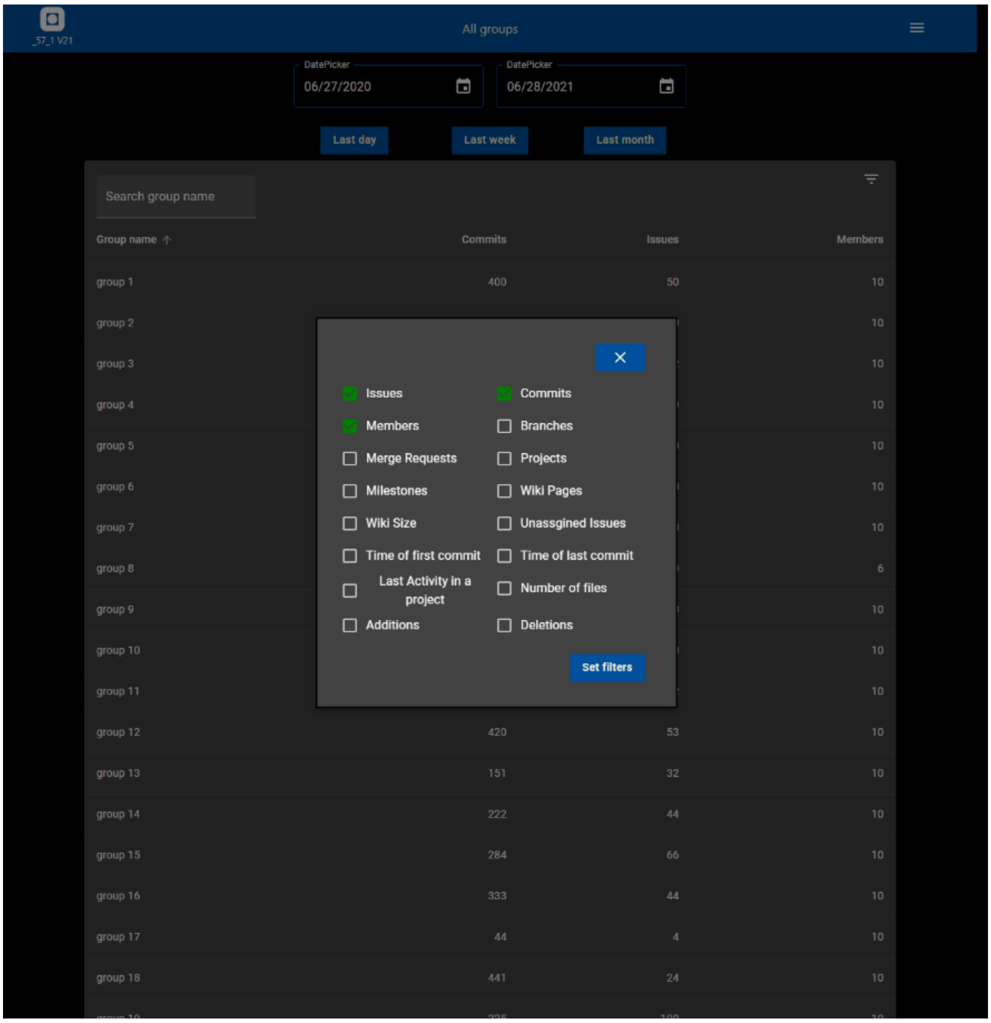


DatePicker 06/27/2020

DatePicker 06/28/2021

Group name ↑	Commits	Issues	Members
group 1	400	50	10
group 2	410	100	10
group 3	150	32	10
group 4	480	100	10
group 5	160	30	10
group 6	180	50	10
group 7	350	50	10
group 8	10	0	6
group 9	190	60	10
group 10	400	100	10
group 11	150	42	10
group 12	420	53	10
group 13	151	32	10
group 14	222	44	10
group 15	284	66	10
group 16	333	44	10
group 17	44	4	10
group 18	441	24	10
group 19	301	100	10

3. Group List Option Module Page



DatePicker: 06/27/2020

DatePicker: 06/28/2021

Group name ↑	Commits	Issues	Members
group 1	400	50	10
group 2			10
group 3			10
group 4			10
group 5			10
group 6			10
group 7			10
group 8			6
group 9			10
group 10			10
group 11			10
group 12	420	53	10
group 13	151	32	10
group 14	222	44	10
group 15	284	66	10
group 16	333	44	10
group 17	44	4	10
group 18	441	24	10
group 19	325	100	10

