Henrik Backer & Robin Lamy

Team Accelerator 2.0

Web-application for monitoring group work in an academic setting.

Master's thesis in Computer Science & Master of Science in Informatics
Supervisor: George Adrian Stoica
July 2022

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Henrik Backer & Robin Lamy

Team Accelerator 2.0

Web-application for monitoring group work in an academic setting.

Master’s thesis in Computer Science & Master of Science in Informatics
Supervisor: George Adrian Stoica
July 2022

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

NTNU
Norwegian University of
Science and Technology

Abstract

Group work is playing a major role in higher education's pedagogy. Exploiting the full potential of group work requires building a solid team core, which will be able to reach its best capability while being monitored by professors and teaching assistants. In a physical context, people knew how to communicate and work together. Technology and information system offered new ways and arenas for group interaction to happen. Remotely or not, information systems offered new possibilities to collaborate, create, and work for groups. In the specific case of software-assisted group work in higher education, different parameters must be taken into account. In this master thesis, we present results obtained from improving a group work support tool in higher education, with a special focus on the teaching staff side monitoring group work. Starting from a previous Master Thesis work [1], a web application, we present several improvements in data processing and visualization, as well as new functionalities. Two iterations of development and tests lead us to this paper's results. Then, we discussed the outcome of these iterations and our methods to check the validity of our work, as well as investigating the impact of our application improvements on group work at NTNU. To conclude we open this paper by exploring potential future work linked to our usability tests.

Sammendrag

Gruppearbeid spiller en stor rolle for pedagogikk innen høyere utdanning. For å utnytte hele potensialet til gruppearbeid trenger man en god struktur og gode forhold, mens professorer og læringsassistenter passer på, og følger med. Ny teknologi med nye informasjonssystemer åpnet opp for andre måter å kommunisere og sammarbeide med gruppearbeid på. Programvareassistert gruppearbeid under høyere utdanning krever derfor at man tar andre parametere i betraktning. I denne rapporten begynner vi med en analyse av gruppearbeid i høyere utdanning og potensiale til informasjonssystemer for å forbedre effektiviteten. Ved å begynne fra en tidligere masteroppgave [1], en web-applikasjon, så presenterer vi flere forbedringer til dataprosessering og visualisering, i tillegg til å implemetere nye funksjonaliteter. To iterasjoner med utvikling og brukertester førte oss til resultatet til denne rapporten. Så diskuterer vi utfallet til dette resultatet og våre metoder brukt under prosjektet. Til slutt utforsker vi påvirkingen applikasjonen har hatt for gruppearbeid ved NTNU, samt åpner opp for videre utvikling og testing ved å presentere forslag til fremtidig arbeid.

Code Repositories

Backend Repository in Gitlab
`https://gitlab.stud.idi.ntnu.no/henrikmb/master_api`

Frontend Repository in Gitlab
`https://gitlab.stud.idi.ntnu.no/henrikmb/master_react`

Contents

Figures

Tables

Chapter 1

Introduction

Group work is seen everywhere and is how most people get things done. Nevertheless, group work has both pros and cons, in contrary to individual work, group work requires coordination and communication to be efficient. Furthermore, group work also involves individual work, surrounded by group management and information sharing. As detailed in [2], many issues might influence group work efficiencies, such as poor communication, personal issues, or unequal contribution. These aspects need to be taken care of to be efficient and fast while working in a group. In an academic environment, people may be placed involuntarily in groups which comes with challenges. It also happens that students can fully or partially create groups themselves, which involves other problems such as unequal affinities within the group. Many students experience their first group project over a longer duration. Others have experiences that may influence how they interact in such an environment. Everyone, however, will need to adapt to working in a group usually with people you have never met, or people you have never been together with within a group work-related project. How people talk with each other, in terms of collaboration, planning, and dealing with issues, which always come up to some degree, is crucial to the overall result of the project [3]. In this thesis, we investigate how a tool can be used in an academic group work-related project, and in what value such a tool brings. Specifically, the tool is a web-based application called Team Accelerator, which is being tested in a real-world course at NTNU and improved following a user-centered design process. Team Accelerator is an application made by a student at NTNU during a previous master's project, and we are continuing his work.

1.1 Motivation

Our motivation for this master thesis can be summed up in two parts. Having both lots of experience with group-related projects during our education, and oftentimes witnessing firsthand problems being procrastinated, our first part of the motivation regards finding ways to improve such a scenario. The lack of aids provided for dealing with issues in terms of group work calls for further research into this topic. This leads us to the second part of our motivation, developing a full-stack web application. Team Accelerator, which aims to improve group work for those who are struggling, was a working prototype at the start of this thesis. This would mean we had the possibility of running a proper experiment in a real-world scenario within a course at NTNU which relies heavily on group work. Gathering useful and relevant data, along with improving the application using a human-centered approach, was something we both were motivated by. This would include maintaining a web server, talking with students, and implementing multiple iterations of the design based on user testing.

1.2 Context

Team Accelerator is a web application first developed by a previous student in his master thesis [1]. It aims to improve group work by detecting groups that are struggling as early as possible. Students can give a rating each week, a number from 1 to 5, expressing how they feel the group work is going. The application has different views; one for the students, and several for the teaching staff, which includes, teaching assistants, lecturers, and course coordinators. Teaching Assistants may be assigned to supervise a limited number of groups while others are supervising the entire class. As the work done previously in this app [1] focused mainly on the student's view, in this master thesis we will focus on the remaining sides of the application. This means an improved interface for monitoring and managing a course within the app. The app was tested at NTNU this spring semester by TDT4180: Human-Computer Interaction. Courses typically have 500 enrolled students forming around 100 groups and they are evaluated 100% from group work. Groups typically have 5 students that chose their teammates themselves. Students that could not find 5 members would be merged with other students by the course facilitators. Every group has assigned a teaching assistant, who has about five groups each. This course has 23 teaching assistants and two undasses.

1.3 Research Question

When facilitating a course that includes group work with more than 500 students, the teaching assistants and instructors rely on students to manage themselves most of the time. They can't closely watch and direct the groups, to resolve every issue that comes up. Groups that are struggling need to work out problems within the group, or ask their responsible teaching assistant for help. As the preceding thesis that our thesis follows found out during his research [1], there is a need for an application to facilitate easy communication between instructors and teams, where the aim is to prevent bad teamwork. After developing such an application, he concludes his master thesis by saying it needs to be tested and improved more, especially by publishing the application in a real course at the start of the semester. To further investigate this topic, by continuing the development of Team Accelerator, we found it natural to improve the way teaching assistants and instructors interact with the application. A majority of this work lies in the visualization of data from the students. First, we want to find out how users interact with the application by deploying it from the start of the semester. Secondly, we want to find out how Team Accelerator can be improved for teaching assistants and instructors to more easily spot groups that are struggling.

The research questions for this master thesis are as follows:

RQ1: In what ways can a tool such as Team Accelerator improve group-work-related projects in an academic setting?

RQ1.1: How can we improve the design of a web application to better find groups that are struggling?

1.4 Document Outline

The master thesis starts with our background and research for this project. It includes theory on group work, user testing, a human-centered design process, and related work. Chapter 4.1 explains the architecture of the application, Team Accelerator. First, it gives an overall structure and then goes into detail about the architectures of the backend, the frontend, and the database. Also, it describes how the application was deployed using a virtual machine at NTNU. Chapter 3 discusses the methods used and the activities we did to answer our research question. First, we describe our development method, and how we implemented new features. Then we go into detail about how we conducted the user tests, including a test plan and tasks for each test. The chapter ends with a description of the measures we did to get participants for our project. Chapter 4 contains all iterations of the prototype of Team Accelerator. To go through each iteration and goes into detail about the features that we added. The following chapter, chapter 5 holds all of our results which are divided into qualitative and quantitative results. It highlights the feedback given from users both explicit and implicit. Chapter 6 is our discussion using the results shown in the previous chapter. It describes the changes we made to the prototype based on feedback from the user tests, including the validity and biases of our results. Chapter 7 contains our conclusion of this thesis and suggestions for future work if this project would be taken further by someone else. Lastly, chapter 8 is our acknowledgments for this project. Figure 1.1 below shows a timeline that highlights important events and phases throughout this project

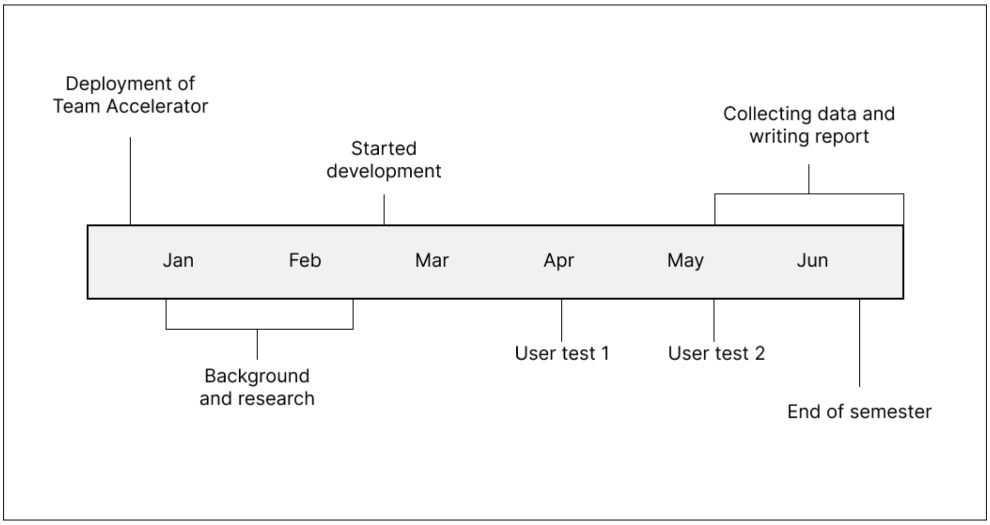


Figure 1.1: Project timeline

Chapter 2

Background and Research

The Background and Research chapter outline the preparatory work that was done ahead of the development of the application, user tests, and results analysis. Our work followed a Human Centered Design paradigm, with Agile development methods for each iteration of improvement. We tried to cover every aspects of research that would occur during this master thesis. Among these subjects, group work theory is the most important one as its understanding allowed us to take core design decision for the first iteration, which was not supported by any user test yet. However, many other subjects played a role in the rest of the work such as, human centered design and usability test, agile methods of development, and research biases related to our field of study. This chapter tends cover and explain the background knowledge necessary to understand our working methods and choices in this master thesis.

2.1 Continuing an existing project

As presented in the Master Thesis [1], different implementation choices were made at the application creation. Of course the goal of this master thesis is not to reshape the existing prototype. Our purpose is to improve its functionalities and reach its full potential to help group work monitoring by Teaching Assistants and Instructors. However, considering the fact that we are taking on an existing project, we decided to discuss and understand the choices made by the first student to work on this application.

2.1.1 Front-End

In his Master Thesis [1], August Lund Eilertsen explains the pros of working with React and Redux. Considering the work that was done when we took over this project, we decided to keep working with the technologies used. However, according to the future of the application and its current state of development, we can discuss the possibility to work with the Angular or Vue frameworks instead. These two frameworks are the most popular with React to develop web applications, but each comes with its specificities. [4] [5]

Angular

While Angular is known to be convenient to maintain large-size application, its learning curves is way steeper than the Vue and React framework (meaning that mastering Angular framework takes a lot more time than mastering React or Vue. The main reason is that Angular's core is a lot bigger than the rest.) [4]. Since we took over the previous work done by August Lund Eilertsen, it is very likely that other students are going to take on our work to keep improving the current application. The master thesis takes no more than a year, and taking on existing code is very time-consuming, mastering a complex framework in addition could shift the focus of the work done by students. Hence it is important to keep the technologies used easy to master so that students can focus on the thesis proper subject.

Vue

On the other hand, the Vue framework's learning curve is flatter, thus it could be a great candidate for student to master it very quickly. The inconvenient is that Vue does not fit large application development, which makes taking over existing code and shared development a lot more complex. With these information, React learning curve and its convenience with large application development, make it an acceptable compromise that fits our requirements. It is worth mentioning that React flat learning curve comes with the necessity to use external packages, such as Redux since the React core is very light.

React, Vue, and Angular, all have high computing performances which makes this aspect of the frameworks less important as performances are completely comparable [5].

2.1.2 Back-End

When it comes to back-end, many other technologies and possibilities were available. However, as detailed in the Master Thesis [1], Python Django Framework fit well the needs and requirements of this application prototype. From Django learning curve to its easy maintainability, even with large application, this framework is accessible and reduces the developer knowledge requirements to Python, excluding the need to write raw SQL code. The Java Spring Framework could have been an alternative also, but considering Python accessibility and simplicity regarding development purposes, we decided to keep using Django.

2.2 Group Work

Group work in higher education became the core of learning with constructivist pedagogy, using alternative teaching strategies. As detailed in [3], contemporary teaching methods involve a lot more of case study, group learning and group projects, in order to be as close as possible to what work after graduation is like. This shift from instructivist to constructivist teaching methods, forced educators to use more and more collaborative teaching and learning techniques with their students. Therefore, there was a true need to deeply understand how to make group work effective and relevant for students to extract all the potential of these new methods.

As detailed before, our focus on this application development is mostly on the TA and course Instructor perspective. In order to build a relevant and useful application for TAs to monitor the groups from their course effectively, we first need to understand what make group work efficient. Hence we can provide TAs the right functionalities, information, and tools to allow them to help groups working on project as much as possible.

2.2.1 Types of group

Teamwork is defined by James T. Scarnati [6] as "a cooperative process that allows ordinary people to achieve extraordinary results". Extraordinary results is a reference to the fact that we can observe different types of group work [7] which depends on the subject treated by participants, the participants themselves, and finally the participants interactions. A group performance is ruled by these aspects, which we will detail in the next section. David W. Johnson and Roger T. Johnson split groups in four different categories:

- Pseudo Group
- Traditional Group
- Cooperative Group
- High-Performance Cooperative Group

2.2.2 Pseudo Group

Pseudo groups can be defined as students being assigned a task, where participants do not have a deep interest in the task they have to solve. Each one think that their grade depends on their rank as a performer in the group. Hence their strategies is to hide information from other participants, as well as getting as much credits from the work they have done as possible. Participants do not completely trust each other and the amount of collaboration is very low. The outcome of such a group work is a loss of potential and poor results from each participants. The final produced work is mediocre compared to what each students could have done on their own. [7]

2.2.3 Traditional Group

A traditional group is defined as participants being assigned a task they have to solve together, and each one of them accept that they have to do so. The grade is structured in a way that rewards them as individuals and not as a part of a whole. It causes members to seek each other's information without sharing what could be relevant information on their own [3], hence improving other participants work. This collaborative climate implies some members to look for a free-ride in the group work, influencing group mates with an unfair feeling and causing them to do less than they would have done otherwise. Such a climate in group work makes the result less than what the most hard working students could have performed on their own, but higher than the sum of the group. [7]

2.2.4 Cooperative Group

The context of this group configuration is similar to the Traditional Group definition. A group of participant are assigned a task to solve together but their perspective is different. The students know that this task is indeed a shared goal to reach, hence they share information that could benefit every members of the group [8]. Each person is encouraged to help their group mate and to contribute to the work, but also to the success of all members in this group work. The individual performance and outcome is often measured and checked to maintain a hard-working climate within the group. The outcome of such a configuration is that the result is better than what every member of the group would have produced on their own. [7]

2.2.5 High-Performance Cooperative Group

This group is very similar to a classic Cooperative Group, to the exception that the commitment of each member to the goals they share is a lot deeper [3]. The participants are deeply concerned by their group mates performance and success within the group, and do everything in their power to help each other reach their objectives. Such a group satisfies several criteria [3] [8] [7], which go from individual requirements to group collaboration methods.

We have seen in the previous paragraph that different types of group work have been theorized. But what kind of requirements are necessary to be a High-Performance Cooperative Group, and what criteria a collaborative group should satisfy to reach the next level of collaboration and efficiency ?

Research showed that there are several criteria which are key to improve group work collaboration and efficiency [8] :

- Commitment to the team achievement and shared goals
- Interdependence
- Interpersonal Skills
- Open Communication and Positive Feedback
- Appropriate Team Composition

- Commitment to team process, leadership and accountability

2.2.6 Commitment to the team achievement and shared goals

Each member of the group work is committed to see the team achieve its shared goal and wants to produce the best quality work as possible in the allowed time. A successful team is motivated and every participants want more than their own success while achieving the task.

2.2.7 Interdependence

In order to be very efficient, team members need to create an environment that allows them to give more to the group than each member alone could give. This helps to reach the full potential of the group by using the forces of every participants while compensating their individual flaws.

2.2.8 Interpersonal Skills

This point details that for a group work to be successful, the group members need to have technical skills related to the task, but also social skills. It includes communication skills, trust between the members to each other, be supportive and committed to a shared goal, and the ability to compromise between their personal opinion and a shared group decision.

2.2.9 Open Communication and Positive Feedback

A positive work climate benefits every member of a group project, by listening to each other and being open to constructive criticism, participants value each other's contribution, increasing their personal commitment along the way.

2.2.10 Appropriate Team Composition

For a team to develop and exploit its full potential, its composition needs to cover a wide spectrum of skills and knowledge. In that case, all the members can give their best while having point of views from others that may not be as expert in a specific field related to the task. An optimal team composition also includes member to be self conscious of what their role in the group is, and trying to exploit the maximum of what is required from them.

2.2.11 Commitment to team process, leadership and accountability

Effective leadership and shared decision-making is a key part in the success of a team. Each member needs to be aware of the team processes and every step achieved or to complete.

2.3 Human Centered Design

Before starting the development of new features, we wondered what kind of development strategy we should follow, and especially how to test our work ? Since the application prototype was used in a real course by students, teaching assistants and instructors, we decided to catch the opportunity to have the users directly involved in the testing process. To do so we followed a Human Centered Design approach (HCD).

Human Centered Design places the end user of a product at the center of the design process [9]. Therefore, the developed solution suits the real need of users, as they were involved and questioned at the very beginning of the design process. The first phase is to shape personas of users to understand their needs and concerns, this investigative phase let the designers to go in the right direction of development. The HCD method allows to create a deep connection between the product developer and the end-user from the beginning to the end of the development [10], taking into account the users thoughts, emotions, and behavior, it allows to adapt the product to fit perfectly the expectations of the users.

Successful innovation is done by crossing three constraints, viability (from a business point of view), desirability (human needs) and feasibility (from a technical point of view) [11]. HCD methods are a key point to satisfies the user needs. By looking at innovation from a human perspective, the end-user's idea of the product and the developer's actions can be as close as possible, hence it answers most of the needs. One aspect of HCD is user testing, as sometimes users themselves are not sure of what is best for them, putting them in "real life" situations with the product let the developer target what is key in the product characteristics, as well as letting the user specifying its own idea of the product.

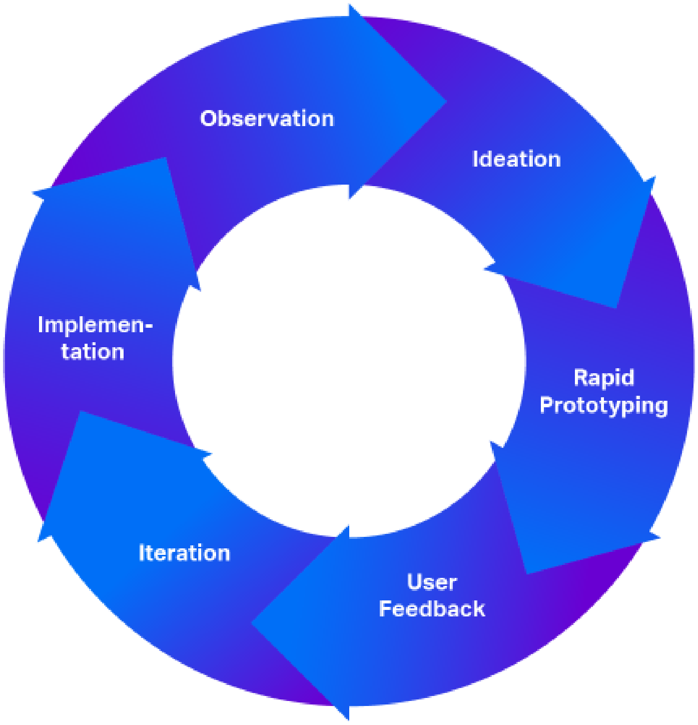


Figure 2.1: Human Centered Design Model

The process to implement Human Centered Design is detailed in [10] and [9] and is structured around the following steps :

2.3.1 Observation

This first step focus on the understanding of the users, their habits and expectations. Even though this part of the work might seem negligible, a true understanding of the user perspective help to make the next steps a lot easier [10] by giving direction to the development problematic. Observing the usage environment of the product as well as the conditions of usage is a key point to start the next step of the process.

2.3.2 Ideation

Clearly defining the limits and purposes of a development helps keeping a crystal clear sight on which problem is solved by the product. The Define part of the Human Centered Design method is conducted by asking "Why?" every choice in the mapping is done [**unified_theory_of_HCD**] [10]. This step is necessary and encouraged in order to build a quick and efficient prototype or concept of the final product.

2.3.3 Rapid Prototyping

After understanding the global environment of usage and the end-user problems, it is time to put ideas into actions by developing several prototypes. Each low-cost model is being tested in simulation to eventually focus on the best prototype. []

2.3.4 User Testing

Once a prototype has been selected, product developers are testing it on users and real situation to identify its flaws and advantages. Because the user tests are part of a Human Centered Design method, it is fundamental not to advocate for the solution but to leave a total freedom to participants as they constructively criticize the prototype.

2.3.5 Iteration

After completing all the previous tasks, some modifications and potential improvements were found. Even though it could seem that the development process is finished, Human Centered Design is often cyclic as it involves several iterations before concluding a final choice of prototype.

2.4 User Testing

In order to set up a Human Centered Design method, Usability Tests were the most appropriate way to gather information on the quality of our work on the application prototype. It also allows us to understand if our features and modifications are relevant for end-users and especially TAs to monitor their group. User Testing or Usability Test is a testing methods that put the user in the middle of the design test. It is often considered that user testing aims to test the user, when in reality the purpose is only to test a design, functionality, or data relevancy. The variety of User Testing goals is wide and it differs regarding what is being tested, but it mostly includes [12][13]:

2.4.1 Identifying Problems

This part aims to uncover problems with a design that users could trigger when using the application. It can be spotting bugs or errors, or check that the application is an appropriate solution to the users needs and expectations. With a wide spectrum of tasks, user testing allows the product developers to gather information on different usage of the prototype by creating scenarios.

2.4.2 Discover Opportunities

This testing methods relies mostly on the user to give feedback while using the application. By performing out loud thinking and critical thinking, the user might have comments and suggestions to make the product more intuitive, add new features or change the data available. After performing the tasks, control questions can help to understand how the user feels about the product and have the user wondering about aspects that he might have forgotten during the test itself.

2.4.3 Learn About Users

By creating scenarios and use cases, the product developers can gather a lot of data on the users habits and way of thinking, in order to adapt it during the next development iteration. The human brain is very complex and each individual might have a different reaction in front of an application, it is hence fundamental to properly understand users behaviors and preferences to build a relevant and efficient product.

User Testing includes three main elements, which are a facilitator, a participant, and a tasks set up toward using the product following a specific scenario.

2.4.4 Facilitator

The facilitator is usually part of the development team and is here to provide the tasks, monitor the tasks performance, and gather data regarding the user critical thinking, behavior and suggestions.

2.4.5 Participant

The participant profile depends on the test purpose, he can be a target user of the product or not according to the point of view that the product developers want to gather data on. Its goal is to perform the provided tasks without any help from the facilitator while thinking out loud and giving constructive feedback.

2.4.6 Tasks

Tasks are provided by the facilitator and aim to follow a specific scenario or usecase of the application. Depending on what developers want to test, several different tasks scenarios can be developed and each one will give a variety of information regarding, intuitiveness, simplicity, or even features relevancy.

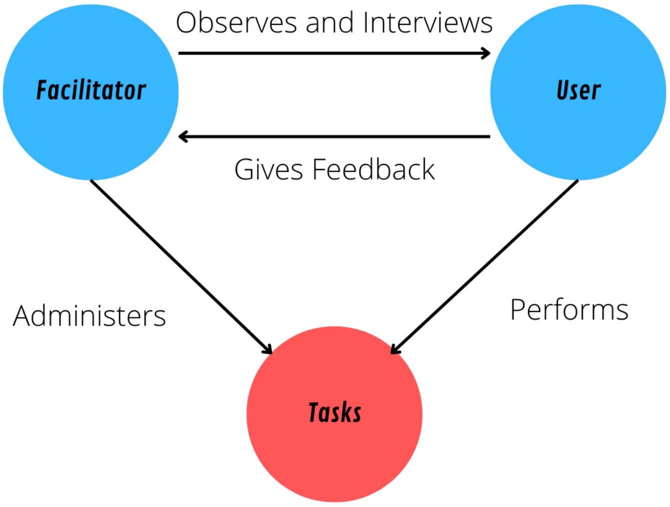


Figure 2.2: User Testing Model

2.5 Agile Methods

Agile methodologies is a type of project management method that aims to break a project into several smaller phases which are easier to handle. Hence, Agile methods tend to work with iterative processes over one large preparation phase followed by one straight development phase [14].

The core values of Agile methods are the following :

- Frequent deliveries

By splitting a project into several smaller sprints, the goal is to be able to deliver a product at the end of each sprints to get feedback from the customer or user.

- Customer-Developer collaboration

Agile methods emphasizes customer-developer collaboration by maintaining a close relationship throughout the whole development process and involving the customer as much as possible.

- Customer satisfaction

As explained in the point above, customer satisfaction is a key part of agile methods, by involving the client in the development process, the product team is able to gather all the customer's remark and concerns and quickly correct the product in consequence.

- Simplicity

Following a mathematical and methodological approach to project management, splitting a long project into smaller phases allows to have a better understanding of the different challenges and difficulties along the way.

- Working software

In the continuity of frequent deliveries, having nearly constant working software makes it a lot easier to test the application throughout the development process, as well as giving a better vision of the final application right from the beginning.

- Motivated teams

Smaller challenges and short term projects give more satisfaction and motivation to the development team, which is very beneficial to the product on a long term sight.

- Changing technical requirements

Working with short term projects allows the development team to review technical requirements on a daily basis, instead of one complete and complex documentation related to the whole project.

- Constant pace

Another aspects of working with several sub-projects is the rhythm, having a well defined and structured plan for each sprint let the developers feel like the next step is always at a short reach. Defining short deadlines of approximately two weeks makes it a lot easier to follow a schedule instead of a several-months long plan where all the delay is accumulated until the end of the project.

Among Agile methods advantages, we can quote better control, increased productivity, higher customer satisfaction, increased flexibility and lower cost. The fact that a potentially shippable product emerge from the end of each sprint, the customer is ahead of the competition and remains in control over the cost of the product, which is usually evolving after each sprint definition.

2.6 Research Biases

When doing research, method is a crucial point to verify the legitimacy of the work done. The human mind is very keen to logical and reasoning biases. It is therefore fundamental to question the methodology to target potential biases interfering when seeking the truth. Different biases are known to influence research and its result, by spotting which bias may apply to your work, one can compensate and adapt the research method to balance the results. [15][16]

Within all the human mind biases, several are keen to influence research work, among these, we can quote the following in respondents or users :

2.6.1 Acquiescence Bias

This behavior bias also known as the friendliness bias happens during user testing or interviewing. In this situation, the subject tends to agree and be positive with whatever the facilitator presents. They are going to act like every suggestion is right and won't tend to show reserve towards the suggestions. This bias also shows that it is usually easier to say 'yes' than to say 'no', which can influence responses to questions within a user test session. [17]

2.6.2 Social Desirability Bias

This bias consist of a behavior change where the participant seek social acceptance by modifying its answers in consequence. People tend to avoid conflict and disagreeable behavior as put one in a delicate and sensitive situation. This can result in a false test report, or at least a different report from the one that could emerge if the participant had no social or moral pressure when doing the tasks.

2.6.3 Habituation

Habituation biases make the participant answer similar responses to similar questions. Thinking and paying attention to detail require a lot of energy, hence our body and mind find it a lot easier to put the mind in "auto-pilot" when we face several similar situation that we know the correct reaction about. [15]

2.6.4 Sponsor Bias

According to the tests type and requirements, it happens that the test participants know the facilitator. Such a situation trigger a classic behavioral reaction from the user to change its answers either to comply with the moderator expectations or on the contrary to go against it. [15]

2.6.5 The Hawthorne Effect

The Hawthorne Effect bias occurs when a participant is observed which can consciously or unconsciously change its behavior in front of the moderator [18]. Some techniques allow to mitigate against this effect, like clearly stating that the test is about the product, not the user, or by creating a very calm and safe atmosphere to let a complete liberty to the participant.

Beside users biases, several others are more keen to influence the researcher itself and are hence worse, as they are more difficult to spot and measure.

2.6.6 Question-order Bias

When it comes to testing a product or doing research, it is extremely important to verify that tests are biases-free. A very common bias when building tests is the question order. Having previous questions may influence the user answers compared to answering each question with no past questions [19]. Therefore, taking care of ordering questions in a way that does not lead to a specific reasoning helps to keep the test bias-proof.

2.6.7 Leading Questions and Wording Bias

Languages are very complex and there are many way of asking the same question. Some might orient the subject toward a specific direction. By choosing the right wording and word order, the facilitator can make sure to influence the least possible the user to gather authentic information [20]. This bias can also be experienced with behavior, facial expression, or answers validation, which would comfort the user in its decision.

2.7 Taking on future work

The previous Master Thesis work we are taking on [1] details a few features and improvements that can be done as future work for next students to work on the application. Those improvements are his own ideas and the results of several user tests. As we focused our work on the TAs and Instructors side of the prototype, we decided not to implement the new features related to the student point of view like :

- "Make a ”Gather group” button. This button should make it possible to call your team into a meeting."
- "Provide the students with the possibility to say a few words about their score when they rate."

This decision made us work on a few other features from August’s master thesis such as :

- "The professor page should view the gap between the scores within a team. This means showing the difference between the person that rate the lowest in average vs the person that rated the highest."
- "The instructor/TA should have the ability to search for groups and apply filters."
- "The instructor should have an overview screen of simple statistics regarding the teams in the course."

Hence, if other students want to take on this work, it may be very interesting to check August’s previous work according to the aspect of the application they want to focus on, as his tests resulted in different information than ours.

2.8 Related work

As of now, several softwares and digital tools are available to help students communicate, collaborate, and share documents. Furthermore, some of them are specialized in school project and offer the possibility to involve teachers in the digital process. The following subsections briefly present what tools are currently available.

2.8.1 BlackBoard

Within NTNU's websites, BlackBoard offers the possibility to share documents and discuss with your group, as well as allowing to contact your teaching assistants by email. BlackBoard has other functionalities such as a group blog, a group wiki and a group journal where students can ask questions about a project in a course [21]. However, few students use BlackBoard as a communication tool as more popular applications are already well established.

2.8.2 Trello

Trello is a project management tool broadly used within companies, but also in higher education. This tool allows a group of students to manage their tasks within different pools, set deadlines, todo list, and set goals. It does not allow communication between members of a group work but its collaborative platform let participants modify every state of the project. A teacher can be included in a project as an observer in order to track the work done, check the progress and leave some comments [22]. This tool is free and hence very accessible for students and universities.

2.8.3 Monday

Monday is also a project management tool mostly used within the companies market. However, this tool offers all the functionalities that a students collaboration tool could require. It offers complex and detailed tasks management functionalities as well as deadline management tools like Gantt diagram. Monday also allow member to communicate with each other, which makes it a more complete tool than Trello. Nevertheless, Monday is a complex tool to use, which makes it challenging for students to dig into. Monday is also not free to use, hence not accessible by every students and universities [23].

2.8.4 Piazza

Piazza is an online Q&A platform for both students and instructors [24]. It allows students to post any question or comments they have and works as a communication platform between the students, the instructors and the teaching assistants.

This web application is an already established tool within NTNU and is being encouraged to use in most of the courses. Usually, Piazza is integrated in Blackboard so that students within each course can very easily enroll, and start using the app. One of the main features of Piazza is the ability to post questions or comments anonymously. This opens up the platform to a lot more people, enabling more activity from students.

2.8.5 Team Compass

Team Compass by Weekdone is a team management software that aims to monitor team performance [25]. It is targeted towards team leaders to provide them with an insight of what is going on with the team. This includes the progression in terms of tasks in a project and their satisfaction and happiness within the team. Their satisfaction levels are evaluated with measures such as team spirit, job satisfaction, and if their are confused. Additionally, the application offers the ability to set and track goals within the team. Team Compass shares a lot of features with both Trello and Monday, and costs about 300 dollars annually. One feature that separates Team Compass from other application we have looked at is the ability to monitor team happiness and satisfaction.

2.8.6 Conclusion

Application	Pros	Cons
BlackBoard	• Team communication tool • Contact Teaching Assistants by email	• Not very popular • Students must reach for help first
Trello	• Project management functionalities • Great collaboration tools • Teachers can track projects	• No communication functionalities • Does not include individual satisfaction toward the group
Monday	• Complete project management functionalities • Communication functionalities for students and teacher	• Complex to manipulate • Not free of use
Piazza	• Q&A features • Anonymity	• Require students to reach out first • No team functionalities
Team Compass	• Satisfaction with the team • Happiness toward the team • Goals tracking	• Not free of use

Table 2.1: Related Work Pros and Cons

Chapter 3

Method

To answer our research question presented in section 1.3 in chapter 1, we constructed this project involving several parts. First, we deployed and tested the state of the application developed during the specialization project in the fall semester. Secondly, we started our literature review regarding the different subjects that our thesis would encounter. To get the maximum out of our readings, we tried to follow the guidelines of literature review as a research method [26]. This helped us to structure our readings while avoiding most research biases [15].

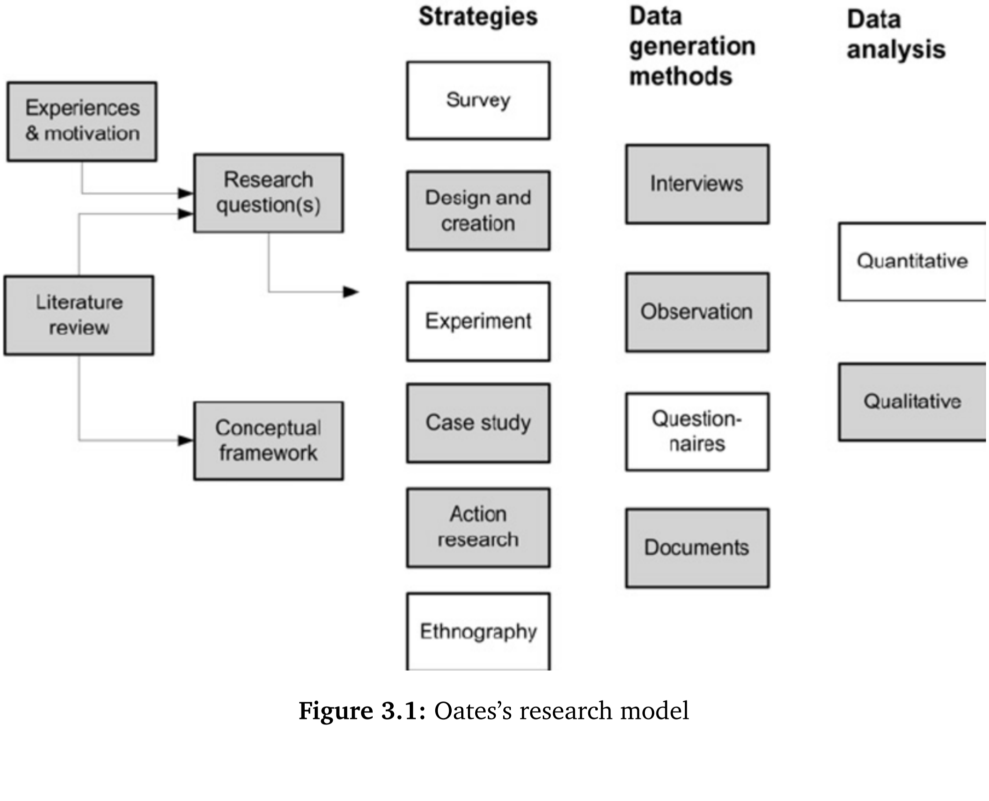


Figure 3.1: Oates's research model

The Oates's research model [27], structured our research methods and played the role of a guideline. The elements that we used according to our field of study and our subject were :

- Experiences and motivation
- Literature review
- Research questions
- Design and creation
- Interviews
- Observations
- Qualitative

Our Human Centered Design method is the main element part of the Design and Creation box, Interviews and Observation were done with our User Testing sessions involving Teaching Assistants, finally, quantitative results emerge from these sessions and gave us information for each new iteration in our cyclic Human Centered Design paradigm of work.

Once the literature review was done, we continued the development of the application, which included two iterations of prototypes as a result of user testing. Chapter 3 contains the methods we used. It explains how we worked during the development of the application. It goes into detail about the conducted user tests, describing the test plan along with the test tasks and control questions. The end of this chapter tells how we got participants for both using the application during the semester, and for the user tests. In the specialization project where the initial state of the application was deployed and set up, only Henrik was a part of this project. For this master thesis, Robin joined and we were now two people working together. This meant spending some time at the start getting to know the code base, along with the architecture and the technologies used.

3.1 Development Method

During the development phase, we have been using agile development methods. We worked on short and delimited development sprints, with the following steps:

- Clearly delineate the purpose of the sprint
- Distribute the workload between us
- Coding the features included in the sprint
- Peer Reviewing what has been done
- Push the sprint's output on the development branch

3.1.1 Git and Issues

As a pair working on this project, we took on the existing Git repositories, respectively for the back-end and front-end. We decided to keep using the versioning tool Git to keep track of the different steps of development and maintain a working version of the app both on the code on production and the code in development using different branches. We used the Gitlab issues feature to cluster our new developments in autonomous and independent parts as much as possible.

3.1.2 Prototypes iteration

Part of doing an agile development process is delivering prototypes frequently, and having iterations of the software that builds upon feedback from users [14]. The initial state of the application when launching the application to the website, ready for students to use at the start of the semester, was developed during the specialization project.

3.1.3 Peer Reviewing

In order to ensure a high code quality during the development part of the Master Thesis, we have been using code peer review methods. It allowed us to have a stable and easily maintainable project during all steps of the development. Both of us had the knowledge and capacity to improve and modify each part of the project from the frontend to the backend.

3.2 User Testing

An important step when designing interfaces is user testing. This is when we get direct feedback from users when they interact with the application. We set up two sessions, with a few weeks in between. Based on feedback from the first session, we worked on the design and prepared for the second iteration. Both times we followed a planned procedure, giving out tasks for the test subjects to perform. After the tasks were complete, we had a conversation to have them further elaborate on their experiences with the design. In the first user test session, we had two TAs from the TDT4180 course. Being active users of the application, they would provide the most valuable feedback as they are in the target group, as well as being familiar with the previous iteration of the design. In the second user test session, we also had two TAs from the course, only this time we had added two test subjects that were not in the course but had relevant experience with being a TA at NTNU. They would still be in our target group, but as a first time looking at the application, bringing new eyes to the design, they might have a different approach than established users.

When performing the user test sessions we first gave the test subjects an introduction, explaining the purpose and goal of the application. A complete description is shown in section 3.2.1. Then we explained the flow of the user test and the purpose of doing such a test. Next, we followed 10 steps for performing a user test session, as described by Tognazzini in his book 'Tog on Interface' [28], which includes teaching the test subject to think out loud. This is an important aspect, as it gives an insight into how the users think when navigating through the application. All 10 steps are presented in section 3.2.2.