☑ Issues ☑ Commits

☑ Members □ Branches

□ Merge Requests □ Projects

□ Milestones □ Wiki Pages

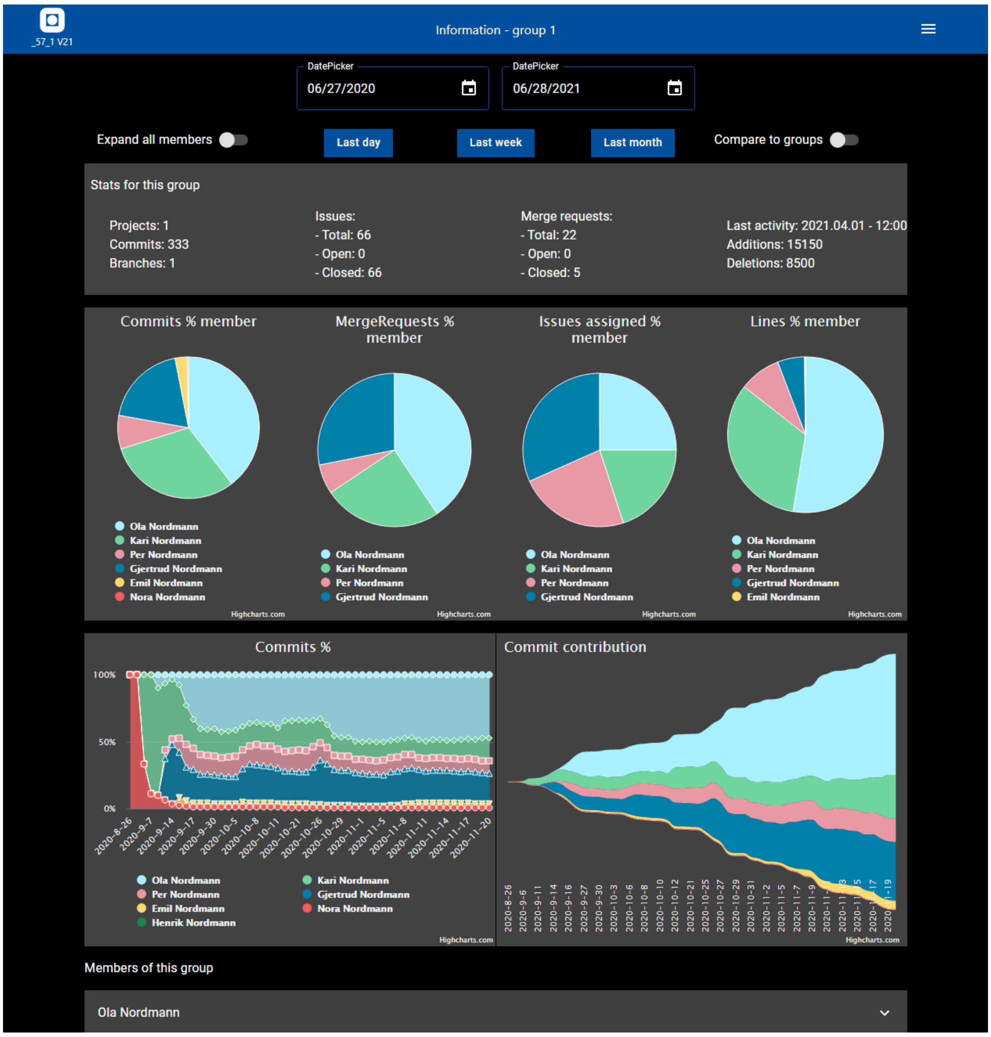
□ Wiki Size □ Unassgined Issues

□ Time of first commit □ Time of last commit

□ Last Activity in a project □ Number of files

□ Additions □ Deletions

4. Group Page



Information - group 1

DatePicker: 06/27/2020

DatePicker: 06/28/2021

Expand all members ○

Compare to groups ○

Stats for this group

Projects: 1
Commits: 333
Branches: 1

Issues:
- Total: 66
- Open: 0
- Closed: 66

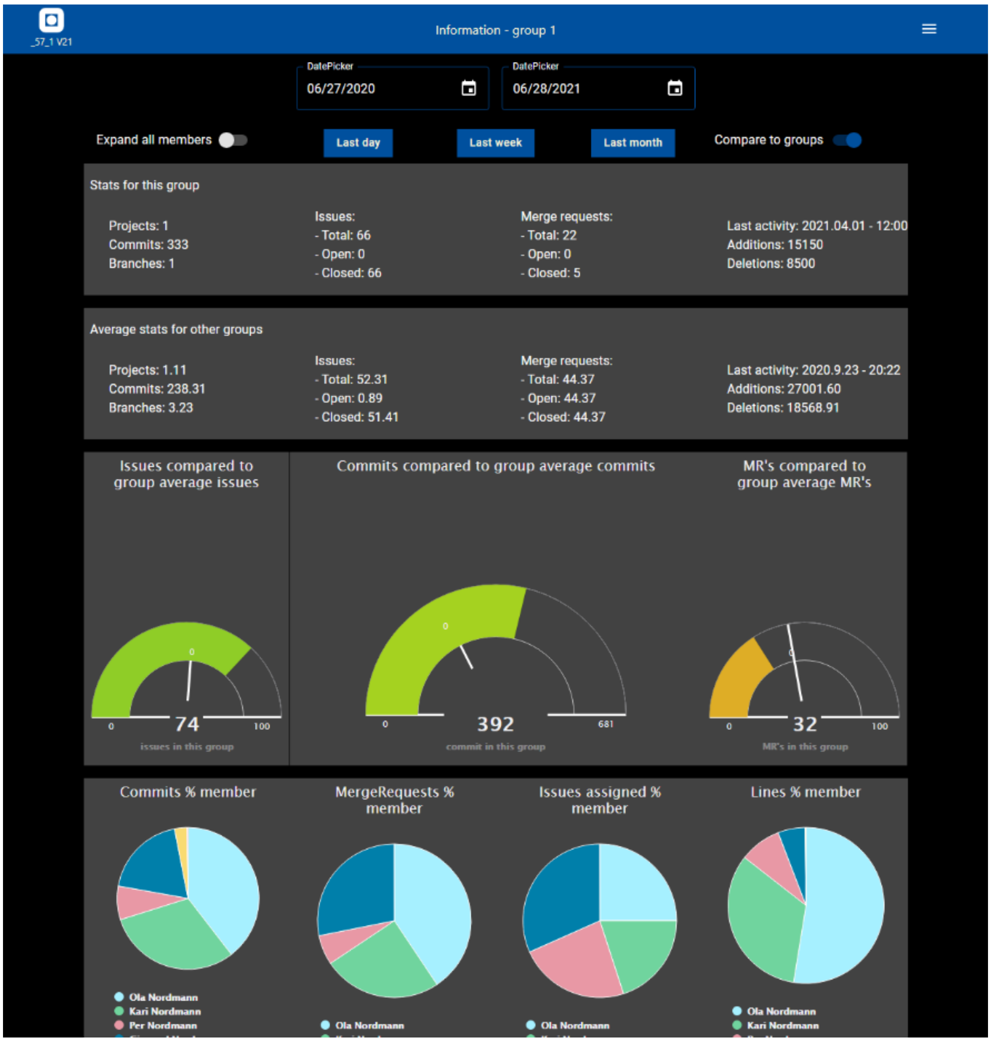
Merge requests:
- Total: 22
- Open: 0
- Closed: 5

Last activity: 2021.04.01 - 12:00
Additions: 15150
Deletions: 8500

Members of this group

Ola Nordmann ⌄

5. Compare a Group with Other Groups Page



06/27/2020

06/28/2021

Stats for this group

Projects: 1
Commits: 333
Branches: 1

Issues:
- Total: 66
- Open: 0
- Closed: 66

Merge requests:
- Total: 22
- Open: 0
- Closed: 5

Last activity: 2021.04.01 - 12:00
Additions: 15150
Deletions: 8500

Average stats for other groups

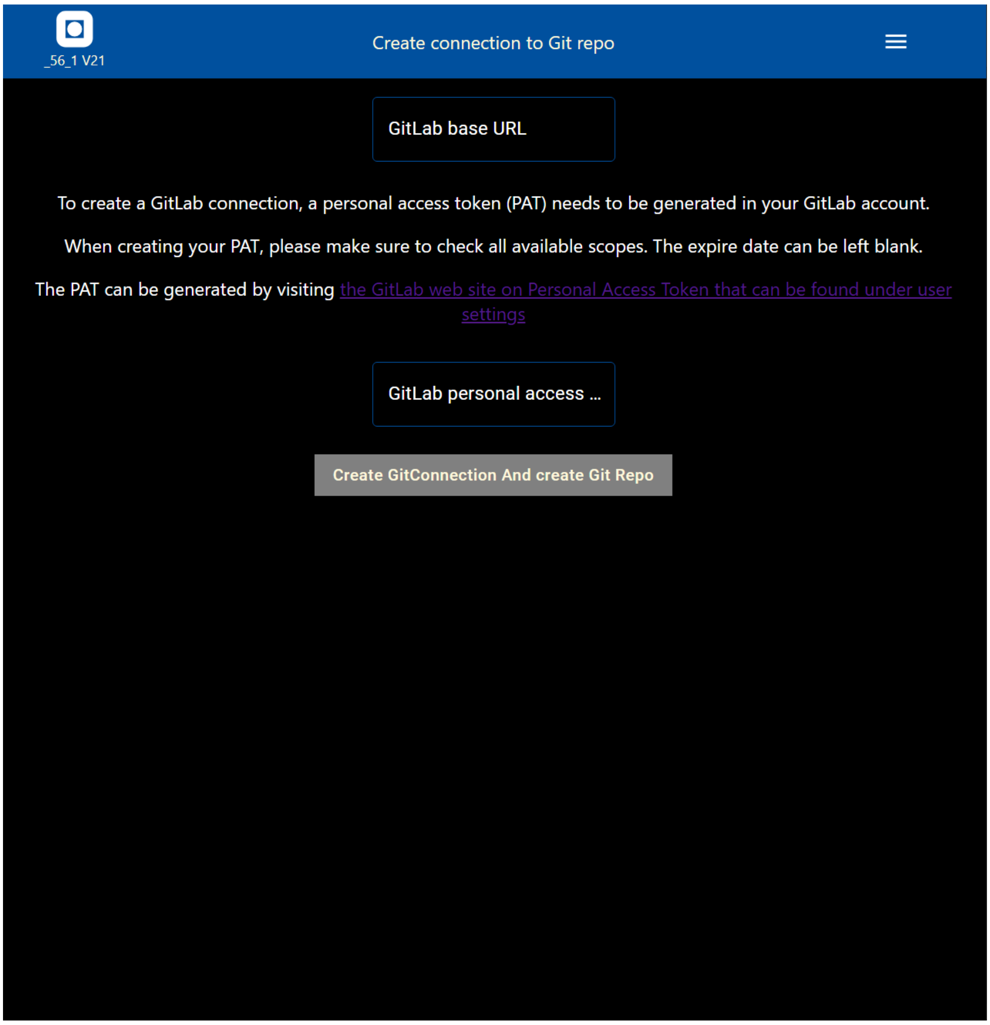
Projects: 1.11
Commits: 238.31
Branches: 3.23

Issues:
- Total: 52.31
- Open: 0.89
- Closed: 51.41

Merge requests:
- Total: 44.37
- Open: 44.37
- Closed: 44.37

Last activity: 2020.9.23 - 20:22
Additions: 27001.60
Deletions: 18568.91

6. Create New Connection Page



_56_1 V21
Create connection to Git repo
≡
GitLab base URL

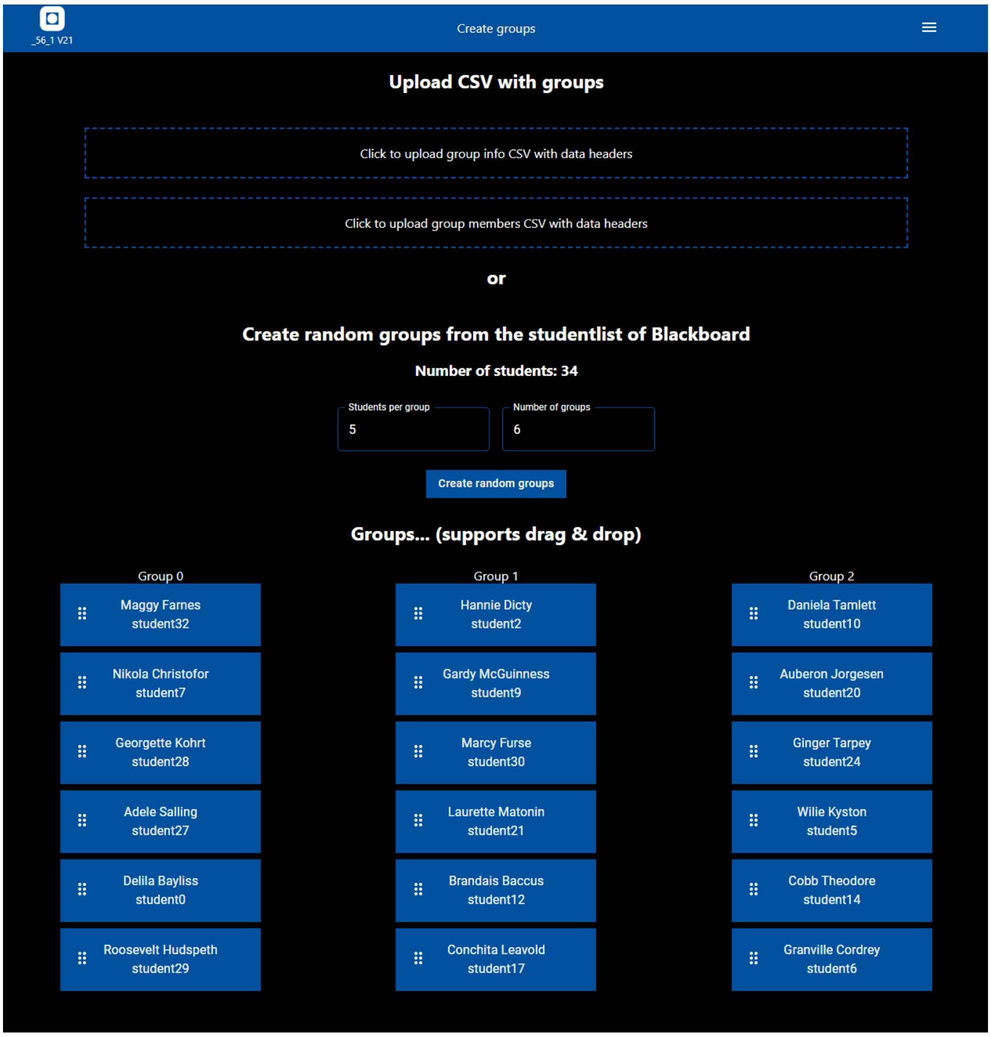
To create a GitLab connection, a personal access token (PAT) needs to be generated in your GitLab account.

When creating your PAT, please make sure to check all available scopes. The expire date can be left blank.

The PAT can be generated by visiting the GitLab web site on Personal Access Token that can be found under user settings

GitLab personal access ...
Create GitConnection And create Git Repo

7. Create Groups in Course Page



Upload CSV with groups

Click to upload group info CSV with data headers

Click to upload group members CSV with data headers

or

Create random groups from the studentlist of Blackboard

Number of students: 34

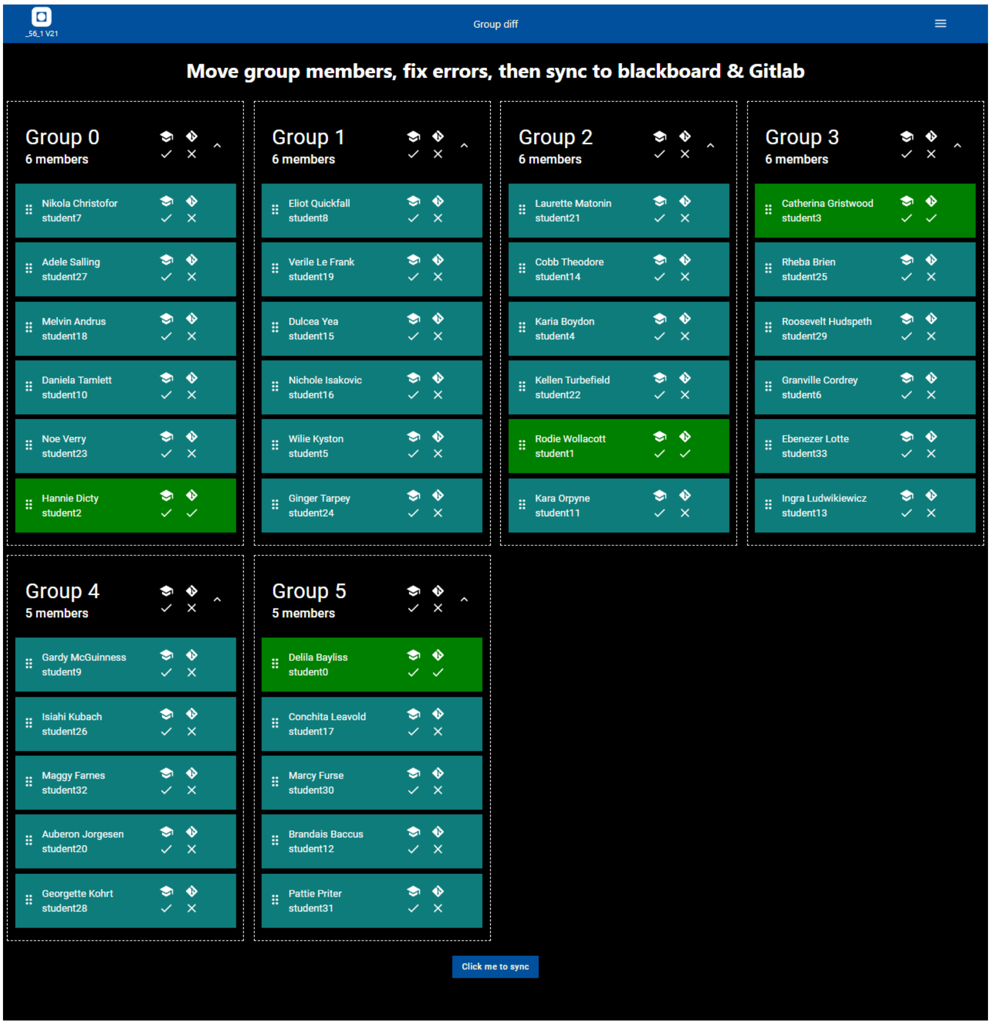
Students per group: 5

Number of groups: 6

Groups... (supports drag & drop)

Group 0	Group 1	Group 2
Maggy Farnes student32	Hannie Dicty student2	Daniela Tamlett student10
Nikola Christofor student7	Gardy McGuinness student9	Auberon Jorgesen student20
Georgette Kohrt student28	Marcy Furse student30	Ginger Tarpey student24
Adele Salling student27	Laurette Matonin student21	Wilie Kyston student5
Delila Bayliss student0	Brandais Baccus student12	Cobb Theodore student14
Roosevelt Hudspeth student29	Conchita Leavold student17	Granville Cordrey student6