3.2.1 Introduction before performing the tests

This was the introduction we gave to the test participants explaining the purpose and goal of the application:

"*Team Accelerator is a web application where instructors and teaching assistants can monitor groups in terms of their health status. It aims to capture and highlight groups that are struggling so that the facilitators can help those groups as early as possible. The overall goal of the application is to improve group work, especially for groups that are struggling. Students within each group are asked to leave a rating every week from 1 to 5 representing their overall satisfaction with the group, where 1 is very unhappy and 5 is very happy. As an instructor or a teaching assistant (TA), you will see a different view of the application than the students. This includes an overview page showing data from all the groups. In this user test, you have the role of a TA of the course and will be presented some tasks to perform. The tasks are meant to represent typical scenarios that occur when using the application. The prototype used in this user test is a fully working state of the application, where we have added some new functionalities and improvements which we want to test. We are not testing you, but rather the user quality of Team Accelerator. Keep in mind that it is still a prototype where bugs might happen. Feel free to comment if experiencing such errors or if anything comes to mind. All feedback is greatly appreciated.*"

3.2.2 Steps when performing the user tests

1. Introduce yourself
2. Describe the reason for this test
3. Tell participants they can cancel at any time
4. Describe what tools/equipment is being used and the limitations of the prototype
5. Teach how to think out loud when doing the test
6. Explain that you cant help them during the test
7. Explain the situation/task and introduce the product
8. Ask if they have any questions and then do the test
9. Wrap up the test by letting the user speak before you connect loose threads
10. Use the result as input to the next iteration of the design

3.2.3 User Test 1

Test plan

- Scope
  The state of Team Accelerator as of April 20th 4.3 viewed from a TA's perspective.
- Purpose
  - Testing the improved Overview page, specifically the addition of the weekly ratings box.

  - Testing the improved list of all teams, including a new sorting feature and a new last login feature.

- Location
  Realfagbygget, Gløshaugen, NTNU
- Session Time
  30-40 minutes
- Equipment used
  A laptop configured with the testing environment, a camera to record the tests, a laptop for notes-taking purposes
- Participants
  Two TA's from the course TDT4180
- Qualitative metrics
  Oral feedback when performing tasks, answers from the control questions
- Quantitative metrics
  Task success rate, time spent during each task
- Roles
  - User : The test participant (TA)
  - Facilitator (Henrik): Explaining how the sessions will be carried out and conducting the users tests
  - Note Taker (Robin): Taking notes of the user out loud thinking and questionnaire's answers

Test tasks

1. You are a teaching assistant in the course TDT4180 and you want to monitor your group's activity. Log in to the app and show the overview page.
2. In group 93, how many students posted a rating during week 8?
3. What was group 93's average rating in group 8?
4. Display all the team members in group 93.
5. Find the groups that have all team members registered in the app.
6. What is the total number of teams in the course TDT4180?
7. When was the last time TA Eva logged into the app?
8. When was the last time TA Frank logged into the app?

Control questions

- What are your immediate thoughts about the design?
- On a scale from 1 to 5, how difficult were these test assignments?
- Did you have any hesitations or difficulties in any of the tasks?
- Were there any features/functionalities that stood out, either in a good way or a bad way?
- How would you describe your overall experience with this app?
- If you could change something in the app, what would it be?

- As a TA in the course where this application is being tested out, have you had a good/bad experience with the app?
- Has any of the teams you are responsible for reported bad ratings and was this acted upon?

3.2.4 User Test 2

Test plan

The test plan for our second user test was mostly similar to the test plan displayed in section 3.2.3. The changes from the previous session are shown below:

- Scope
  The state of Team Accelerator as of May 26th 4.4 viewed from an undass' perspective
- Purpose
  - Testing the new and improved Manage Course page, which includes registering and updating team lists and TA lists.
  - Testing the improved Overview page, including a rearrangement of the boxes and the addition of statistics in terms of ratings.
  - Testing the new Settings feature.
- Participants
  Four in total: Two TAs from the course TDT4180 and two people with previous experience as a TA at NTNU.

Test tasks

1. There have been some changes in the groups in this course, and you are asked to update this in the application. Upload the new list of teams (The updated .csv-file is located in the Desktop-folder)
2. A new teaching assistant (TA) has been assigned to new groups. Upload the new list of TAs. (The updated .csv-file is located in the Desktop folder)
3. How many students registered a rating this week?
4. What was the average rating for group 39 in week 9?
5. The instructor of the course asks if you could send him a list of all the ratings in week 9. Export a .csv file of such a list.
6. You are not interested in seeing all the weekly ratings on the overview page. Hide this view.
7. Out of all students, how many of them have registered in the app?

Control questions

- What are your immediate thoughts about the design?
- On a scale from 1 to 5, how difficult were these test assignments?
- Did you have any hesitations or difficulties in any of the tasks?

- Were there any features/functionalities that stood out, either in a good way or a bad way?
- How would you describe your overall experience with this app?
- If you could change something in the app, what would it be?
- Most important data/feature to have to monitor a group working on a project?
- What would you say could be the value of such an app at NTNU?

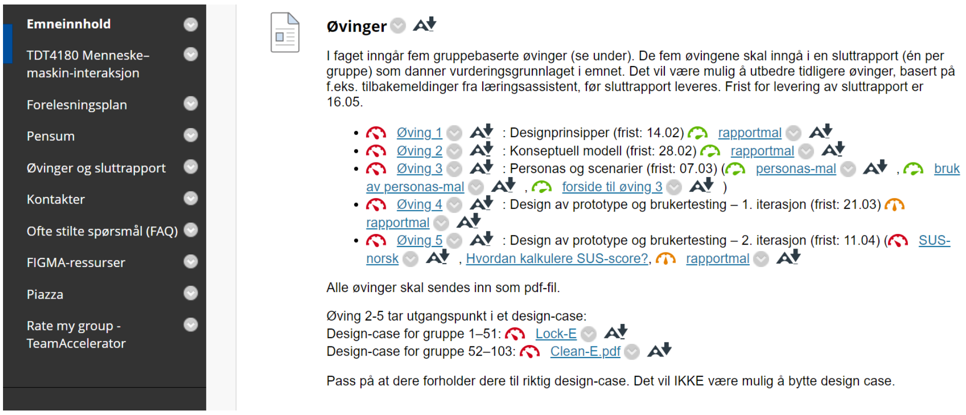
3.3 Getting participants

Getting people to participate in any voluntary exercise or experiment can be difficult. As mentioned in August's thesis, if using the application is not mandatory then students tend to not do extra work if not being rewarded somehow. A measure that was done to achieve this was rewarding 3 students each with a 500kr gift card. Every student in the course TDT4180 that had more than 8 ratings was eligible for winning a gift card. This would mean they needed to log in to the app at least once a week for 8 weeks. Out of the eligible students, the winning students were chosen randomly.

Similarly, to get participants for the user tests, we handed out gift cards with a value of 200kr for each TA that would participate.

3.3.1 Link to Team Accelerator in Blackboard

To make the application more accessible for the students, a link to Team Accelerator was added to the course page of Blackboard. The link can be seen in the navigation bar on the left side of the screen in figure 3.2 below. Furthermore, several reminders were sent to students by instructors of the course in order to encourage them to rate their group each week.


Emneinnhold
TDT4180 Menneske-maskin-interaksjon
Forelesningsplan
Pensum
Øvinger og sluttrapport
Kontakter
Ofte stilte spørsmål (FAQ)
FIGMA-ressurser
Piazza
Rate my group - TeamAccelerator

Øvinger

I faget inngår fem gruppebaserte øvinger (se under). De fem øvingene skal inngå i en sluttrapport (én per gruppe) som danner vurderingsgrunnlaget i emnet. Det vil være mulig å utbedre tidligere øvinger, basert på f.eks. tilbakemeldinger fra læringsassistent, før sluttrapport leveres. Frist for levering av sluttrapport er 16.05.

- Øving 1 : Designprinsipper (frist: 14.02) rapportmal
- Øving 2 : Konseptuell modell (frist: 28.02) rapportmal
- Øving 3 : Personas og scenarier (frist: 07.03) (personas-mal , bruk av personas-mal , forside til øving 3)
- Øving 4 : Design av prototype og brukertesting – 1. iterasjon (frist: 21.03) rapportmal
- Øving 5 : Design av prototype og brukertesting – 2. iterasjon (frist: 11.04) (SUS-norsk , Hvordan kalkulere SUS-score?, rapportmal

Alle øvinger skal sendes inn som pdf-fil.

Øving 2-5 tar utgangspunkt i et design-case:
Design-case for gruppe 1–51: Lock-E
Design-case for gruppe 52–103: Clean-E.pdf

Pass på at dere forholder dere til riktig design-case. Det vil IKKE være mulig å bytte design case.

Figure 3.2: Link to Team Accelerator in Blackboard

3.4 Privacy policy

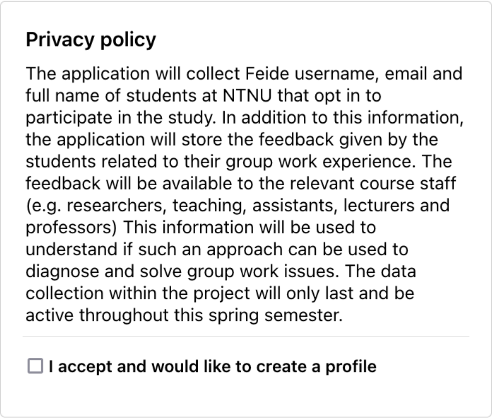
3.4.1 NSD

When gathering real-world data, it is necessary and important to make sure we are following all guidelines to protect the people who participate. Before the semester started, we sent an application to NSD explaining the project, and what type of data we would gather. For the students registering and using Team Accelerator, we only gather their names and email, and their weekly ratings which are only visible to the instructors and TAs. No students can see the individual ratings, only their group's total average. At least 3 or more ratings need to be registered within a group for this average to show.

When doing the user tests, we used a video camera from NTNU to film the process. We recorded both audio and video, which were both specified in the application to NSD.

3.4.2 In-App

Every user that logs in to the application for the first time is presented with a privacy policy explaining the type of data that will be collected. It explains who the data is available for and that it will only be available for the spring semester. The user needs to accept this privacy policy and its conditions to be able to make a profile. If they do not accept, they are not registered as a user and are not able to leave any data. The policy is shown in figure 3.3 below.



Privacy policy

The application will collect Feide username, email and full name of students at NTNU that opt in to participate in the study. In addition to this information, the application will store the feedback given by the students related to their group work experience. The feedback will be available to the relevant course staff (e.g. researchers, teaching, assistants, lecturers and professors) This information will be used to understand if such an approach can be used to diagnose and solve group work issues. The data collection within the project will only last and be active throughout this spring semester.

☐ **I accept and would like to create a profile**

Figure 3.3: Privacy policy when first logging in

Chapter 4

Prototype

This chapter includes all the iterations of the prototype of Team Accelerator. First, it describes the initial state of the application when it was deployed at the start of the semester. Then, it describes the changes that were made before we had our first user test session. Lastly, it describes the changes that were made between the first and the second user test session, based on the feedback we received from the first session. It includes screenshots of the added features along with a description that goes into detail on what has been done. The main focus of development in this project was from the instructor's and the TA's perspectives. As the previous master project [1] focused most on the student's view and functionalities, it was a natural step to improve the other sides of the application. As we were able to gather a lot of data from students, the visualization of this data could be greatly improved from the current state of the app. All the iterations of the prototype were fully implemented on a development branch in both the backend and the frontend. The application that was being tested in the course at NTNU was only in the initial state.

4.1 Architecture

This section describes the architecture of the application Team Accelerator. First, subsection 4.1.1 provides an overview of the overall architecture at a higher level. Subsection 4.1.2 and 4.1.3 goes more into detail of both the backend and the frontend architecture respectively. Then, subsection 4.1.4 describes the database setup including a database model. Lastly, subsection 4.1.5 explains how this project was set up and deployed before the start of the spring semester.

4.1.1 Overall Architecture

The application is built following a RESTful architecture style. It consists of a client, a server, and a database structure. All messages between the client and the server are sent over HTTP and follow a set of constraints. When the client sends a valid request to the server API, the API returns the requested resource statelessly.

All communication between the client and the server is separated and independent of each other. The data is sent as readable JSON objects, which makes parsing easy and compatible with any programming language. The client and the server are developed independently and are separated, making the application structure more flexible and open to other types of clients in the future. For authentication, the client sends a request to the FEIDE-API, which returns an authentication token to the client, which then can be sent to the server. The overall architecture can be seen in figure 4.1 below.

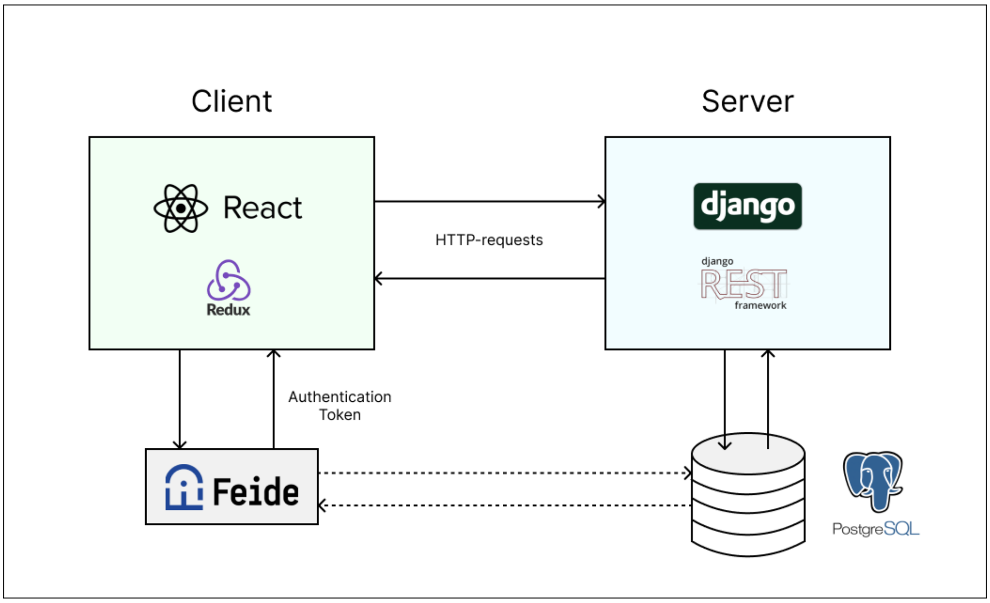


Figure 4.1: Overall architecture

4.1.2 Backend Architecture

As mentioned in subsection 4.1.1, the application follows a RESTful architecture style. The constraints for following such a style are defined and implemented in the backend. When a client sends a request to the server, it comes as a type of HTTP request, either a GET-, a POST- or a PUT request. The API receives the request as a URL and can be a more general request, such as fetching the home page of the application, or a more specific request containing an ID. All requests are handled by class-based views that serialize the processed data to JSON, then return the data to the client. Django creates a model of the database, where it converts python code to SQL to read or write to the database. This happens all under the hood as part of Django. The database configurations are specified in the Django settings file. The architecture for the backend is illustrated in figure 4.2 below.

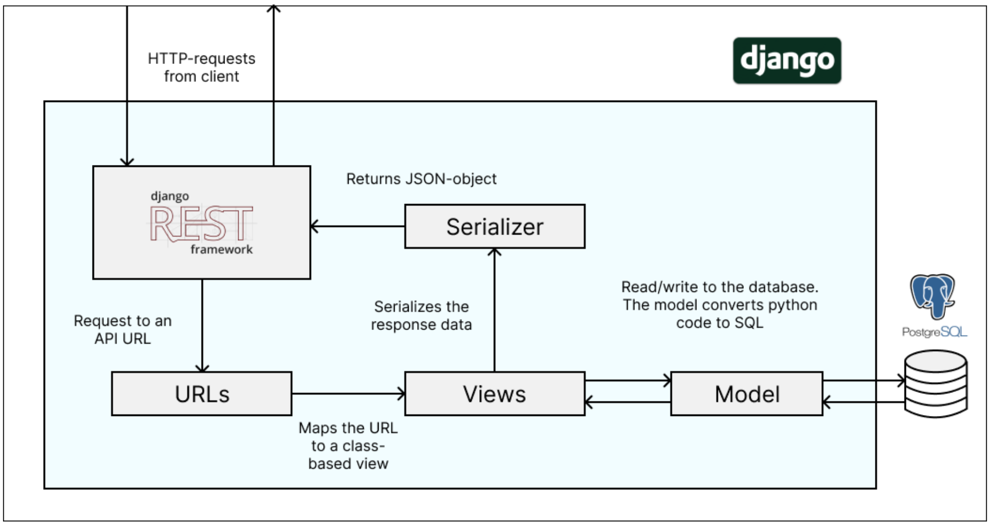


Figure 4.2: Backend architecture

4.1.3 Frontend Architecture

The frontend architecture of Team Accelerator is built using React and is therefore component-based. Every page and feature of the application is a component that manages its own state. When the state of the component is changed, only the component that has the relevant data that needs updating will re-render. This makes the code more efficient and easier to debug.

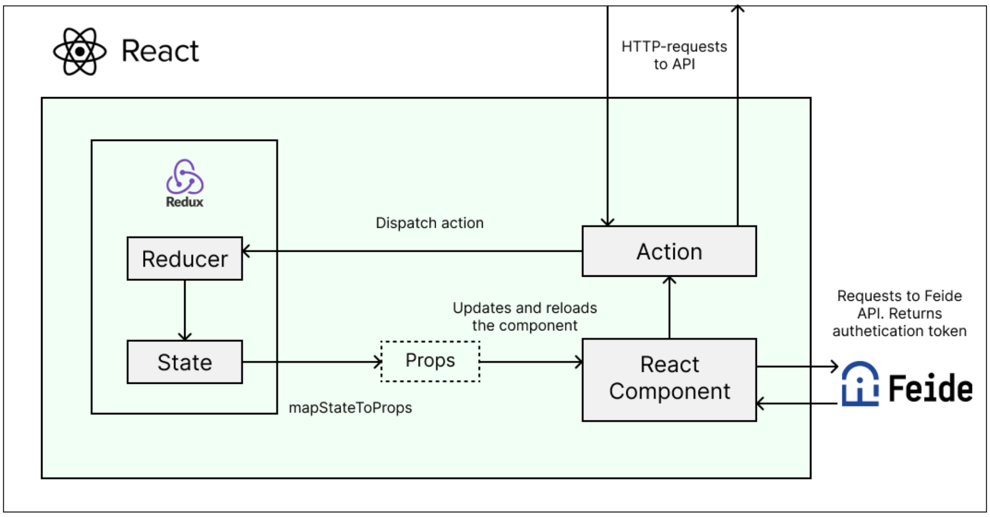


Figure 4.3: Frontend architecture

4.1.4 Database Architecture

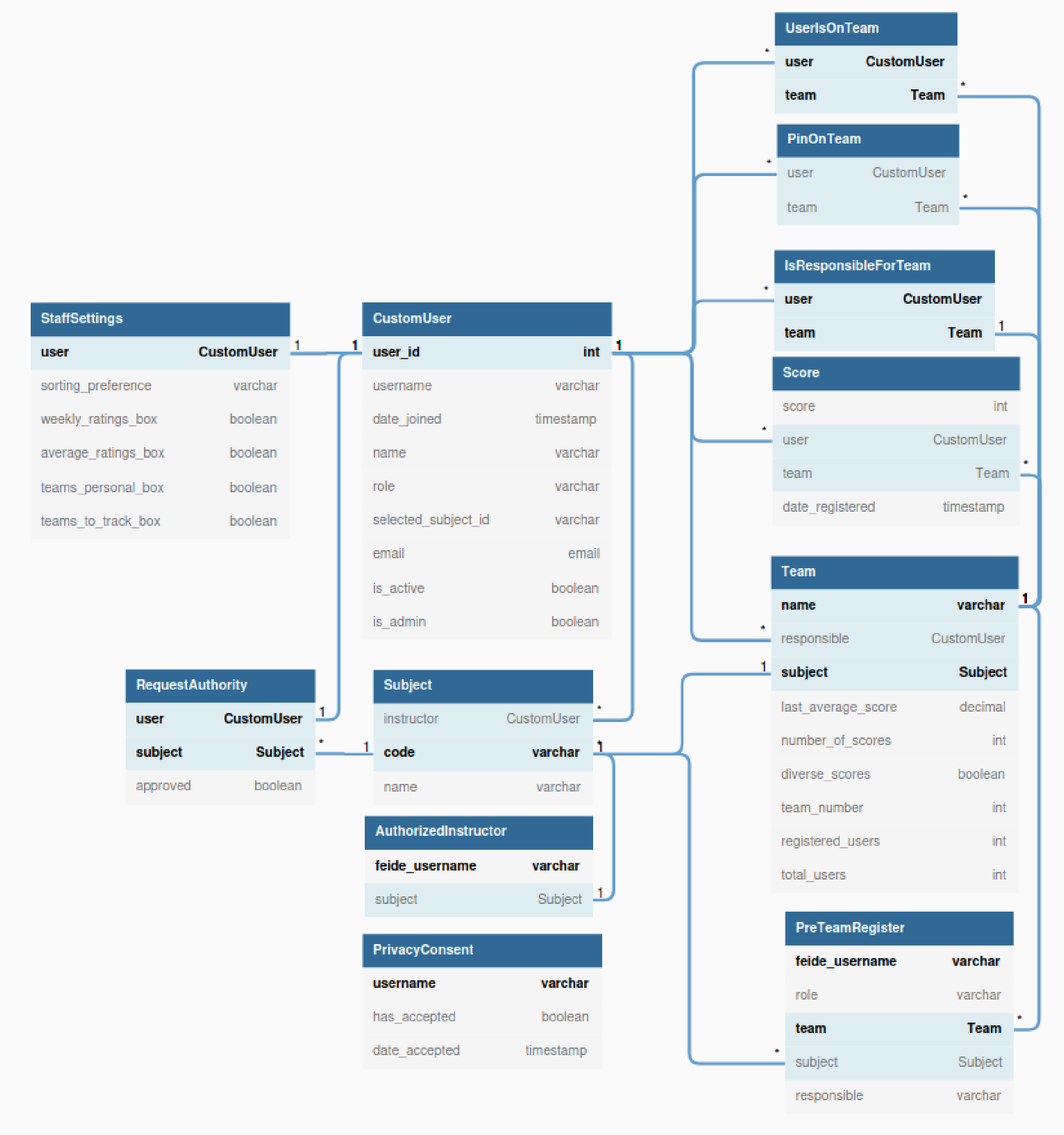


Figure 4.4: Database Architecture

Most of the database structure was already implemented in the initial state of the application. We tried to only add what was necessary to keep it as simple as possible, by processing the data from the backend into frontend readable information. We added the StaffSettings table which was necessary to store the configuration information of each TA and Instructor.