8. Group Edit and Difference Page



Move group members, fix errors, then sync to blackboard & Gitlab

Group 0 ✓ ✕
6 members

Nikola Christofor student7 ✓ ✕
Adele Salling student27 ✓ ✕
Melvin Andrus student18 ✓ ✕
Daniela Tamlett student10 ✓ ✕
Noe Verry student23 ✓ ✕
Hannie Dicty student2 ✓ ✓

Group 1 ✓ ✕
6 members

Eliot Quickfall student8 ✓ ✕
Verile Le Frank student19 ✓ ✕
Dulcea Yea student15 ✓ ✕
Nichole Isakovic student16 ✓ ✕
Wilie Kyston student5 ✓ ✕
Ginger Tarpey student24 ✓ ✕

Group 2 ✓ ✕
6 members

Laurette Matonin student21 ✓ ✕
Cobb Theodore student14 ✓ ✕
Karia Boydon student4 ✓ ✕
Kellen Turbefield student22 ✓ ✕
Rodie Wollacott student1 ✓ ✓
Kara Orpyne student11 ✓ ✕

Group 3 ✓ ✕
6 members

Catherina Gristwood student3 ✓ ✓
Rheba Brien student25 ✓ ✕
Roosevelt Hudspeth student29 ✓ ✕
Granville Cordrey student6 ✓ ✕
Ebenezer Lotte student33 ✓ ✕
Ingra Ludwikiewicz student13 ✓ ✕

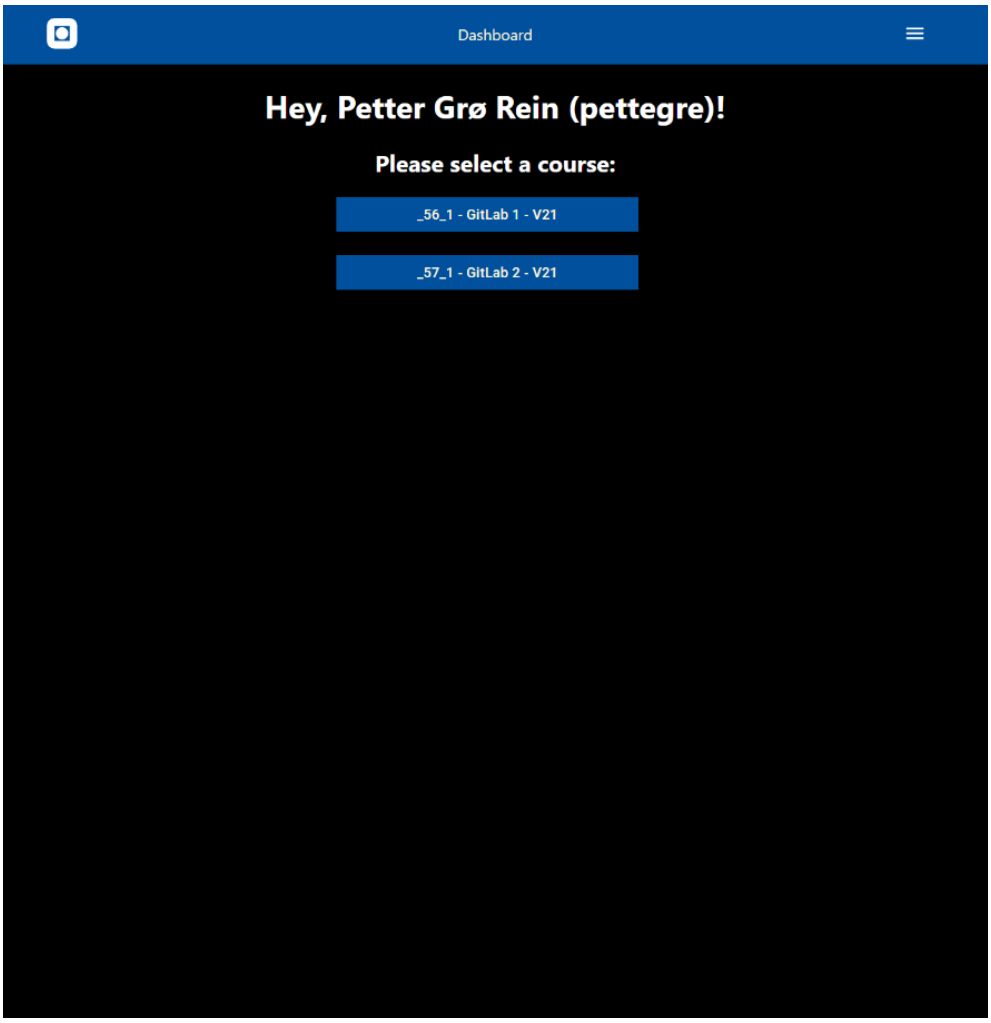
Group 4 ✓ ✕
5 members

Gardy McGuinness student9 ✓ ✕
Isiahi Kubach student26 ✓ ✕
Maggy Farnes student32 ✓ ✕
Auberon Jorgesen student20 ✓ ✕
Georgette Kohrt student28 ✓ ✕

Group 5 ✓ ✕
5 members

Delila Bayliss student0 ✓ ✓
Conchita Leavold student17 ✓ ✕
Marcy Furse student30 ✓ ✕
Brandais Baccus student12 ✓ ✕
Pattie Priter student31 ✓ ✕

9. Your Courses Page

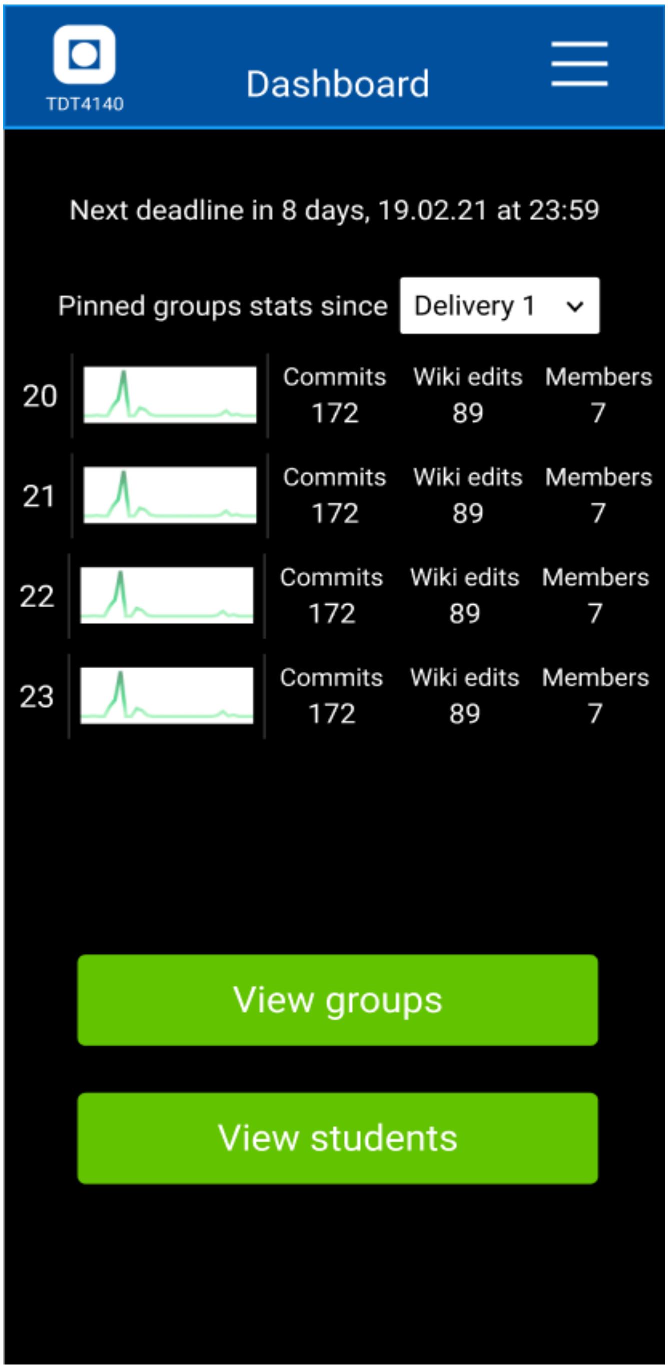


Hey, Petter Grø Rein (pettegre)!