4.1.5 Project setup

One of the major drawbacks due to the coronavirus during the previous master project [1], was the lack of real-world testing in a course at NTNU. This year, students were back at school and we had a ready application to test, to gather real-world data. Before the semester started, the application needed to be up and running so that when the website was announced, the students could go to the website without any trouble. To have the application up and running, it first needed to be deployed for production. The backend was deployed using Gunicorn, a Python WSGI server for UNIX. The frontend was deployed using 'serve', and it creates a static build folder that the backend reads. Both the backend and the frontend were hosted on a server at NTNU, specifically a virtual machine running Ubuntu. Two domains had already been reserved, one for the backend, and one for the frontend. To redirect the traffic from the domains to the production processes on the virtual machine, configuration files for Apache2 were set up.

One of the features of the application is the preregistration of a course which was improved from the previous version. A list of students with their FEIDE username and their corresponding group number was uploaded using the Manage Course section. A list of TAs with their responsible groups was also uploaded. The preregistration would mean that when the students and the TAs first logged in to the application, they would immediately get assigned to their group and could start using the app straight away.

The importance of having a smooth experience at the beginning of the semester was crucial. If students had a bad experience where the application crashes and is not functioning, they might not continue using it during the rest of the semester. Therefore, monitoring the traffic and state of the application was highly important, especially at the start, but also during the entire semester.

Backup of data

To secure the data we took daily backups of the database. In case anything would happen to the server where both the application and the database archive run, we could at any day restore the database. The database system we used, PostgreSQL, has built-in functionality for taking a dump of the database. These daily dump files were stored in a different location than the VM at the server, in case this VM would crash and had to be reset.

Exporting weekly ratings

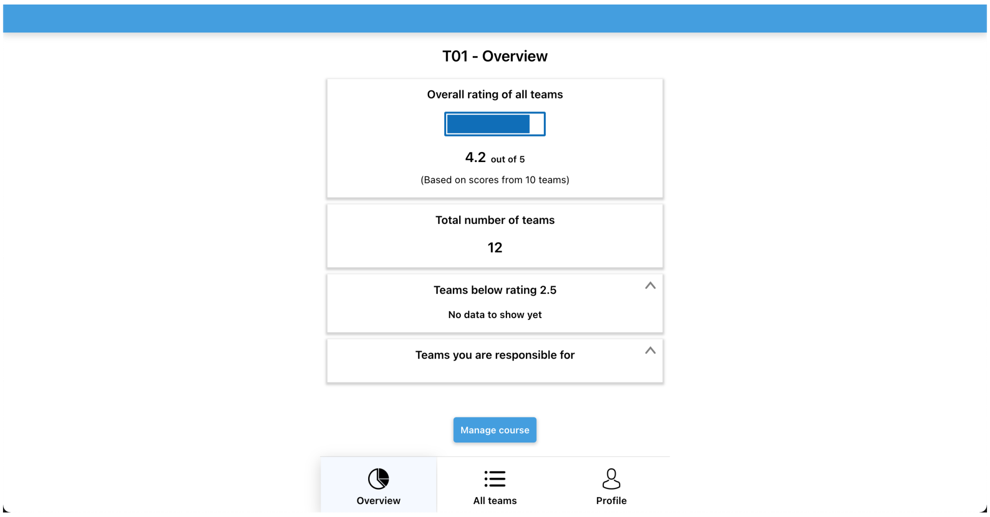
To provide a better overview of all the ratings that were registered by the students, every week we exported a .csv-file with a list of ratings. The list contained the username, the rating registered, their team number, and the date. By gathering the data in a weekly overview, the instructor of the course could get a better view of the data. As the list was exported as a .csv-file, the data could easily be integrated with other applications, such as Excel.

4.2 Initial State

The initial state of Team Accelerator of this master project was developed during the specialization project in the fall semester. It was the state of the application that was deployed and ready to be used by students of the course TDT4180 in January. The changes that were made compared to the original prototype from the previous master thesis [1] was about the registration process of teams and TAs. When registering a new course to be used in Team Accelerator, you need to upload a .csv-file containing the list of all teams with Feide-usernames. The first improvement made in the specialization project included handling changes in such a .csv-file. Now, the application allowed a re-upload of such a .csv-file, where all the changes got applied, either in terms of students changing teams, or students dropping out (removed from the list). Additionally, a similar way of adding TAs with their corresponding teams they would be responsible for was implemented. Before this, you would need to use the Django Backend Panel, which meant connecting directly to the server API. Now, this was possible in a similar manner as with team upload, within the client-side of Team Accelerator. This feature can be seen in figure 4.7 and is identical to the Teams upload page. Instead of being directly sent to a page to upload a team list, the button Manage Course shown in figure 4.5, directs the user to a new page seen in figure 4.6. Then the user can choose between managing either teams or TAs.

Another feature that was added during the specialization project was manually uploading either teams or TAs directly into the application. This can be useful when only adding one or a few team members to the course, without the need of uploading any .csv-file. This is shown in figure 4.8

The interface of the initial state of the application was mostly unchanged and the Overview page, which is the first page an instructor or a TA sees after logging in, can be seen in figure 4.5 below.



T01 - Overview

Overall rating of all teams

4.2 out of 5

(Based on scores from 10 teams)

Total number of teams

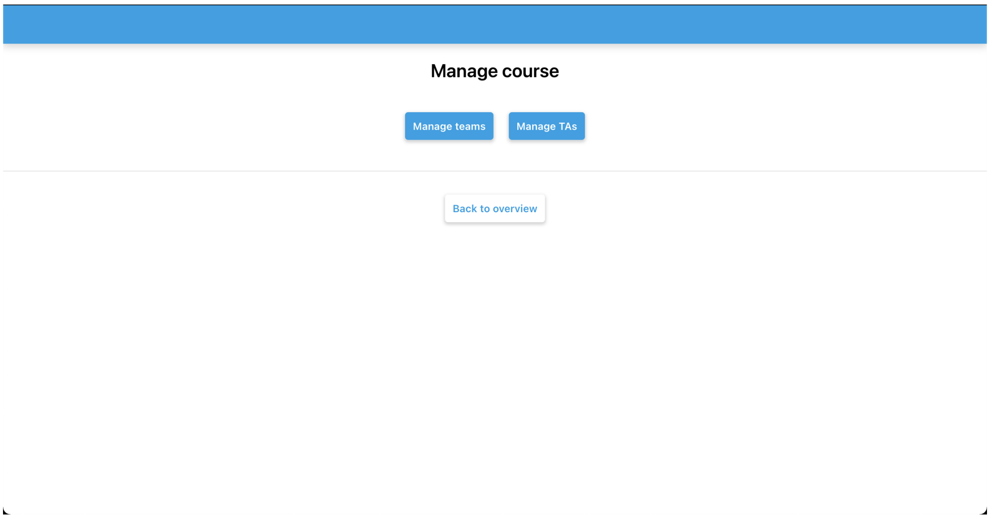
12

Teams below rating 2.5

No data to show yet

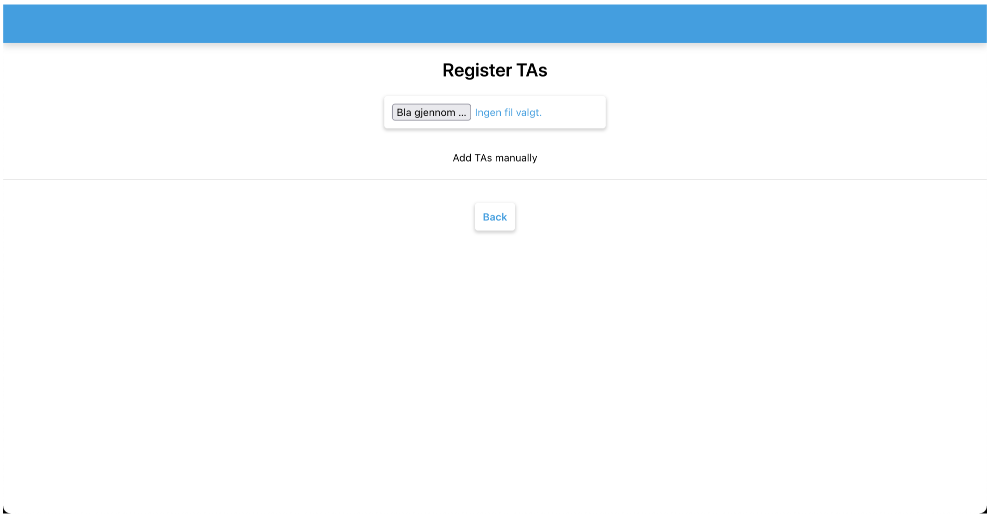
Teams you are responsible for

Figure 4.5: Overview page



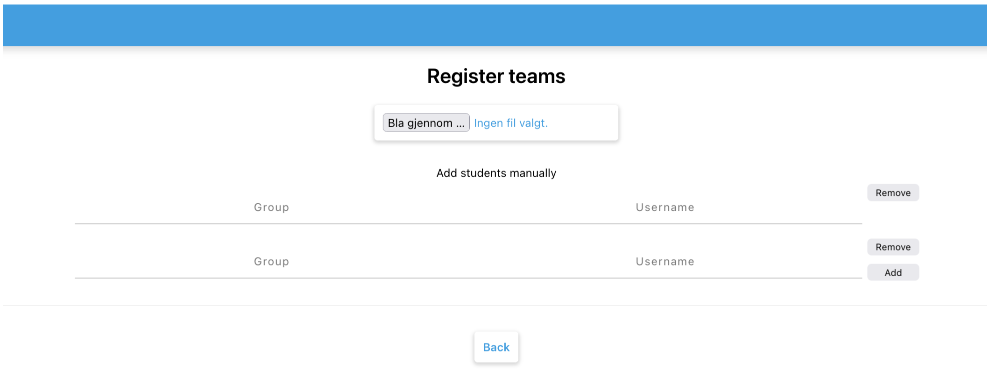
Manage course

Figure 4.6: Manage Course



Register TAs

Figure 4.7: Upload of TA's



Register teams

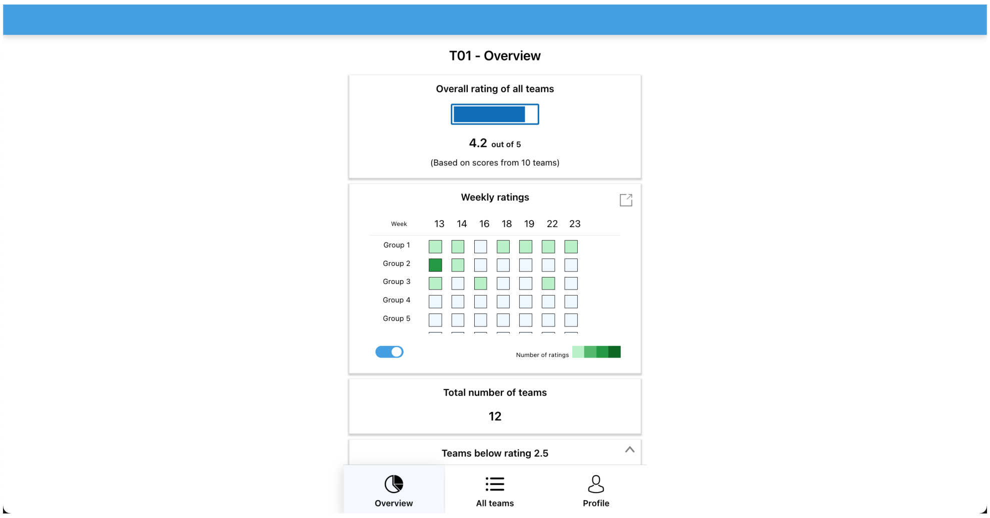
Figure 4.8: Manual upload

4.3 First Iteration

As mentioned in the introduction of this chapter, our main focus for development in this master project was from an instructor or a TA's perspective. The initial state included improvements in the registration process and management of team lists and TA lists. However, it did not see any changes or improvements on the Overview page. Being the homepage for monitors of the course and acting as a dashboard to visualize data, this became one of our main focuses to improve. Thus, the next phase of development focused on the visualization of data on the Overview page. In the initial state, it has a modular view containing boxes of information. This includes an overall rating for all teams of the entire semester as a whole, as well as the number of teams. It has a box containing a list team that has a below 2.5 average rating when clicked if any. It also has a box of the teams the user logged in to the app is responsible for. For instructors that have no specific teams to monitor, it is empty. Overall, it gives a summary of all teams for the duration of the entire semester, and not so much detailed insight into the data.

4.3.1 Weekly Ratings

To get a more nuanced view of the data in the Overview section, we implemented a new weekly view of the data. It is a heat map showing week-by-week all the teams and their corresponding activities. Either in terms of the number of ratings or the average rating for each team. Changing between these two views is done using a toggle switch. This allows the monitors of the course to more easily see statistics in the current week, which is more relevant than only showing the overall statistic. It also shows the progress within each group as a timeline throughout the entire length of the semester. The weekly ratings followed the design conventions from earlier, using a box that contained the information. To fit the box with the others, only the last 7 weeks are visible. If you would want to see the entire semester including all weeks, a pop-up view is possible using the button in the top-right corner of the box. This view is shown in figure 4.10. When hovering over any of the small boxes within the 'Weekly Ratings' view, a detailed value appears giving the exact value. If you click on one of them, you get directed to the More Details view for that specific team. The coloring of the heat map works as follows. For the number of ratings, the number of ratings within each week is divided by the number of students registered within the same team. A higher number results in a darker green color. This accounts for different group sizes and gives a more realistic result. The average ratings are divided into red, yellow, and green colors. If the average rating for a group within any given week is 2 or less, it results in red color. An average rating of 3 gives a yellow color, and a rating of 4 or more gives a green color. These two views combined can give a much more detailed view of the data, providing information about what teams are registering their ratings, and not only how they are doing in the current week, but also their progression over the duration of the entire semester. The red colors in the heat map would highlight groups that are struggling, which is the main goal of this app. The new Overview page can be seen in figure 4.9 below. Figure 4.11 shows what happens when the toggle switch is pressed.



T01 - Overview

Overall rating of all teams

4.2 out of 5

(Based on scores from 10 teams)

Weekly ratings

Week	13	14	16	18	19	22	23
Group 1	🟩	🟩	□	🟩	🟩	🟩	🟩
Group 2	🟩	🟩	□	□	□	□	□
Group 3	🟩	□	🟩	□	□	🟩	□
Group 4	□	□	□	□	□	□	□
Group 5	□	□	□	□	□	□	□

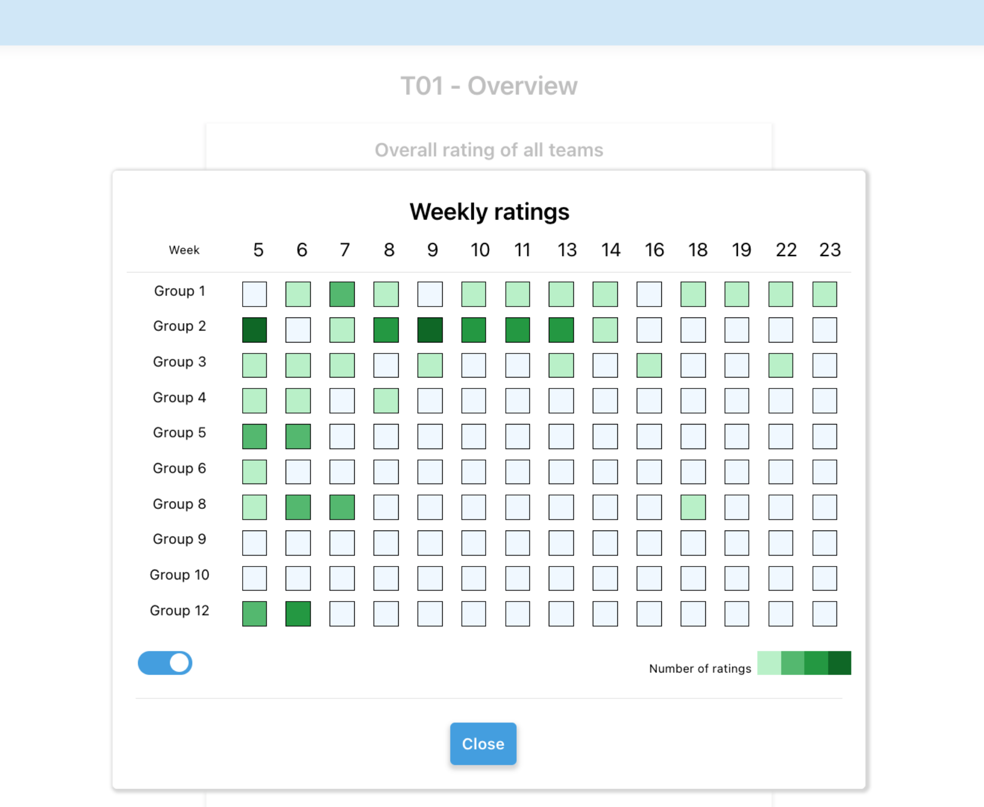
Number of ratings

Total number of teams

12

Teams below rating 2.5

Figure 4.9: Overview page



T01 - Overview

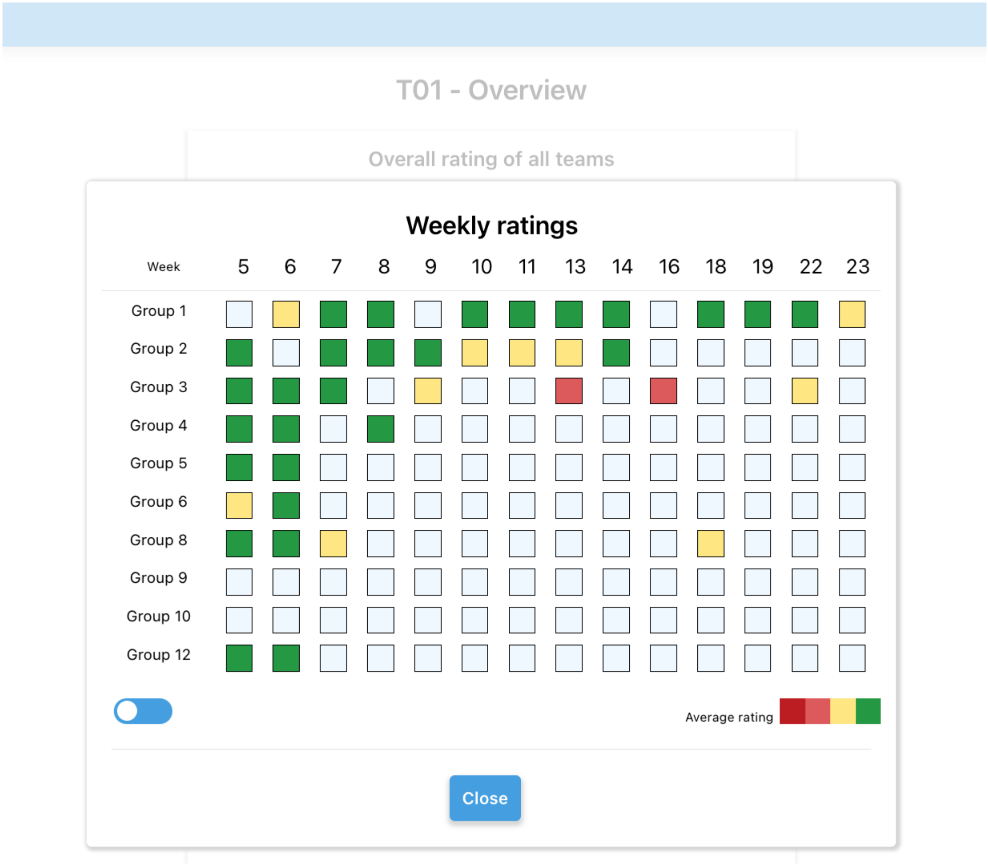
Overall rating of all teams

Weekly ratings

Week	5	6	7	8	9	10	11	13	14	16	18	19	22	23
Group 1	□	🟩	🟩	🟩	□	🟩	🟩	🟩	🟩	□	🟩	🟩	🟩	🟩
Group 2	🟩	□	🟩	🟩	🟩	🟩	🟩	🟩	🟩	□	□	□	□	□
Group 3	🟩	🟩	🟩	□	🟩	□	□	🟩	□	🟩	□	□	🟩	□
Group 4	🟩	🟩	□	🟩	□	□	□	□	□	□	□	□	□	□
Group 5	🟩	🟩	□	□	□	□	□	□	□	□	□	□	□	□
Group 6	🟩	□	□	□	□	□	□	□	□	□	□	□	□	□
Group 8	🟩	🟩	🟩	□	□	□	□	□	□	□	🟩	□	□	□
Group 9	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Group 10	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Group 12	🟩	🟩	□	□	□	□	□	□	□	□	□	□	□	□