Please select a course:

Appendix B – Mobile Screenshots of Pages in Figma Prototype

1. Dashboard Page



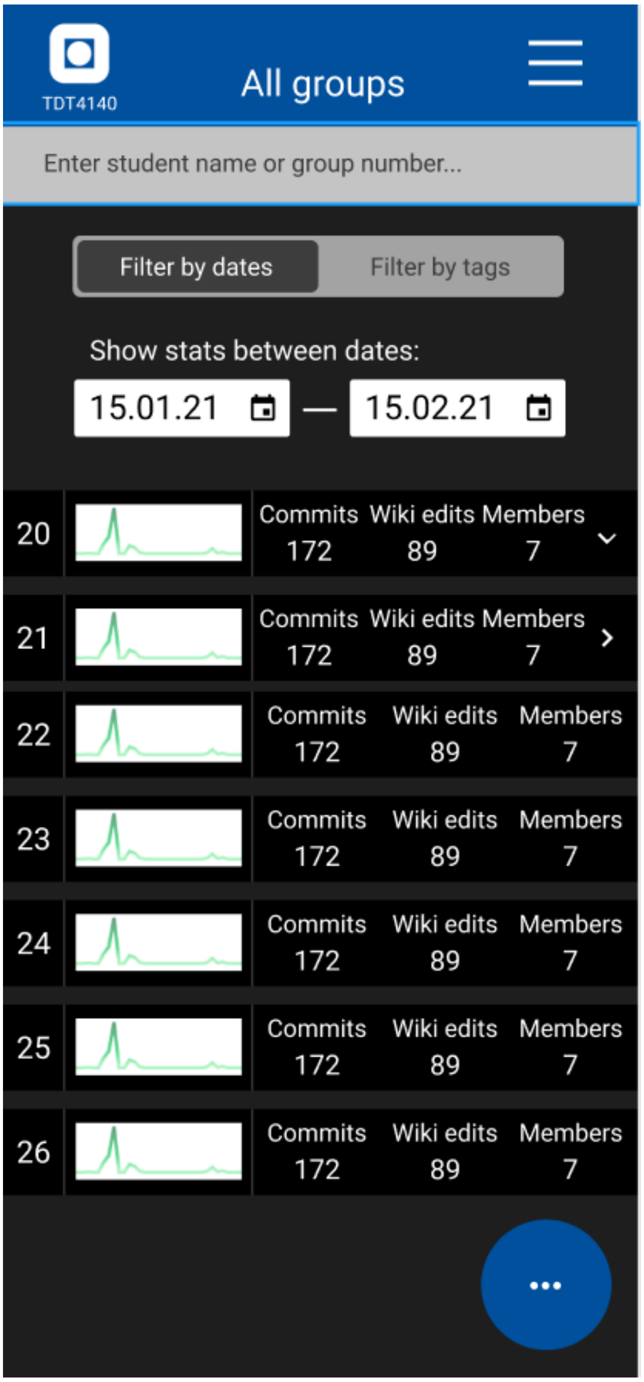
Dashboard

Next deadline in 8 days, 19.02.21 at 23:59

Pinned groups stats since Delivery 1

	Commits	Wiki edits	Members
20	172	89	7
21	172	89	7
22	172	89	7
23	172	89	7

2. Group List Page

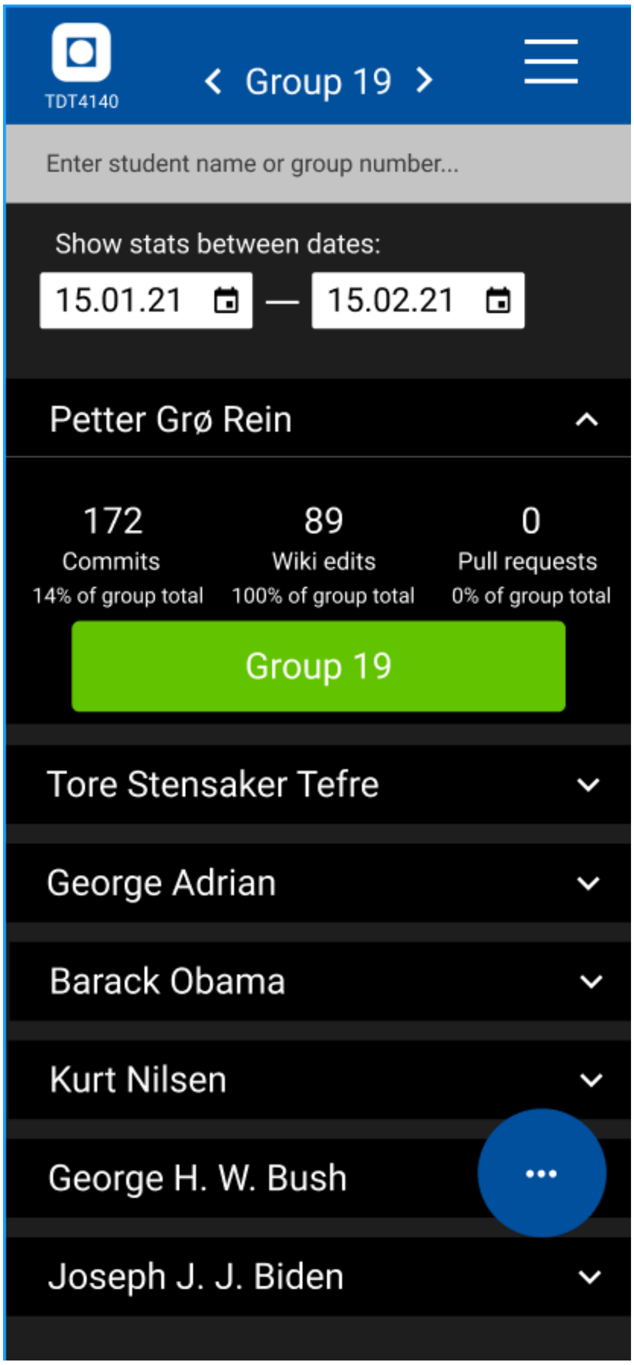


Show stats between dates:

15.01.21 — 15.02.21

	Commits	Wiki edits	Members
20	172	89	7
21	172	89	7
22	172	89	7
23	172	89	7
24	172	89	7
25	172	89	7
26	172	89	7

3. Group Page



TDT4140

‹ Group 19 ›

Enter student name or group number...