Number of ratings

Figure 4.10: Weekly Ratings Pop-up



T01 - Overview

Overall rating of all teams

Weekly ratings

Week	5	6	7	8	9	10	11	13	14	16	18	19	22	23
Group 1	□	🟨	🟩	🟩	□	🟩	🟩	🟩	🟩	□	🟩	🟩	🟩	🟨
Group 2	🟩	□	🟩	🟩	🟩	🟨	🟨	🟨	🟩	□	□	□	□	□
Group 3	🟩	🟩	🟩	□	🟨	□	□	🟥	□	🟥	□	□	🟨	□
Group 4	🟩	🟩	□	🟩	□	□	□	□	□	□	□	□	□	□
Group 5	🟩	🟩	□	□	□	□	□	□	□	□	□	□	□	□
Group 6	🟨	🟩	□	□	□	□	□	□	□	□	□	□	□	□
Group 8	🟩	🟩	🟨	□	□	□	□	□	□	□	🟨	□	□	□
Group 9	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Group 10	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Group 12	🟩	🟩	□	□	□	□	□	□	□	□	□	□	□	□

Average rating 🟥🟥🟨🟩

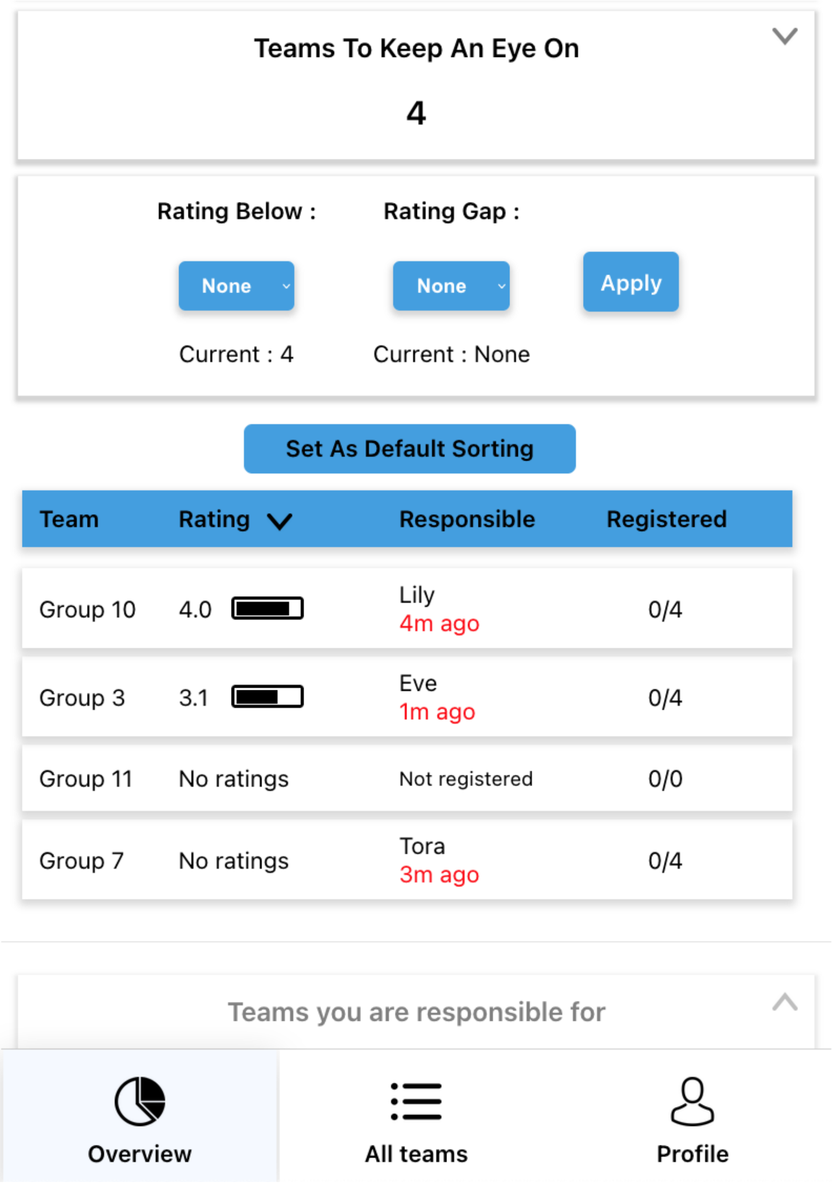
Figure 4.11: Weekly Ratings Average

4.3.2 Teams To Keep An Eye On

Another feature we added to the Overview page was a box we called Teams To Keep An Eye On. Throughout the semester, we gathered more and more data from students. We understood from user tests that the data spectrum available on the application, and especially the overview screen, was too wide and general to be relevant for TAs. Therefore we implemented some new filters to spot the teams that require help the most. Thanks to user tests and the background research that we did, we now allow TAs and instructors to easily track teams according to two conditions

- Teams with an average rating below a chosen value. On one hand, this condition allows instructors, and TAs, to spot very easily teams that are having a hard time with the project. It is therefore quickly accessible to provide them the help they may need to improve their experience in a course.
- Teams with at least 2 ratings with a gap superior to a chosen value On the other hand, since we know that group collaboration is key to the assured quality of the final work, having a significant rating gap within a team is an

indicator of bad group chemistry. Once again, the intervention of a TA could help fix any personal issue, but only if those teams are easily traceable.



Teams To Keep An Eye On

4

Rating Below : None

Current : 4

Rating Gap : None

Current : None

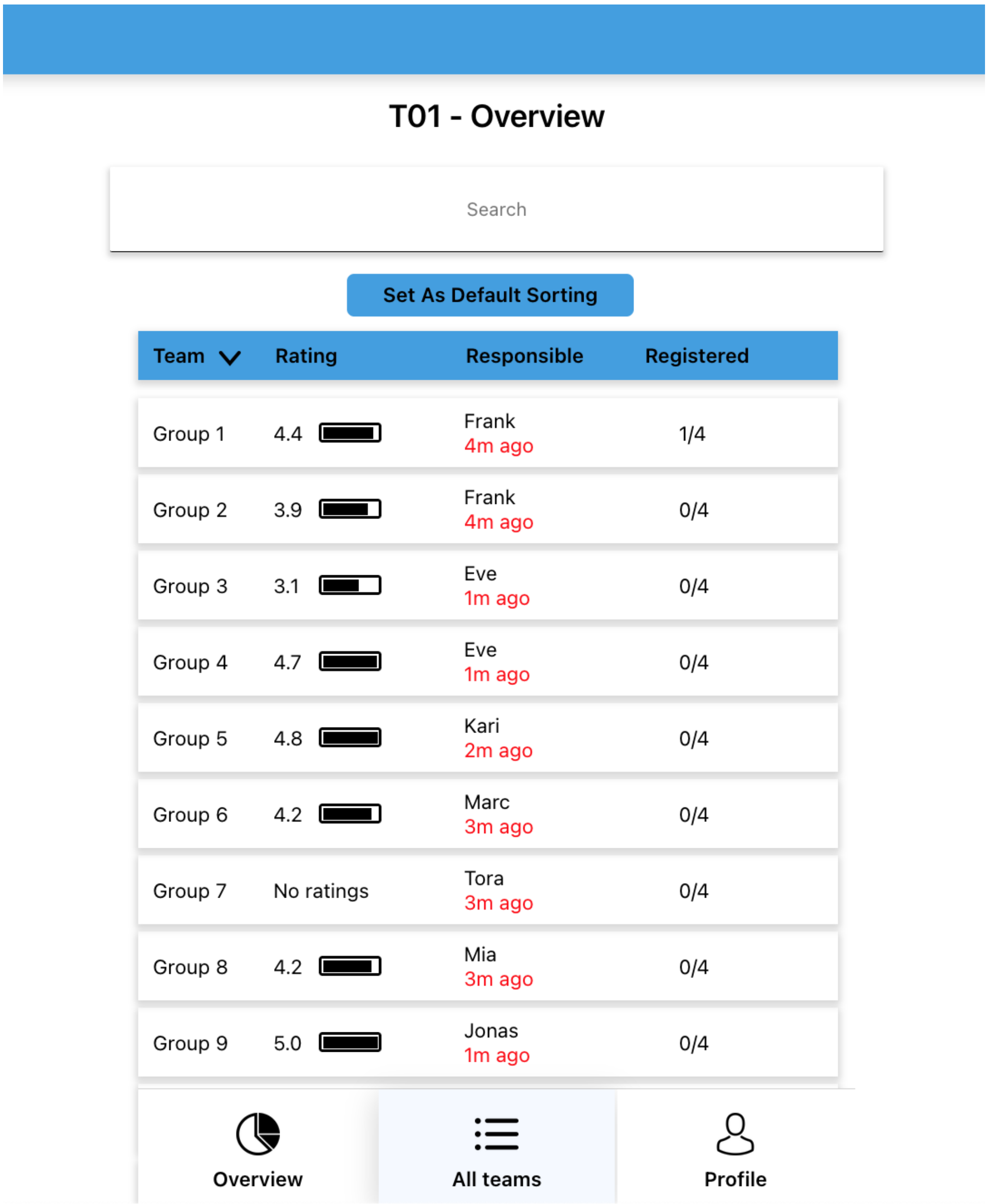
Team	Rating	Responsible	Registered
Group 10	4.0	Lily 4m ago	0/4
Group 3	3.1	Eve 1m ago	0/4
Group 11	No ratings	Not registered	0/0
Group 7	No ratings	Tora 3m ago	0/4

Teams you are responsible for

Figure 4.12: Teams To Keep An Eye On

4.3.3 All Teams

A list containing all the teams registered in the course is shown on the All Teams page. The initial state of the application, includes the team name, their overall rating average, and which TA is responsible for the team. To see more of what is going on for each team, we added another column called 'Registered'. It contains how many students of the groups have registered within the application out of how many students are pre-registered in the course. We implemented the possibility to sort a list of teams according to this data to track teams with little participation in the process of improving group work with the prototype. We wanted to make relevant data as accessible as possible so that TAs and instructors do not give up on the tool. To do so, we added a button to save a default sorting preference for each user, allowing them to get the data they are looking for right after connecting to the application. The initial prototype included a small pop-up view of each team when clicked, which shows a more detailed view of the selected team. To improve the navigation and limit the number of clicks when using this view, we added ’previous’ and ’next’ buttons to quickly browse through all teams without needing to go back to the list view. This can be seen in figure 4.14. Another addition in the first iteration of the prototype, only visible to the instructor and undasses, was a ’last log in’ feature. It shows a small text below the name of the TA describing when each TA logged in to the app. The text is color-coded with a green color for the most recent activity, a yellow color for a bit longer period, and a red color when it has been a while since the TA last used the application. The All Teams page is shown in figure 4.13 below.



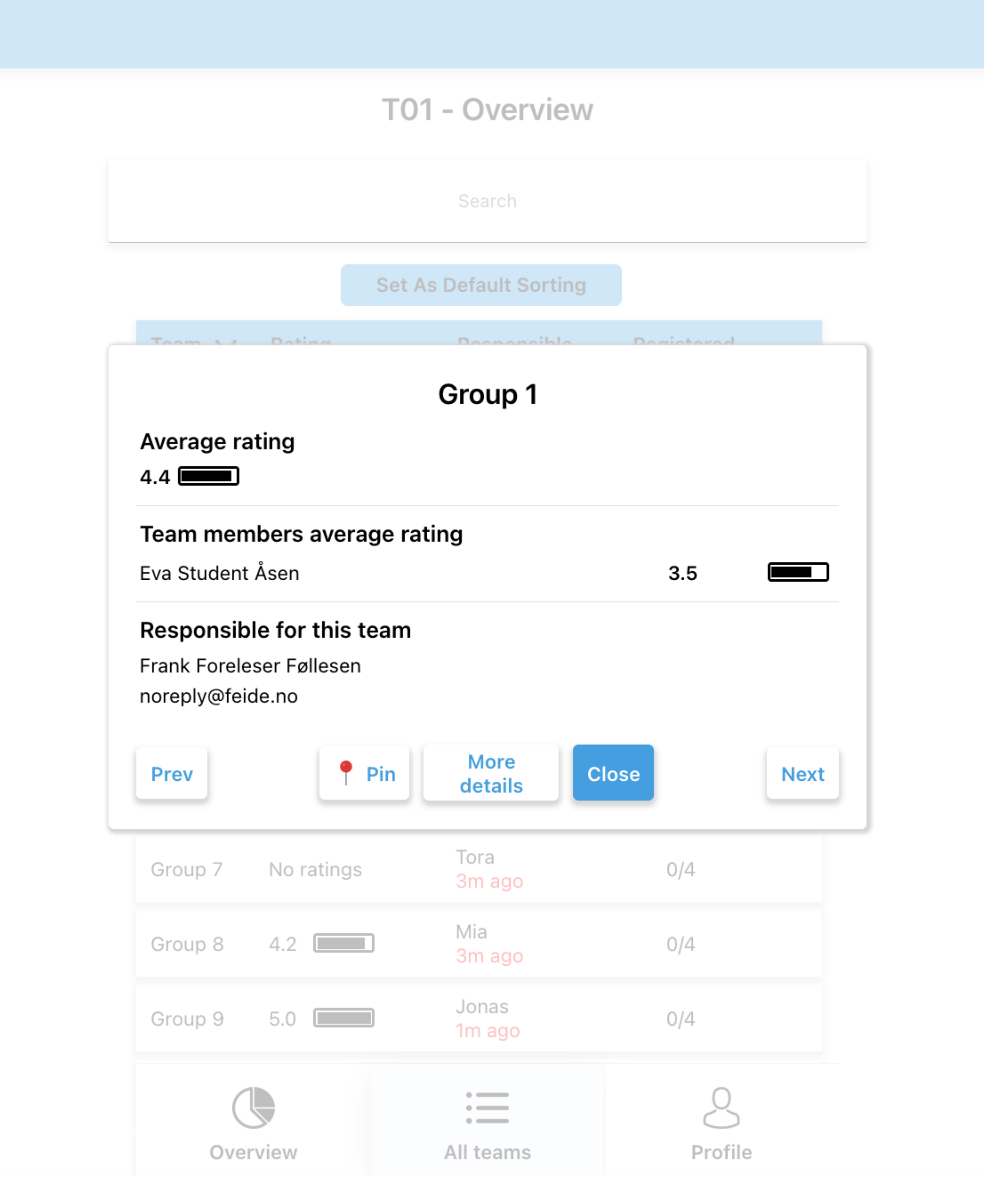
T01 - Overview

Search

Set As Default Sorting

Team	Rating	Responsible	Registered
Group 1	4.4	Frank 4m ago	1/4
Group 2	3.9	Frank 4m ago	0/4
Group 3	3.1	Eve 1m ago	0/4
Group 4	4.7	Eve 1m ago	0/4
Group 5	4.8	Kari 2m ago	0/4
Group 6	4.2	Marc 3m ago	0/4
Group 7	No ratings	Tora 3m ago	0/4
Group 8	4.2	Mia 3m ago	0/4
Group 9	5.0	Jonas 1m ago	0/4

Figure 4.13: All Teams page



Group 1

Average rating
4.4

Team members average rating
Eva Student Åsen 3.5

Responsible for this team
Frank Foreleser Føllesen
noreply@feide.no

Group 7	No ratings	Tora 3m ago	0/4
Group 8	4.2	Mia 3m ago	0/4
Group 9	5.0	Jonas 1m ago	0/4

Figure 4.14: Team details

4.4 Second Iteration

Based on feedback from the first user test session, we started development on the second iteration of Team Accelerator in this master project. As mentioned in the previous thesis [1], this application was tailored to mobile users. As the main focus then was on students, having a mobile-friendly interface is especially important. The content of the student side of the application does not contain that much information and is suitable for smaller screens. For instructors and TAs registering and monitoring the course, it is usually done using a larger monitor on a PC. The current mobile-tailored interface results therefore in a large unused white space on both sides of the content. Along with the expansion of the Overview page, a more responsive web application was needed.

4.4.1 Overview Page

First, the ’Weekly Ratings’-box was expanded. This would allow for a view that would fit all the weeks in a semester as a standard, discarding the need for a pop-up view. Additionally, a label for the toggle switch was added to make the functionality clear to the user. To complement the ’Weekly Ratings’ heat map, two additional boxes were added showing the data from the current week compared to the previous week. This further highlights more relevant information about the current state of the course. The first box contains the weekly averages of all teams, both in the current week and the previous week. The previous week is colored gray, making the current week in blue more visible, directing the eyes of the user to what is most relevant. The second box shows the number of ratings. Similarly, the data for the previous week is colored in gray, while the current week gets pushed out in a stronger black color. Here, the total number of teams is also shown, making an entire box for this information obsolete, creating more space for the new features added.

By expanding the content of the Overview page when using a larger screen, users might still be using the application on a smaller screen. By using CSS media queries with defined breakpoints, the page would now respond to shrinking the width, making the page utilize the space a lot better, regardless of screen size. For the weekly ratings on a smaller screen, only the last 7 weeks would appear. The Overview page using a PC monitor and a mobile phone is shown in figure 4.15 and figure 4.16 respectively.