Show stats between dates:

15.01.21 — 15.02.21

Petter Grø Rein

172	89	0
Commits	Wiki edits	Pull requests
14% of group total	100% of group total	0% of group total

Group 19

Tore Stensaker Tefre

George Adrian

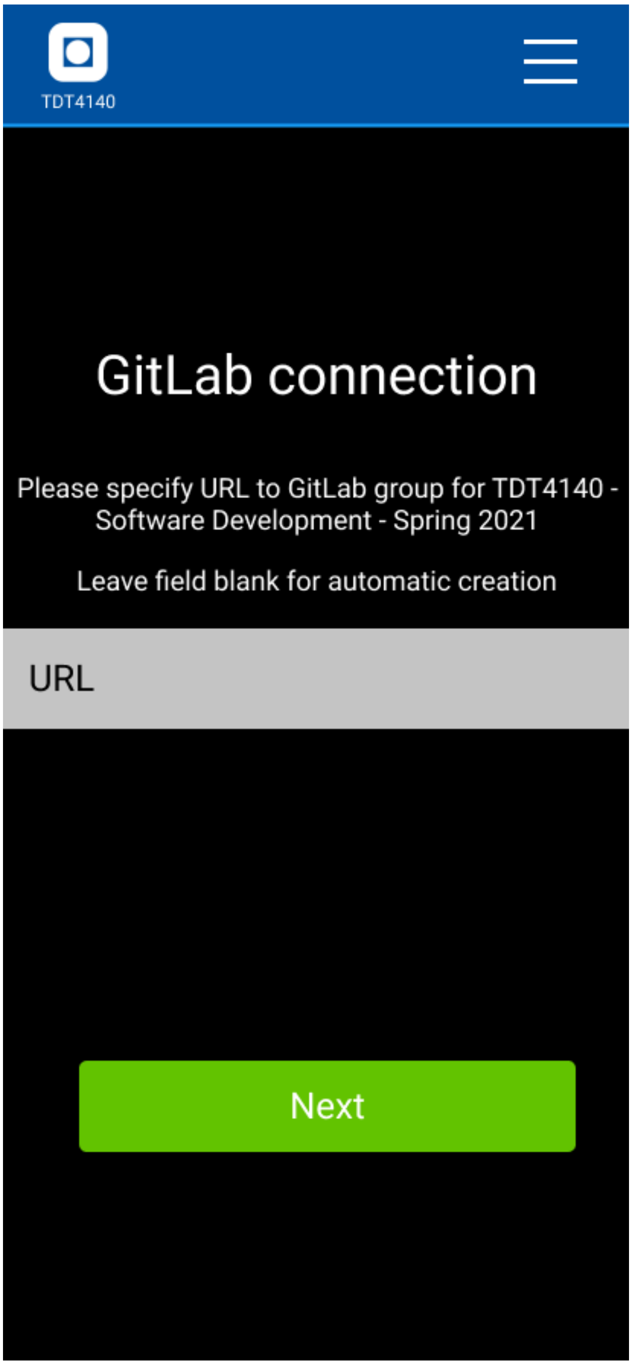
Barack Obama

Kurt Nilsen

George H. W. Bush

Joseph J. J. Biden

4. Create New Connection Page



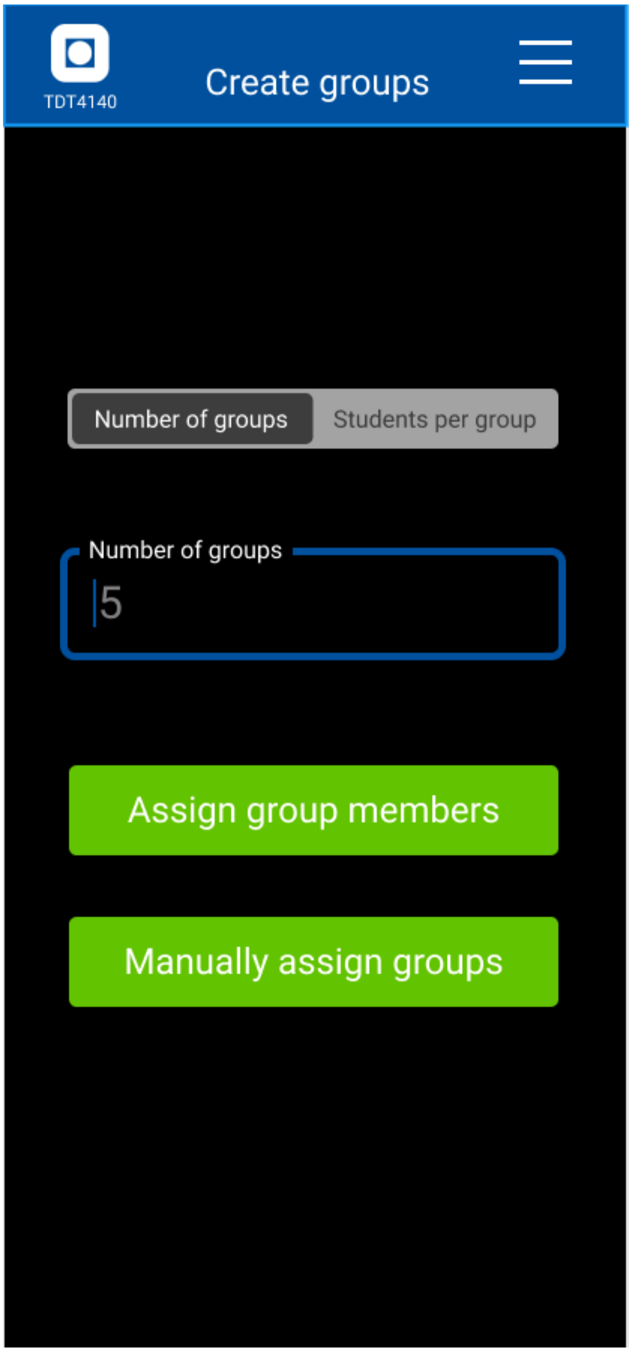
GitLab connection

Please specify URL to GitLab group for TDT4140 - Software Development - Spring 2021