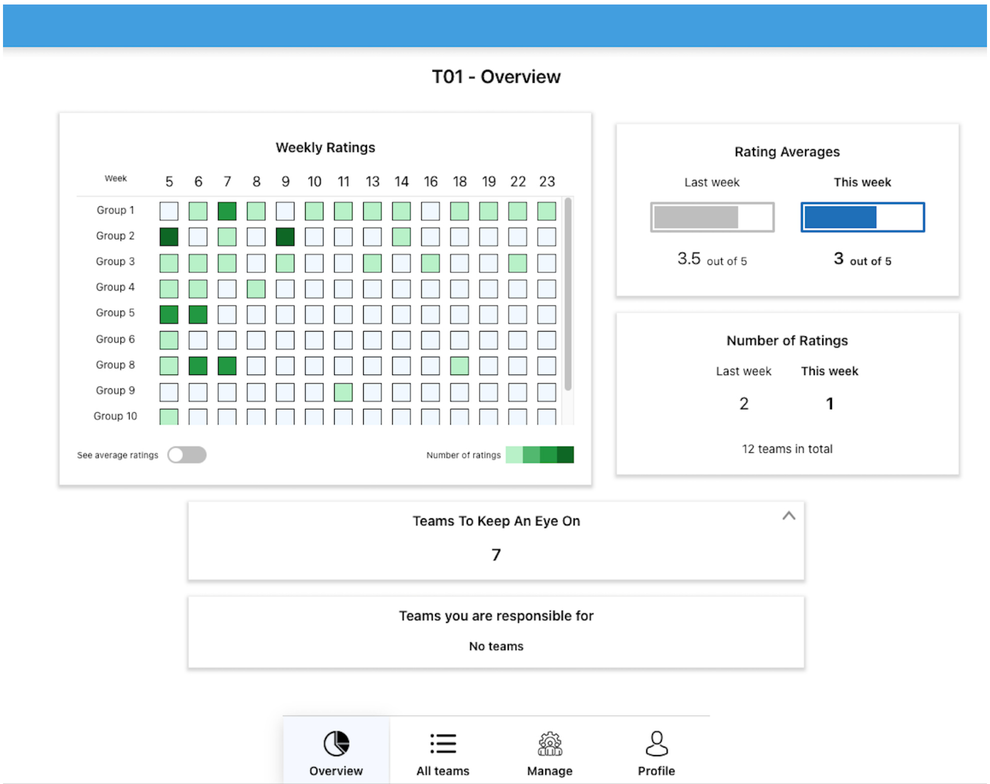
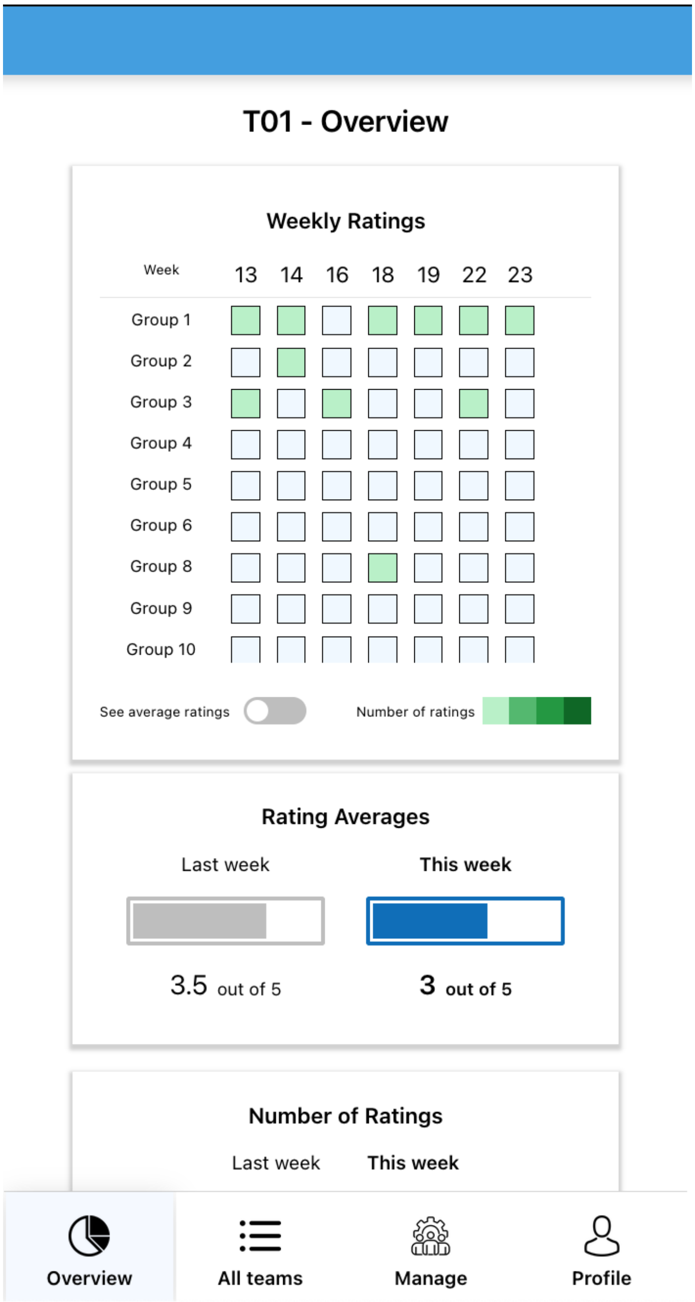


Figure 4.15: Overview Page on a desktop



T01 - Overview

Weekly Ratings

Week	13	14	16	18	19	22	23
Group 1	🟩	🟩	□	🟩	🟩	🟩	🟩
Group 2	□	🟩	□	□	□	□	□
Group 3	🟩	□	🟩	□	□	🟩	□
Group 4	□	□	□	□	□	□	□
Group 5	□	□	□	□	□	□	□
Group 6	□	□	□	□	□	□	□
Group 8	□	□	□	🟩	□	□	□
Group 9	□	□	□	□	□	□	□
Group 10	□	□	□	□	□	□	□

See average ratings ○ Number of ratings 🟩🟩🟩🟩

Rating Averages

Last week: 3.5 out of 5

This week: 3 out of 5

Number of Ratings

Last week This week

Figure 4.16: Overview Page on Mobile

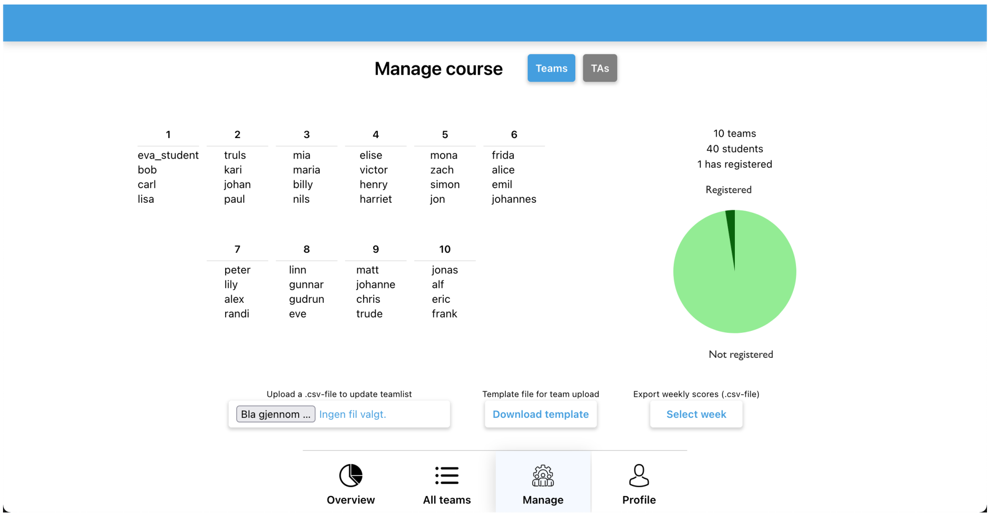
4.4.2 Manage Course

Another major change in the application was how instructors and undasses set up a course, and how to manage both teams and TAs. As most people do not have a programming background, this process had to be as easy as possible. The initial state of the application describes a Manage Course page shown in figure 4.6. The pages for registering both teams and TA's only included a place for uploading a .csv-file, shown in figure 4.7. It did not provide much information or description, and the user that is tasked to register for a new course, or update changes in the course, would need some prior knowledge to be able to do this. The button for accessing the Manage Course page was moved to the navigation bar at the bottom, making it more visible and giving a more cohesive look to the application. Instead of having a dedicated page for choosing between managing teams or TAs as shown in figure 4.6, the user gets directed directly to the Manage Team page where two tabs are visible at the top of the screen. The user can then easily switch between managing teams or TAs.

Teams

On the Manage Teams page, the user is presented with statistics regarding the current course. It includes a table of all students on their respective teams. Next to the list, there are some statistics about the course highlighted. This includes the total number of teams, how many students are pre-registered, and how many students have logged into the app and registered their accounts. Below is a pie chart showing the ratio between the students who have registered against the total number of students in the course. This further provides more insight to monitors of the course, presenting a detailed look at the current state of the application. When uploading a .csv-file to update the team list, the contents of the file are shown in a table. Here the user can see all the teams with their team members. This is an additional step that has two purposes. First, it allows the user to see the content of the file in a much more presentable way. They can more easily look over and make sure that they have entered the correct data. Secondly, it does not automatically send data to the database when adding a file, without the user being sure that's what's happening. By including this additional step, the user is more in control of what's going on, and can either go back, without any data being uploaded, or press upload, and have the data being sent. When the data is uploaded to the database, the user receives a message saying the upload was successful. The updated uploading process is shown in figure 4.18 and 4.19.

At the start of the semester when no data has been uploaded to the application, the user is presented with a text saying they need to upload a file to start with the registration. To guide the user to what type of file, and how the contents should preferably look, they can download a template file that they easily can fill in with real data. The updated Manage Team page is shown in figure 4.17 below.

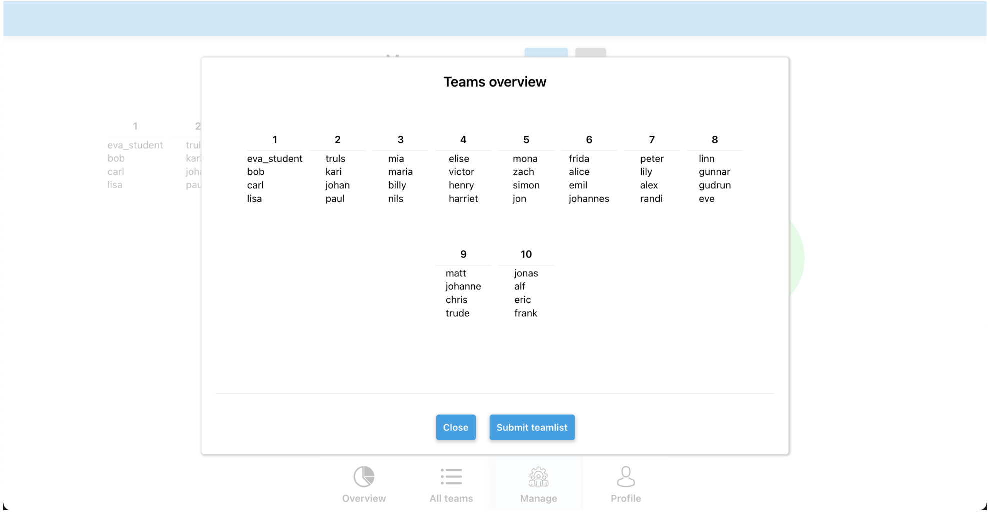


1	2	3	4	5	6
eva_student	truls	mia	elise	mona	frida
bob	kari	maria	victor	zach	alice
carl	johan	billy	henry	simon	emil
lisa	paul	nils	harriet	jon	johannes

7	8	9	10
peter	linn	matt	jonas
lily	gunnar	johanne	alf
alex	gudrun	chris	eric
randi	eve	trude	frank

10 teams
40 students
1 has registered

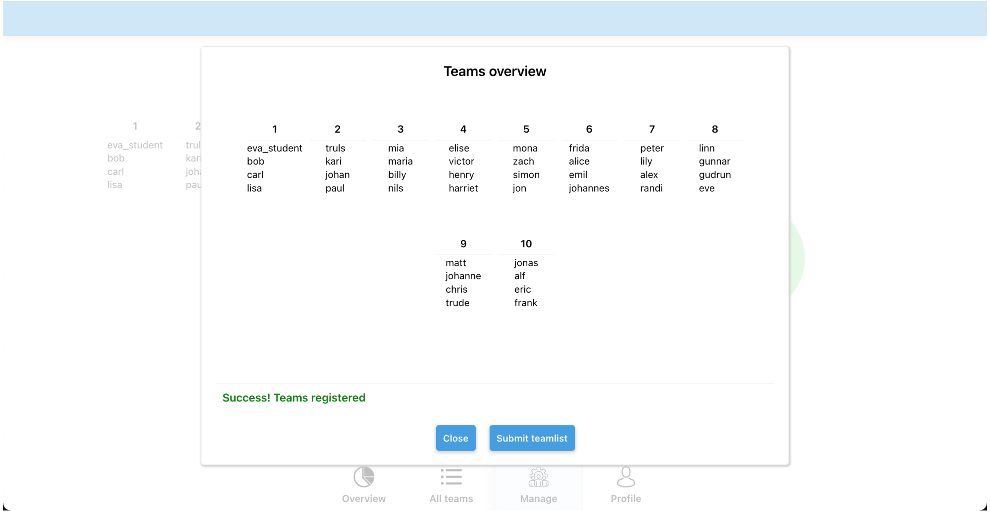
Figure 4.17: Manage Teams Page



1	2	3	4	5	6	7	8
eva_student	truls	mia	elise	mona	frida	peter	linn
bob	kari	maria	victor	zach	alice	lily	gunnar
carl	johan	billy	henry	simon	emil	alex	gudrun
lisa	paul	nils	harriet	jon	johannes	randi	eve

9	10
matt	jonas
johanne	alf
chris	eric
trude	frank

Figure 4.18: Upload Teams



Teams overview

1	2	3	4	5	6	7	8
eva_student bob carl lisa	truls kari johan paul	mia maria billy nils	elise victor henry harriet	mona zach simon jon	frida alice emil johannes	peter lily alex randi	linn gunnar gudrun eve

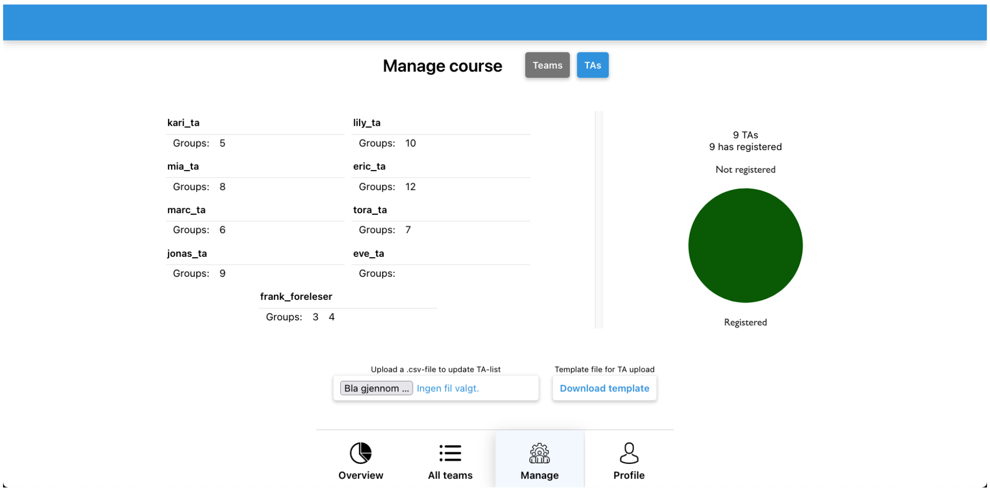
9	10
matt johanne chris trude	jonas alf eric frank

Success! Teams registered

Figure 4.19: Upload Teams Success

TAs

The process of registering TAs with their responsible groups had to be done through the administration panel in Django. We implemented a new way of doing this, in the same style as the team registration. When an instructor or an undass uploads a new .csv-file, they are presented with the contents of the file, identical to the way team registration is shown in figure 4.18 is done. The user has also the option to download a template of the .csv-file. To monitor the TAs, information such as the number of TAs registered compared to the total number of TAs in the course is available. The updated Manage TAs page is shown in figure 4.20 below.



Manage course

TA	Groups:
kari_ta	5
lily_ta	10
mia_ta	8
eric_ta	12
marc_ta	6
tora_ta	7
jonas_ta	9
eve_ta	
frank_foreleser	3 4

9 TAs
9 has registered

Teams
TAs
Not registered
Registered
Upload a .csv-file to update TA-list
Bla gjennom ... Ingen fil valgt.
Template file for TA upload
Download template
Overview
All teams
Manage
Profile

Figure 4.20: Manage TAs

Export Weekly Scores

As described in section 4.1.5 in chapter 4.1, we exported a .csv-file every week to provide a better view of data and to allow for integration with other types of applications such as Excel. This was done manually every week, and once the week was passed, the .csv-file for any previous week could not be retrieved easily. In this iteration of the prototype, we implemented a drop-down menu in the Manage Course page under the Teams tab. By pressing the button, 'Select week', the user could select any of the previous weeks in the semester, and download a .csv-file of the list of ratings in that week. The drop-down menu can be seen in figure 4.21 below.

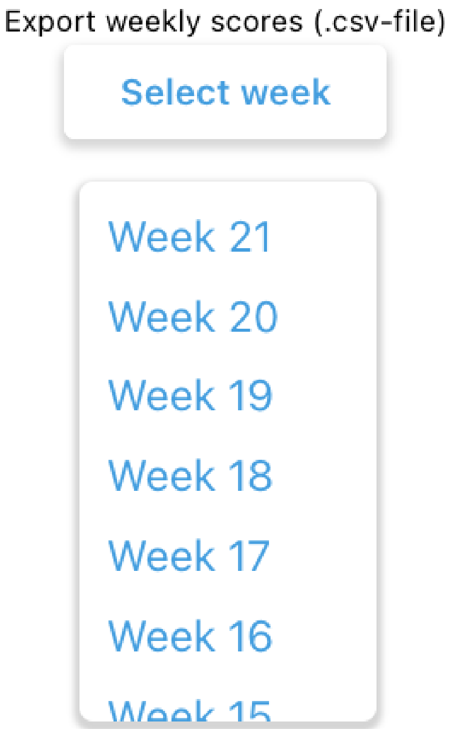
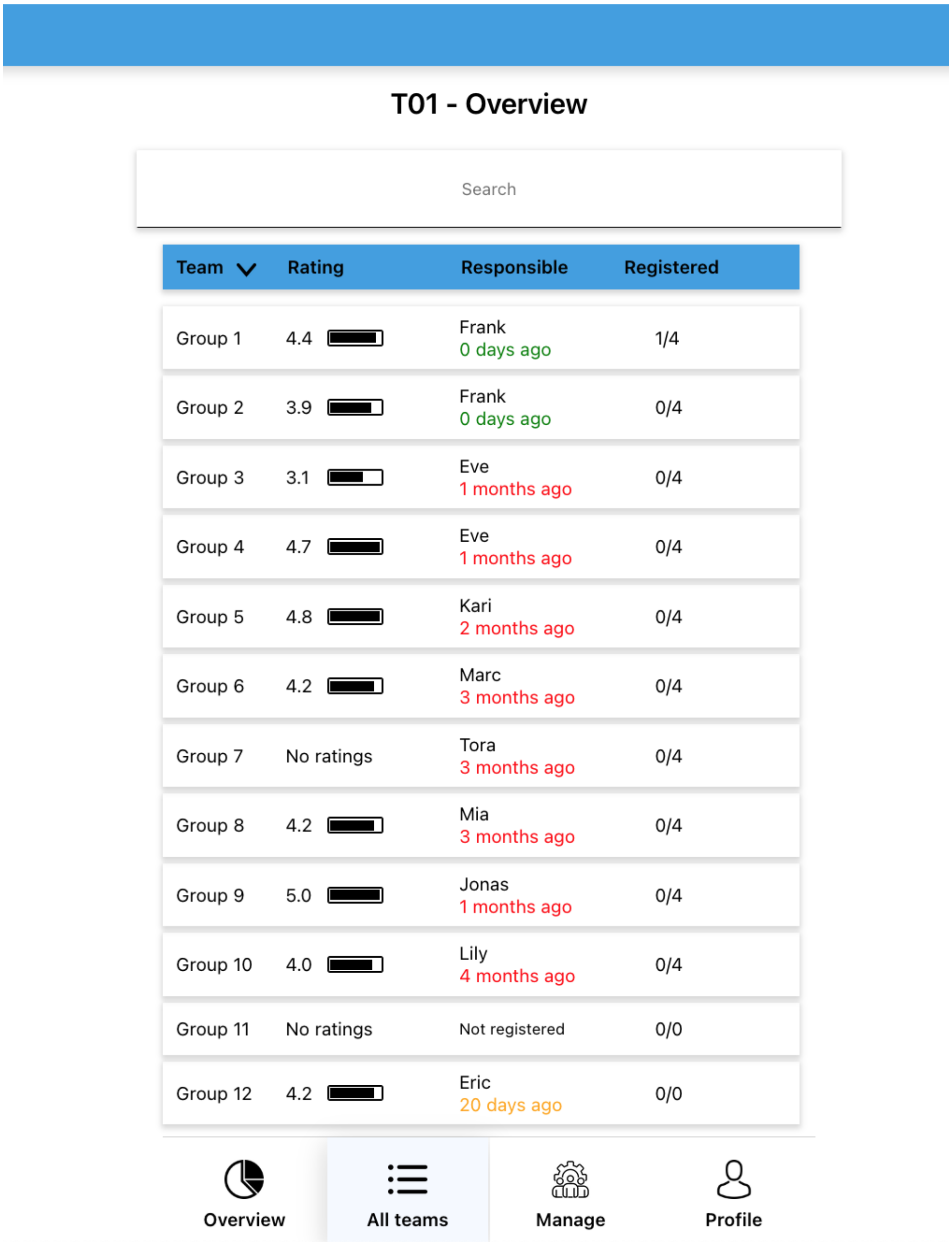


Figure 4.21: Export weekly scores

4.4.3 All Teams

Some changes were made to the list of all teams. First, the button for setting a default sorting of the list was removed. The configurations for sorting preferences were moved to the new Settings page which is described in section 4.4.4. The text in the 'last login'-feature was changed from an abbreviation to the full word, as the abbreviation 'm' for 'months' was misleading. The new list can be seen in figure 4.22 below.