Leave field blank for automatic creation

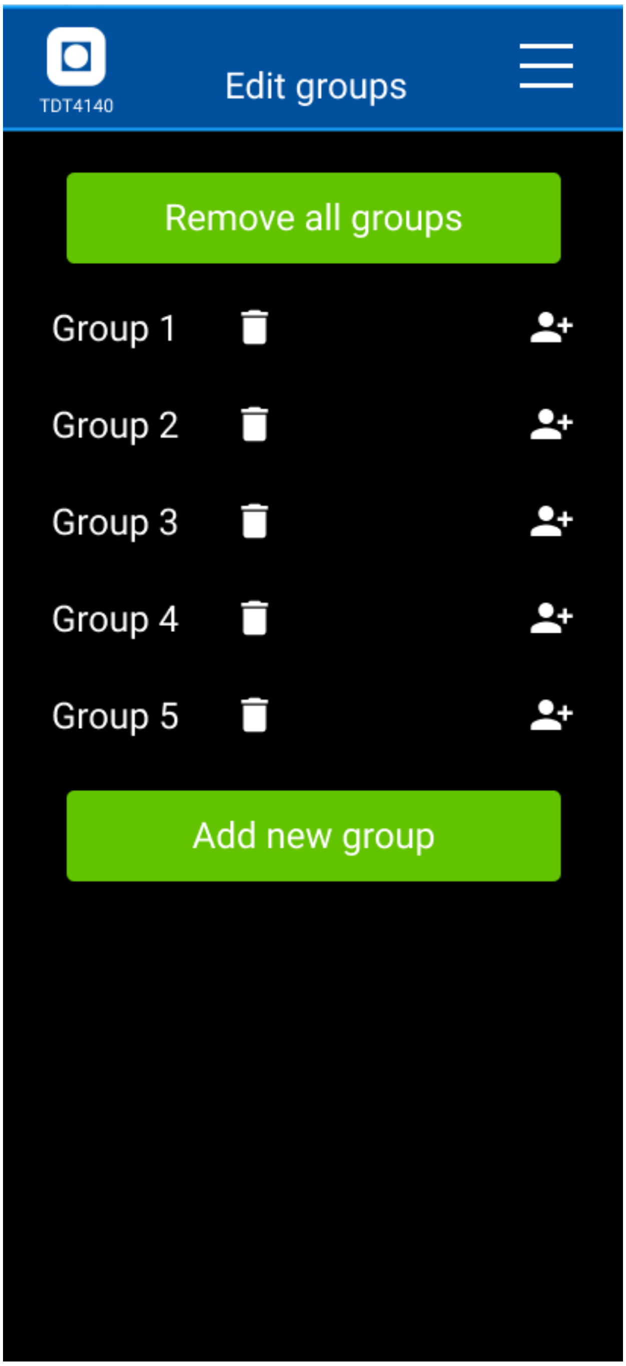
URL

5. Create Groups in Course Page

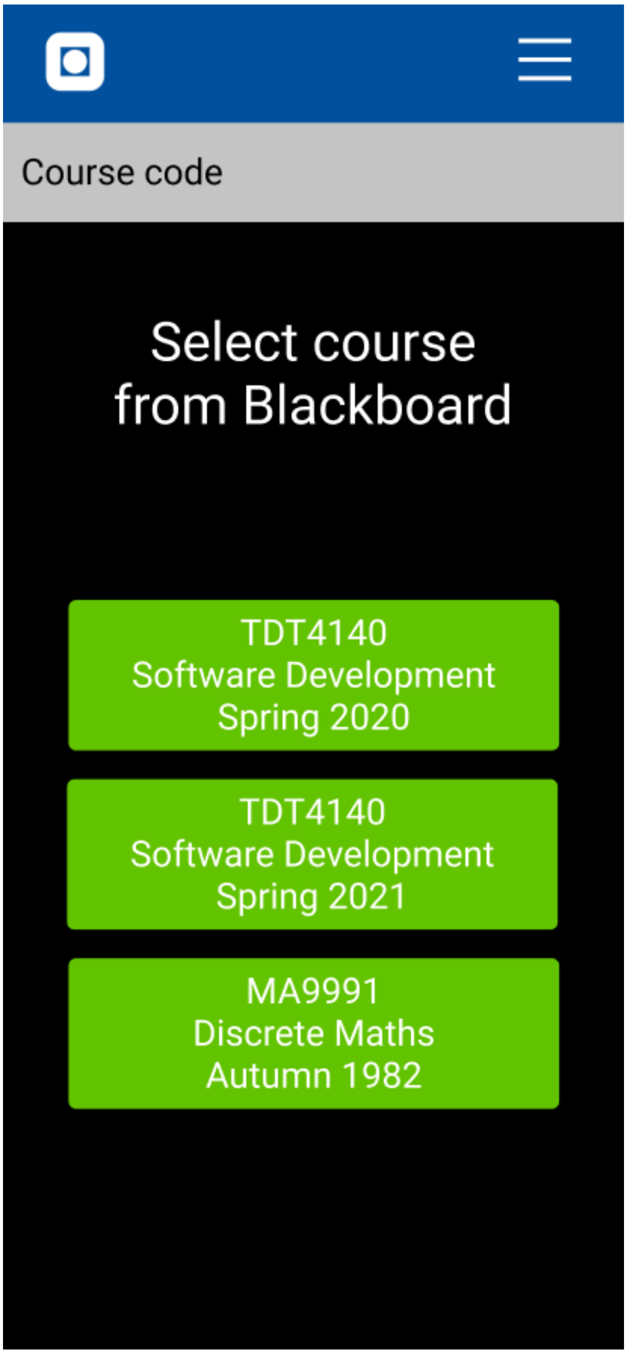


Number of groups: 5

6. Group Edit Page



6. Your Courses Page



Course code

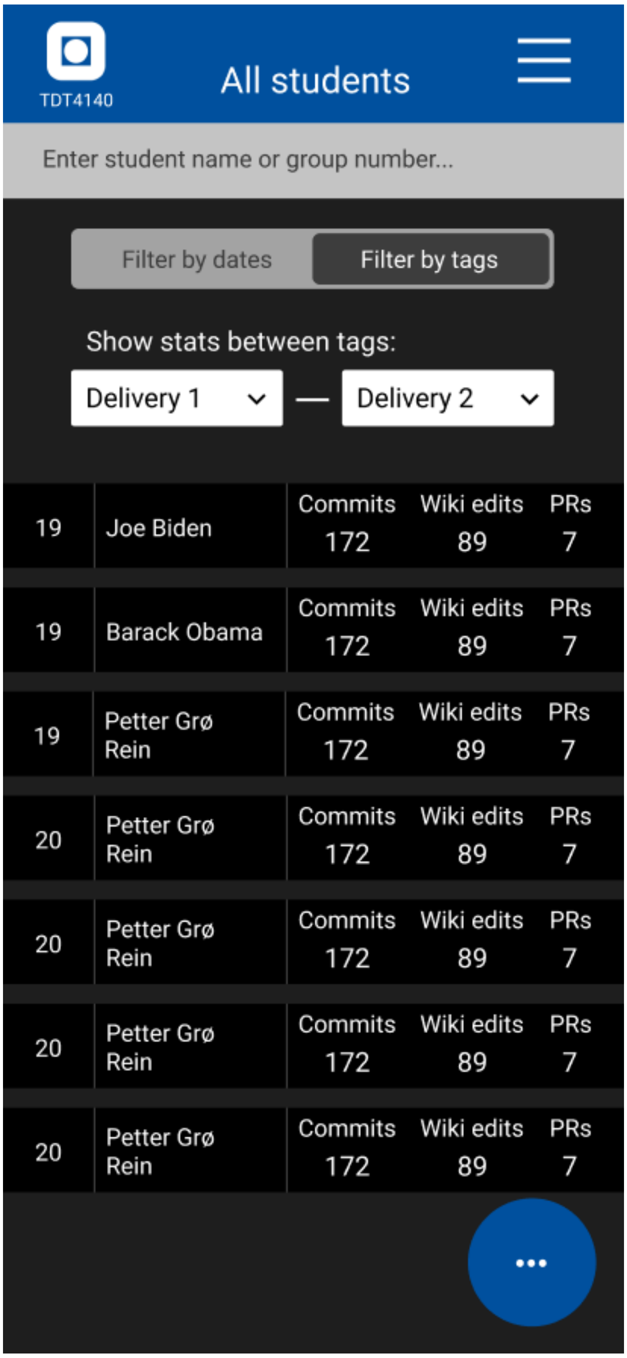
Select course from Blackboard

TDT4140
Software Development
Spring 2020

TDT4140
Software Development
Spring 2021

MA9991
Discrete Maths
Autumn 1982

7. All Students in Course Page



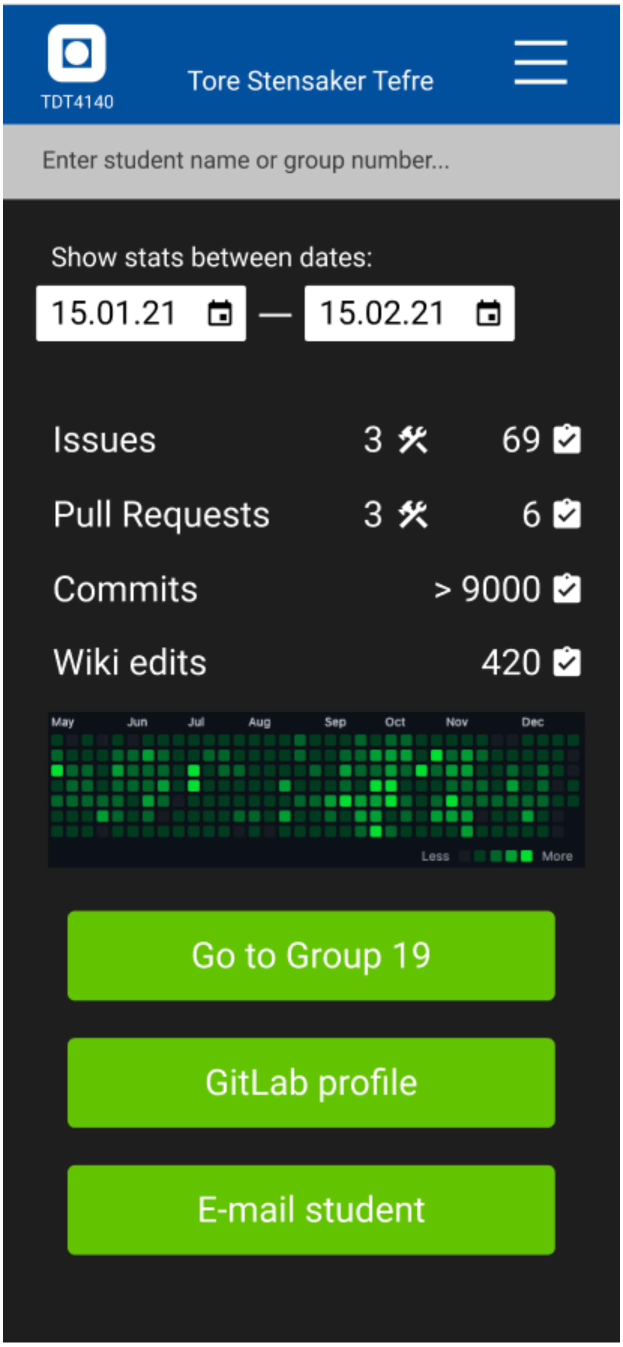
Enter student name or group number...

Show stats between tags:

Delivery 1 — Delivery 2

19	Joe Biden	Commits 172	Wiki edits 89	PRs 7
19	Barack Obama	Commits 172	Wiki edits 89	PRs 7
19	Petter Grø Rein	Commits 172	Wiki edits 89	PRs 7
20	Petter Grø Rein	Commits 172	Wiki edits 89	PRs 7
20	Petter Grø Rein	Commits 172	Wiki edits 89	PRs 7
20	Petter Grø Rein	Commits 172	Wiki edits 89	PRs 7
20	Petter Grø Rein	Commits 172	Wiki edits 89	PRs 7

8. Student Page



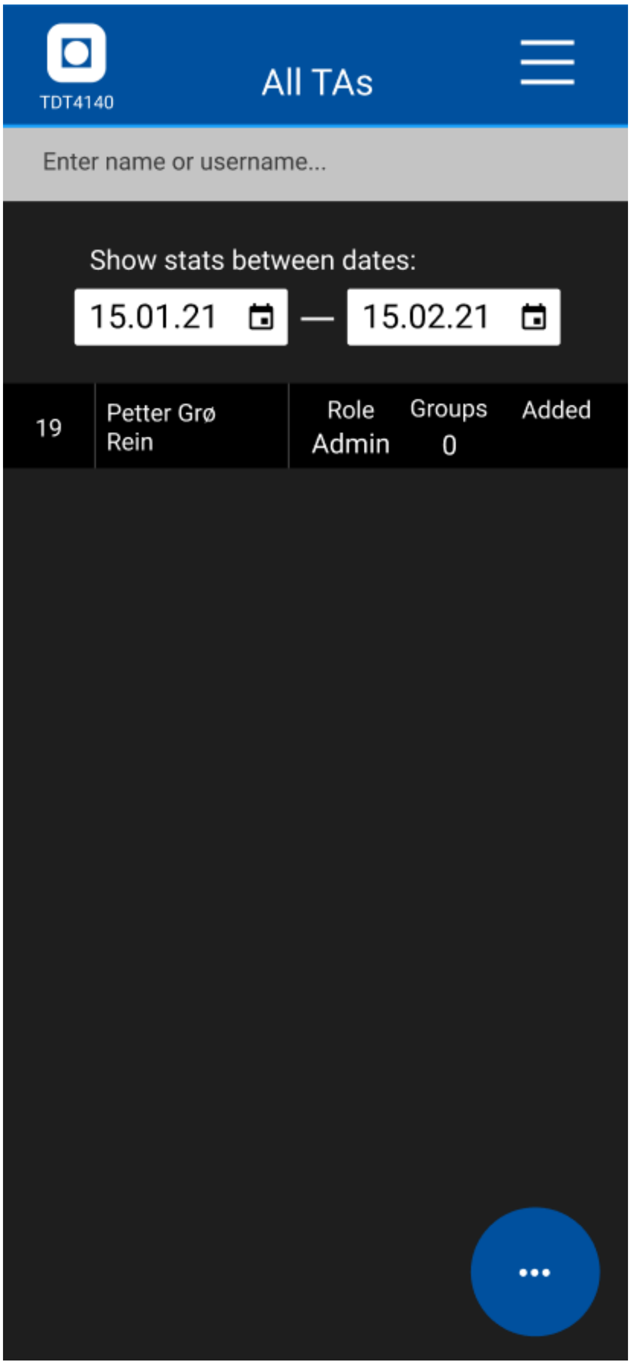
Enter student name or group number...

Show stats between dates:

15.01.21 — 15.02.21

Issues	3 🛠	69 ✓
Pull Requests	3 🛠	6 ✓
Commits		> 9000 ✓
Wiki edits		420 ✓

9. TA's in Course Page

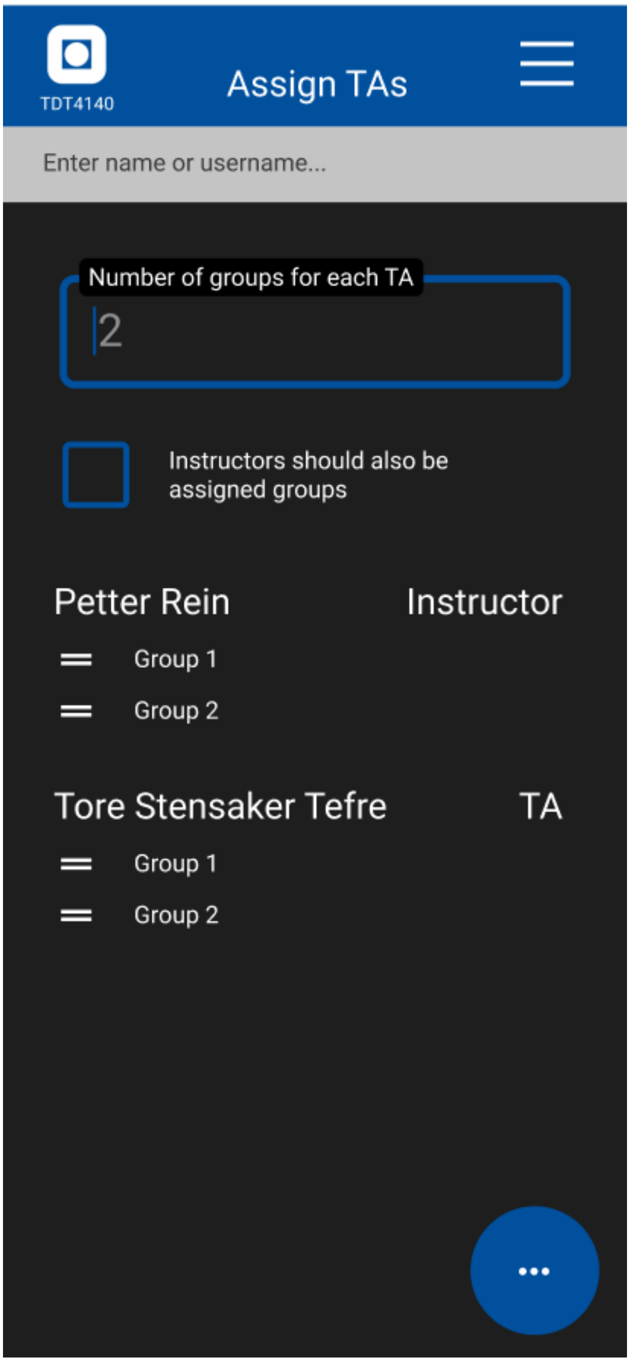


Show stats between dates:

15.01.21 — 15.02.21

		Role	Groups	Added
19	Petter Grø Rein	Admin	0	

10. TA Responsibility Edit Page



TDT4140
Assign TAs
Enter name or username...
Number of groups for each TA
2
Instructors should also be assigned groups
Petter Rein
Instructor
Group 1
Group 2
Tore Stensaker Tefre
TA
Group 1
Group 2
•••

Petter Grø Rein and Tore Stensaker Tefre

Creating a Web Application Supporting Git in Software Development Courses in Higher Education

NTNU

Norwegian University of
Science and Technology