T01 - Overview

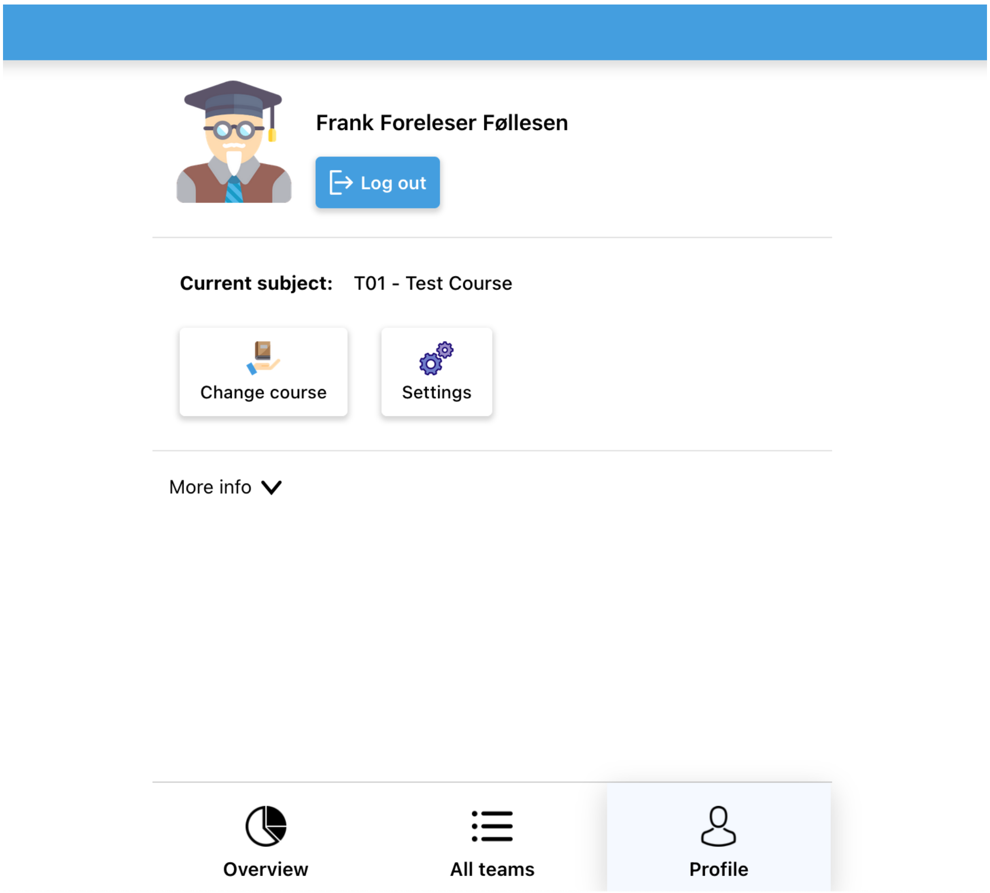
Search

Team	Rating	Responsible	Registered
Group 1	4.4	Frank 0 days ago	1/4
Group 2	3.9	Frank 0 days ago	0/4
Group 3	3.1	Eve 1 months ago	0/4
Group 4	4.7	Eve 1 months ago	0/4
Group 5	4.8	Kari 2 months ago	0/4
Group 6	4.2	Marc 3 months ago	0/4
Group 7	No ratings	Tora 3 months ago	0/4
Group 8	4.2	Mia 3 months ago	0/4
Group 9	5.0	Jonas 1 months ago	0/4
Group 10	4.0	Lily 4 months ago	0/4
Group 11	No ratings	Not registered	0/0
Group 12	4.2	Eric 20 days ago	0/0

Figure 4.22: All teams page

4.4.4 Settings

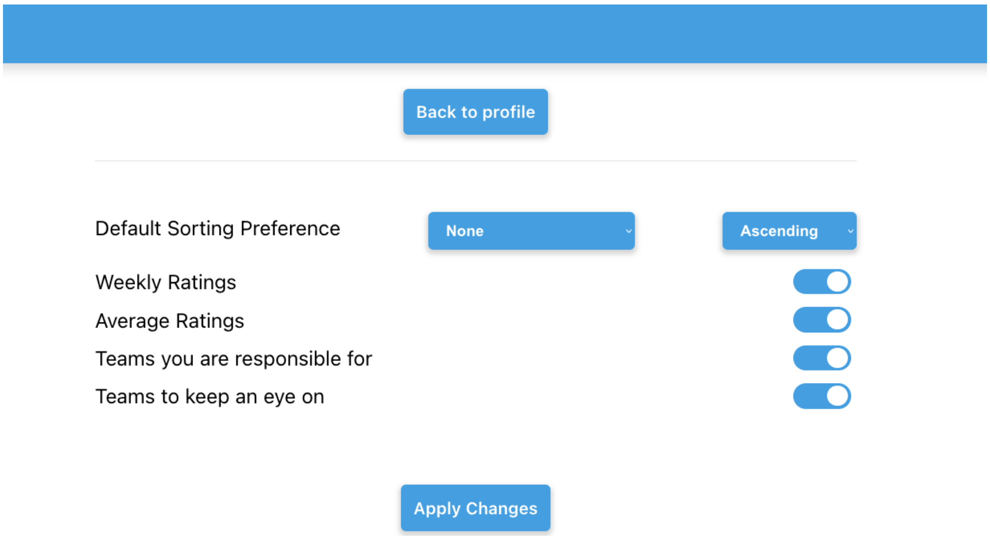
After implementing the default sorting preference and improving the Overview screen, we decided to let room for configuration for TAs and instructors. This decision was taken to let them personalize their application and its use. The main purpose is to make the user experience as smooth as possible, to encourage them to use the application and exploit its potential. To do so, we implemented a Setting screen accessible from the Profile section. This screen allows TAs and Instructors to choose which information boxes they want to display on their Overview screen, as well as set their default sorting preference. The added button for Settings is shown in figure 4.23 and the new Settings page is shown in figure 4.24.



Frank Foreleser Føllesen

Current subject: T01 - Test Course

Figure 4.23: Profile page



Default Sorting Preference None Ascending

Weekly Ratings ●

Average Ratings ●

Teams you are responsible for ●

Teams to keep an eye on ●

Figure 4.24: Settings page

Chapter 5

Results

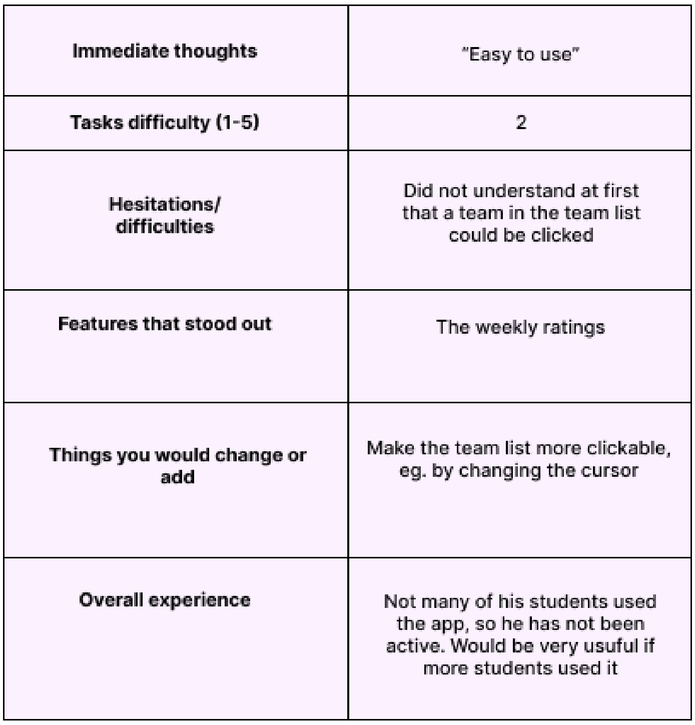
This chapter shows the results we got during this project. First, it contains all the qualitative data we gathered. This includes primarily the results from the user tests we performed. Later, it contains results from the quantitative data we collected after conducting the experiment in a real-world environment within a course at NTNU.

5.1 Qualitative results

The following results are presented as a figure for each of the test subjects, in both user test sessions. They consist of explicit feedback in terms of observations during the completion of the task, and the answers they gave in the discussion part after performing the tasks when answering our control questions. The figures are structured with the following content. The first measure is their immediate thoughts about the prototype. This was the first question we asked the test subjects immediately after completing the tasks. Then, a number from 1 to 5 indicates how difficult they felt the tasks were, where 1 is very easy and 5 is very hard. Next is if they had any hesitations or difficulties, followed by any features in the prototype that stood out. Then comes things they would either change or add before you see their overall experience with the prototype.

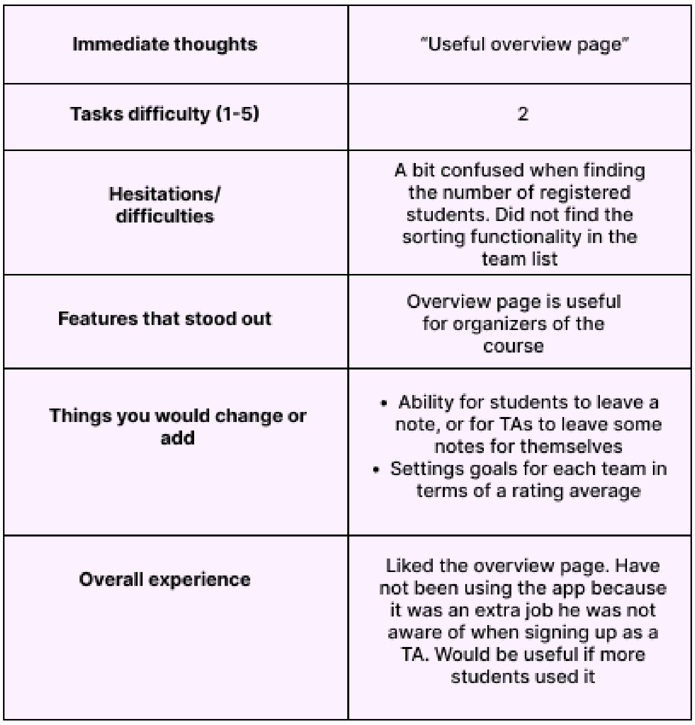
5.1.1 First User Test

The second task, presented in section 3.2.3, asks the user to find the number of ratings within a specific group at a specific week. We wanted to test out one of the new features in the newly added 'Weekly Ratings'-component on the Overview page. The week number they asked to find is hidden in the small box and to find it, they would need to open a pop-up view of the Weekly Ratings by clicking the button in the top-right corner. This view will show all the weeks in the semester. The button can be seen in figure 4.9. Our first test subject did look through the 'Weekly Ratings'-component, and navigated to the correct group by scrolling, but when trying to find the correct week number, he did not find the button for the pop-up-view. Instead, he moved on to a different page, the All teams page, where he did find the correct answer, in the end, resulting in a lot more clicks. Our second test subject went straight to the All teams page, instead of looking at the new Weekly Ratings box. They both found the correct answer, but not by using the new features we added to the Overview page. To summarize the qualitative results from the first user test session, we have categorized the feedback into 6 measures as explained in the introduction of section 5.1.



Immediate thoughts	"Easy to use"
Tasks difficulty (1-5)	2
Hesitations/ difficulties	Did not understand at first that a team in the team list could be clicked
Features that stood out	The weekly ratings
Things you would change or add	Make the team list more clickable, eg. by changing the cursor
Overall experience	Not many of his students used the app, so he has not been active. Would be very usuful if more students used it

Table 5.1: Test subject 1

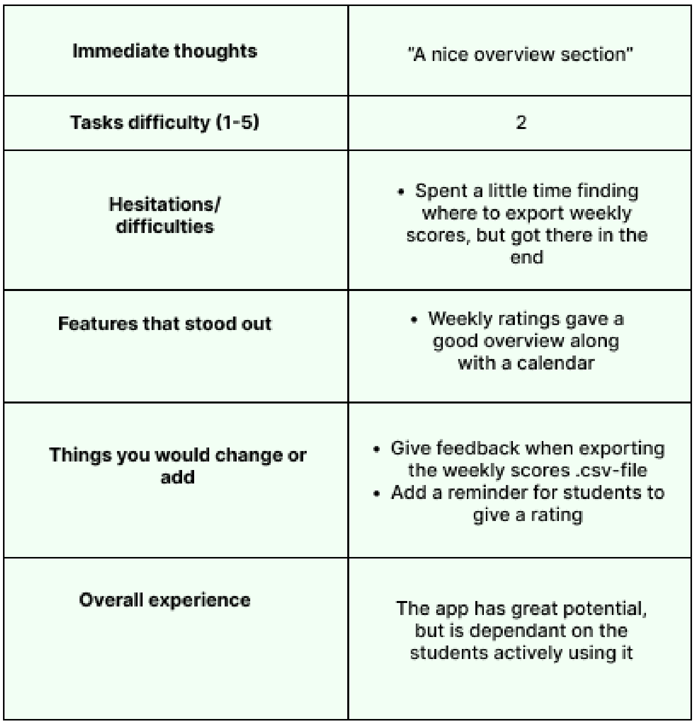


Immediate thoughts	"Useful overview page"
Tasks difficulty (1-5)	2
Hesitations/ difficulties	A bit confused when finding the number of registered students. Did not find the sorting functionality in the team list
Features that stood out	Overview page is useful for organizers of the course
Things you would change or add	• Ability for students to leave a note, or for TAs to leave some notes for themselves • Settings goals for each team in terms of a rating average
Overall experience	Liked the overview page. Have not been using the app because it was an extra job he was not aware of when signing up as a TA. Would be useful if more students used it

Table 5.2: Test subject 2

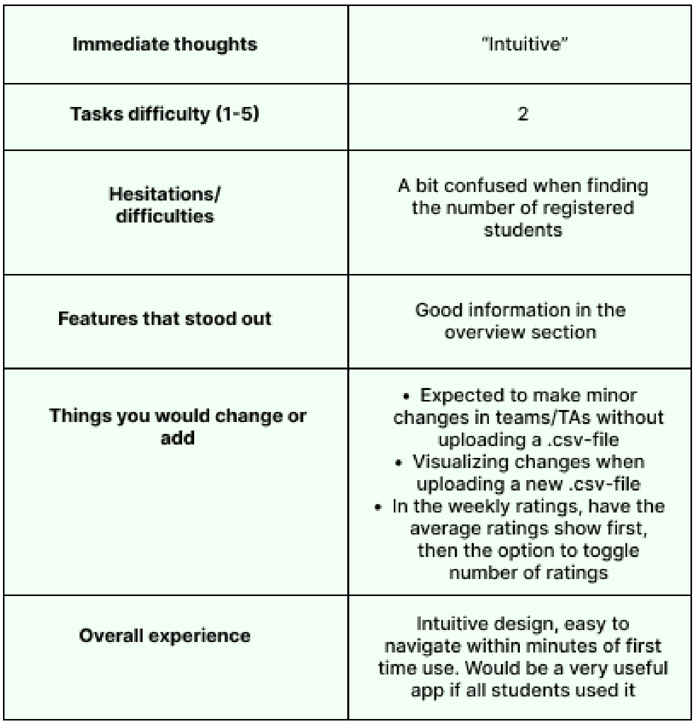
5.1.2 Second User Test

One of the issues the test subjects had in the previous session of user testing, was not fully utilizing the new 'Weekly Ratings'-component. One improvement we did in preparation for this user test, was expanding the component to include all weeks in the semester at first glance. To test this, we asked a similar question as last time, where they were asked to find the average rating for a specific group in a specific week, as presented in section 3.2.4. This time, every test subject used the 'Weekly Ratings'-component to find the answer. Another feature we tested was the improved Manage Course page. The test subjects were asked to update the course data in the backend using the new interface. This includes both the list of all teams with their corresponding team members, as well as the list of all teaching assistants with the teams they are responsible for. All test subjects managed to successfully update both lists. The figures below show the qualitative results from the second user test session, one for each test subject.



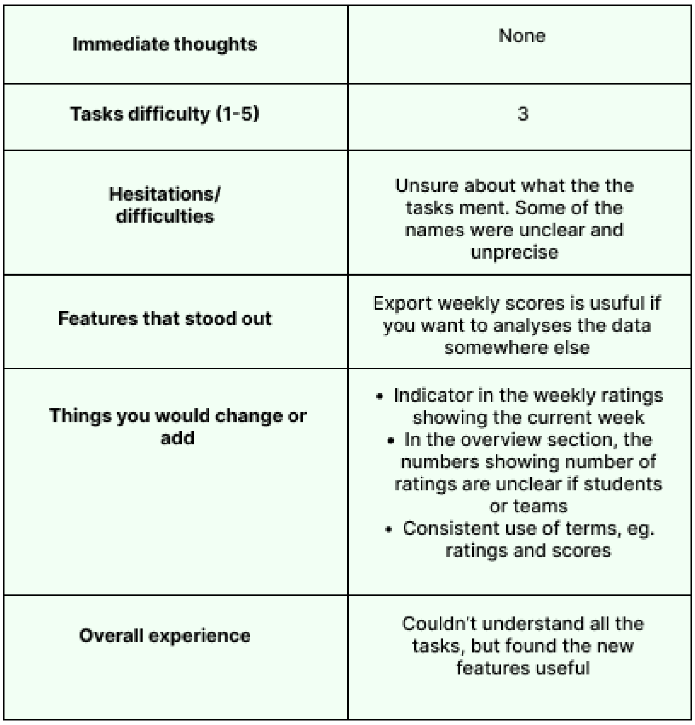
Immediate thoughts	"A nice overview section"
Tasks difficulty (1-5)	2
Hesitations/ difficulties	• Spent a little time finding where to export weekly scores, but got there in the end
Features that stood out	• Weekly ratings gave a good overview along with a calendar
Things you would change or add	• Give feedback when exporting the weekly scores .csv-file • Add a reminder for students to give a rating
Overall experience	The app has great potential, but is dependant on the students actively using it

Table 5.3: Test subject 1



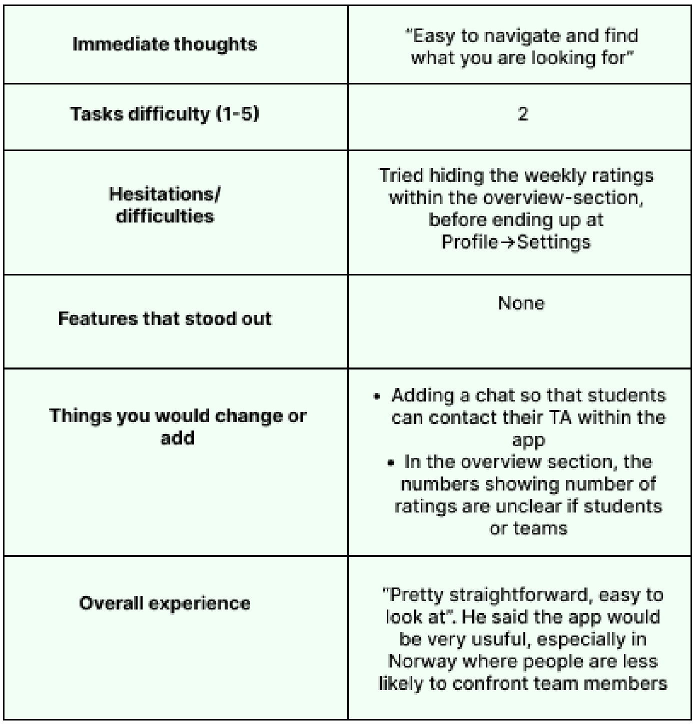
Immediate thoughts	"Intuitive"
Tasks difficulty (1-5)	2
Hesitations/ difficulties	A bit confused when finding the number of registered students
Features that stood out	Good information in the overview section
Things you would change or add	• Expected to make minor changes in teams/TAs without uploading a .csv-file • Visualizing changes when uploading a new .csv-file • In the weekly ratings, have the average ratings show first, then the option to toggle number of ratings
Overall experience	Intuitive design, easy to navigate within minutes of first time use. Would be a very useful app if all students used it

Table 5.4: Test subject 2



Immediate thoughts	None
Tasks difficulty (1-5)	3
Hesitations/ difficulties	Unsure about what the the tasks ment. Some of the names were unclear and unprecise
Features that stood out	Export weekly scores is usuful if you want to analyses the data somewhere else
Things you would change or add	• Indicator in the weekly ratings showing the current week • In the overview section, the numbers showing number of ratings are unclear if students or teams • Consistent use of terms, eg. ratings and scores
Overall experience	Couldn't understand all the tasks, but found the new features useful

Table 5.5: Test subject 3



Immediate thoughts	"Easy to navigate and find what you are looking for"
Tasks difficulty (1-5)	2
Hesitations/ difficulties	Tried hiding the weekly ratings within the overview-section, before ending up at Profile→Settings
Features that stood out	None
Things you would change or add	• Adding a chat so that students can contact their TA within the app • In the overview section, the numbers showing number of ratings are unclear if students or teams
Overall experience	"Pretty straightforward, easy to look at". He said the app would be very usuful, especially in Norway where people are less likely to confront team members

Table 5.6: Test subject 4

5.2 Quantitative Results

This section presents our quantitative results in this master thesis and is divided into two parts. First, we present the data from the usage of the application. This includes the data we were able to gather after an entire semester of usage from the very start in January, to the end in late May. Secondly, we present the implicit feedback from the two iterations of the user tests.

5.2.1 Usage of the app

- In the course TDT4180, 517 students were preregistered in the app. Out of them, 187 registered as a user by logging in to the app.
- 23 TAs registered in the app, every TA in the course.
- The course had two undasses, and they both registered.
- A total of 351 ratings got registered during the entire semester across all teams.
- 65 out of all ratings were 3 or lower.
- 33 out of all ratings were 2 or lower.
- 81% of all ratings were 4 or 5.
- The total number of teams registered was 104, and 71 of the teams left at least one rating.

- Out of the 71 teams that left a rating, 51 of them had an average rating of 4.0 or more.
- 20 teams had a rating average below 4.0
- In the first week, 62 students left a rating.
- The highest number of ratings happened during the second week, week 6, and it had 68 ratings left by students.

As the semester in the course TDT4180 progressed, most of the groups seemed to be okay and did not need any assistance. However, some groups did struggle and could use some help. The course facilitators reported that 14 groups had some issues, and half of them were detected through Team Accelerator. Most of the reported issues were related to communication problems. Figure 5.1 below shows the number of ratings for each week during this project. The students started using the application in week 5 and stopped using it after week 21.

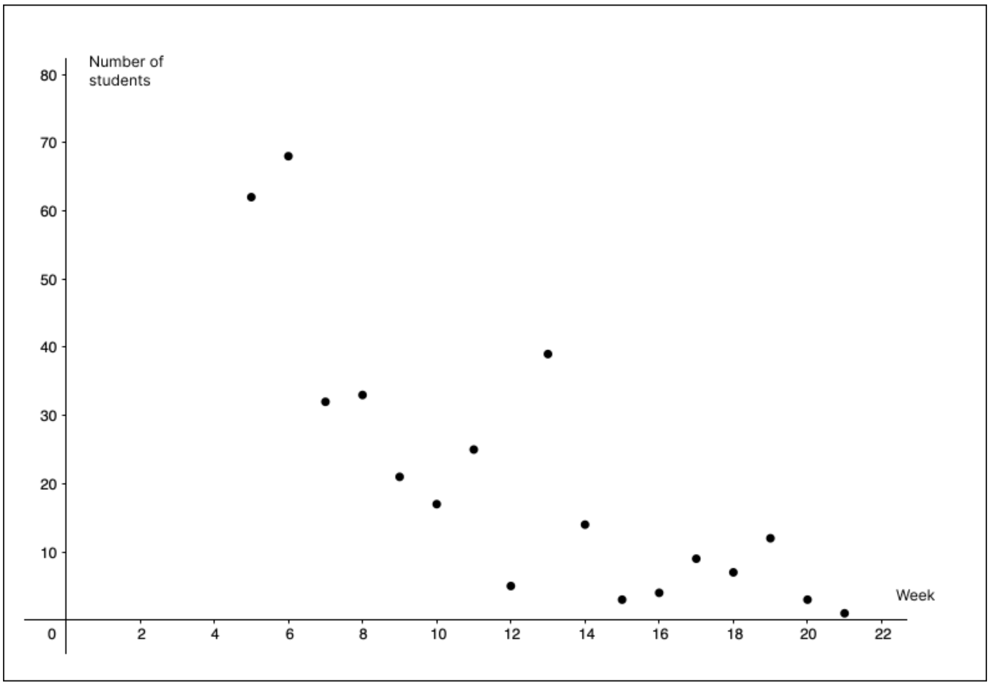


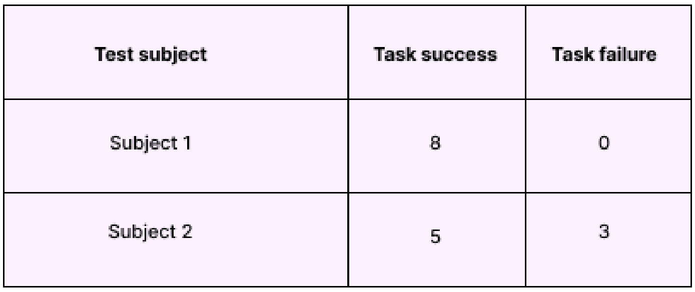
Figure 5.1: Total number of ratings each week

5.2.2 User tests

The qualitative results from the user tests include the success rate of the tasks and the time spent. The results are presented in the following figures.

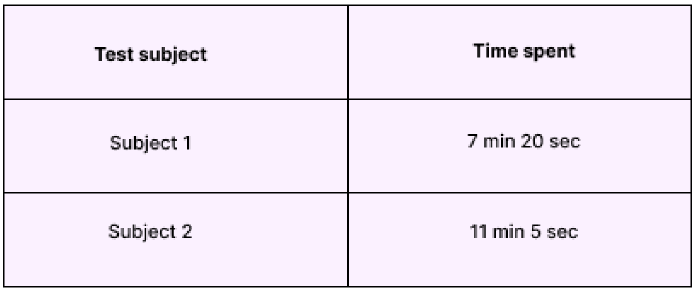
First User Test

As shown in figure 5.7 and figure 5.8 below, test subject 1 had no failures and spent about 7 minutes doing the tasks. On the other hand, test subject 2 had 3 failures which meant spending a little more time figuring the tasks out at about 11 minutes.



Test subject	Task success	Task failure
Subject 1	8	0
Subject 2	5	3

Table 5.7: Success rate of the first user test tasks

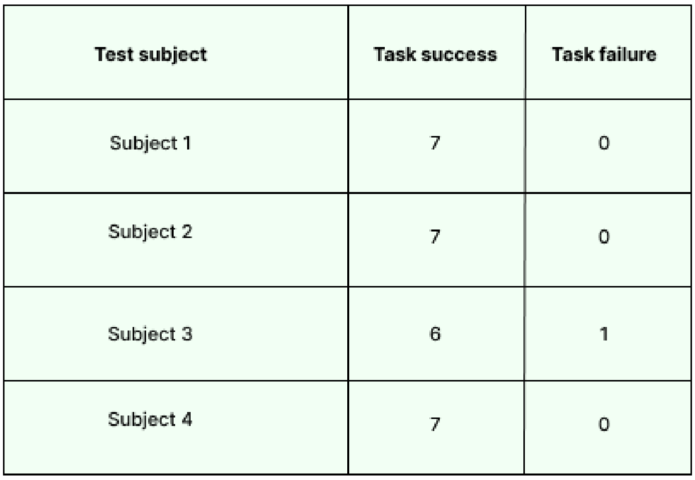


Test subject	Time spent
Subject 1	7 min 20 sec
Subject 2	11 min 5 sec

Table 5.8: Time spent doing the tasks of the first user test

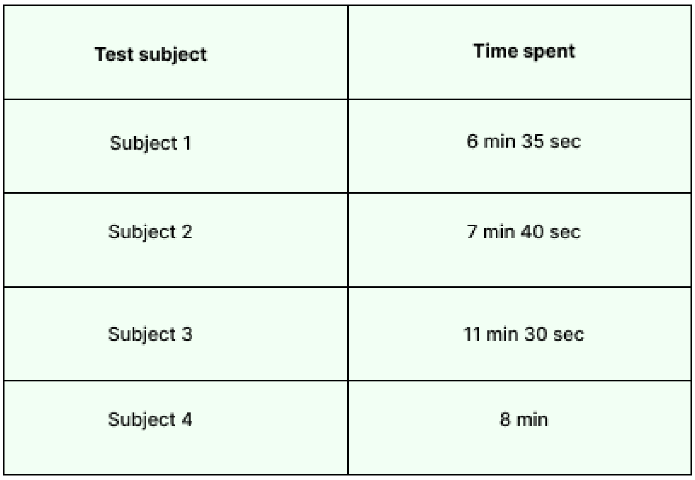
Second User Test

Figure 5.9 shows that there was only one task failure between all the four test subjects. Most of the users spent about 7-8 minutes, while the user that struggled a bit more used 11 minutes.



Test subject	Task success	Task failure
Subject 1	7	0
Subject 2	7	0
Subject 3	6	1
Subject 4	7	0

Table 5.9: Success rate of the second user test tasks



Test subject	Time spent
Subject 1	6 min 35 sec
Subject 2	7 min 40 sec
Subject 3	11 min 30 sec
Subject 4	8 min

Table 5.10: Time spent doing the tasks of the second user test

Chapter 6

Discussion

After collecting and presenting the results, we go into the discussion of this thesis. First, we discuss the findings of the previous chapter 5. This includes the usage of the application within the course TDT4180 during the spring semester. In section 6.2 we go through all iterations of the prototype. Here we highlight all the design changes and feature implementations we did, as a result of the feedback of the user tests. Lastly, we discuss the validity of the tests and what biases that came with them.

6.1 Usage of Team Accelerator

One of the challenges we had going into this project was getting people to use the application. We had multiple strategies to do so, including giving out gift cards to the most active students within the course, as stated in section 3.3 in chapter 3. Out of about 500 students, roughly 200 used the application at least once. August argues in the preceding master thesis project [1], getting about 60 students to register within the app, that giving an incentive to students probably will increase the number of users. This seems to be true, considering an increase of about 200% in users. Another large impact of the increase in users was that the application was deployed and presented at the very start of the semester. As figure 5.1 in chapter 5 shows, the number of ratings left by students was much higher in the first two weeks than any other week. The activity decreases over the duration of the semester. This leads to another challenge, which is keeping users active. One measure we did to make the app more available was to include a link within the course page in Blackboard. It may have worked to some extent, but students still need to remember to use the application, as Team Accelerator itself does not remind users to give a rating each week. Getting more active users is needed to give depth to any answer for our RQ1 1.3, however, we can say that Team Accelerator in fact did help 7 groups that were struggling. The way it helped the groups were that the facilitators spotted bad ratings, and contacted those groups for a conversation. This led directly to find out issues that most likely would not have been spotted as early as they did.

6.2 Prototype

6.2.1 Design changes

Our main focus for implementation was to improve the application for the facilitators of the course. In the application, their main objective is to detect groups that are struggling. To answer our RQ1.1 1.3, we focused on improving the visualization of data on the Overview page. The following were the changes in the application compared to the initial state and the state that was being tested in the first user test session.

- Adding 'Weekly Ratings' to the Overview page.
- Adding 'Teams To Keep An Eye On' to the Overview page.
- Adding a 'Last Login'-feature for TAs on the All teams page.
- Adding the number of registered students within each group on the All teams page.
- Adding 'Prev' and 'Next'-buttons to the 'Team Details'-view in the All teams page.

After performing the first user test session we collected the data and reviewed it. Even getting only two test participants, we did get a lot of useful feedback. Both of the users did not use the 'Weekly Ratings'-component to find the number of ratings of a team in a specific week. This could have multiple reasons. First, the button for opening the pop-up view was not as visible as it could be. Either the color was too light, not making enough contrast, or the icon used did not present itself as the action it performs. Another reason could be that both of the test subjects had previous experience with an older version of the app, which might lead them towards using the interface the same way they had done before. Either way, we decided to make some changes to make it easier to spot the new features and make them more intuitive. Each of the points below explains what was changed and why.

- Expanded the 'Weekly Ratings'-component to include all weeks as a standard. This was because users did not use the pop-up view, and found the information using another page. Having a lot of available space on the Overview page, as it was mobile-tailored, we rearranged the entire screen, utilizing more of the space available. Figure 4.15 shows the new changes.
- Added a toggling text to the 'Weekly Ratings'-component describing the functionality it provides. Users did not try to use the toggle and could therefore not see the heat map of the average ratings. To find the answer they navigated to a different part of the application. New changes are represented on figure 4.11.
- Added two new components on the Overview page, one for rating averages and one for the number of ratings. They both compare statistics from the current week to the previous week. This was a measure to give more insight into the data which is more relevant for the monitors of the course as shown

on figure 4.15.

- Discarded the component containing the average rating for all teams. As the goal of this application is to find teams that are struggling, and as the majority of teams usually give more than 4 in ratings, the struggling teams get drowned in this statistic.
- Discarded the component containing the number of total teams and moved this information to the 'Number of Ratings'-component. When rearranging the Overview page, we found little need for having a dedicated component, only to show the total number of teams. This would also allow more space for the two new components we implemented.
- Improving the responsiveness of the application. A natural step when expanding the Overview page, and implementing a much improved Manage Course page. Users use different screen sizes, and it makes more sense to utilize the screen size that is available at any given time.
- Moving the button for 'Manage course' to the navigation bar at the bottom of the screen. The old button was hidden away a lot of the time, and the user needed to scroll to find it. When the application already has a navigation bar, it makes a lot more sense to include the button here. The button is only available and visible for instructors and undasses.
- Implemented a brand new 'Manage course' page for both teams and TAs. This includes an additional step when uploading .csv-files showing the content of the file. It includes an option to download a template file for uploading such lists, as well as a drop-down menu for exporting weekly scores in any given week as shown on figures 4.20 and 4.17.
- Changed the wording of the 'Last login' feature to include the full word for 'months' and 'days', instead of 'm' and 'd'. The main reason was that 'm' could be misinterpreted as minutes, instead of months. Figure 4.22 shows the new changes.
- Implemented a new Settings page where we added functionality for hiding and showing components in the Overview page. We also moved the configurations for setting a default sorting preference from the All teams page to the new Settings page as shown on figures 4.23 and 4.24.

6.2.2 Test validity

For the first user test, we only got two participants. Jakob Nielsen argues in his article [29] that by using about five people for a user test, you would uncover the majority of design issues. The more users you get, you learn less and less for each added user. The value you get from adding more test users follows a logarithmic curve, Nielsen proposes. We could only get two users for the first user test session, but we still could get a lot more useful feedback and results than just having one. As seen in figures 5.1 and 5.2 in chapter 5, the test users had different experiences with the application. However, by adding a few more participants the validity of the test would increase. This was the case for our second user test session. By doubling the number of users who interacted with the prototype, we were able to retrieve more perspectives and learned more about the design. The users in the first session were both teaching assistants in the course where we tested the application. By being in the target group and somewhat familiar with the previous iteration of the prototype, they were the best types of candidates we could get. We reached out to all 23 of the TAs in the course, but only two of them were able to attend. They also joined our second user test session, which comes with both advantages and disadvantages. The advantage is that they still are end-users of the application, and have a perception of what the application looked like and behaved like before. The disadvantage, however, is that by reusing users for multiple sessions of user testing, the view of the prototype can become more limited and it can bring with it some biases. Additional two users participated, and they used the application for the very first time. They had both experiences with being a teaching assistant at NTNU, and would still qualify for being in our target group. The validity of the second user test session was thus stronger than the first session.

6.2.3 Bias

After concluding our usability tests, and according to the research on bias done as background work, we were able to spot mistakes that could have been corrected. While building our test scenarios and settings the configuration, we tried to avoid biases as much as possible. Nonetheless, we were forced to deal with different issues which may have distorted the authenticity of our results.

Before starting usability tests, we had to find participants. It was natural for us to ask TAs from the course the application was being tested in this semester. Among the 23 TAs of the course, our first demand was not very conclusive as we did not receive a positive answer. After discussing this issue with our supervisor, the prominent solutions were to open the tests to more subjects, while offering gift cards to participants as a reward for their effort. Ideally, users should express the will to participate in tests by themselves to have authentic results. Hence this situation was an issue as gift cards could alter their motivation. While other methods were suggested by August in [1] to encourage students to use the application prototype, like making the use of the application count for a certain percentage of the final grade, such method was not applicable in our situation. After discussing this problem, we decided that its influence on our tests would be negligible considering that it lasted approximately 30 minutes. The test's short length kept the participant involved before any fatigue signs.

For our first usability test session, we gathered two TAs from the course. Besides having all the necessary group monitoring knowledge and some experience with the application, we questioned the results as we did not have a participant from outside of the course to compare their opinion and answers. We focused on this aspect during our second usability test session. Having two TAs from the course, and two old TAs to participate, allowed us to have a different point of view:

- TAs from the course
- TAs not related to the course
- New participants for the second usability tests session
- Participants with first session experience

To balance biases related to user behavioral change, we clearly stated at the beginning of each session, as a reminder, that it was the application and only the application that was being tested, and not the user. Even though we took into account the task order bias and the wording bias detailed in section 2, we manage to spot mistakes in our methodology after the first test iteration. Our second user from the first iteration noticed that our physical expressions, such as nodding and facial expressions, could influence the user by giving indications on whether he is doing well or not.

In addition, after analyzing the results of each session and watching the recorded sessions, we noticed some issues with the order of the questions. Indeed, some consecutive tasks part of a classic scenario of application usage resulted in the user doing several tasks at the same time, or one task success helping a lot for the next one. To balance this issue we modified the scenario from test iteration 1 to test iteration 2 while trying to keep a realistic scenario as much as possible.

Finally, some users mentioned that the wording of some questions could be disturbing as specific terms used in the application were used or not in the questions. Therefore it was either too much help for them or confusing to understand the purpose of the task to complete.

Chapter 7

Conclusion and Future Work

Chapter 7 contains our conclusion of this master thesis and suggestions for future work. First, we conclude our project and go through our research questions. Then, we suggest improvements to Team Accelerator for anyone that decides to continue the development of this application. This includes both design suggestions based on feedback from our latest user test, as well as suggestions for adding new features and integration with Blackboard.

7.1 Conclusion

The number of participants in the course TDT4180 on which we tested Team Accelerator was indeed larger than before, however, it is still not enough to make a verdict if this type of application improves group work. For now, because of the limitations in the size of the data, it has little strength and reliability. What we can conclude, however, is that it shows a lot of potential. This master thesis should therefore act more as a step into further development and testing. Our first research question, RQ1, asks in what ways a tool such as Team Accelerator can improve group-related projects 1.3. As we have seen, 7 groups that were struggling were detected through Team Accelerator, which helped the teaching assistants interact and engage in conversation with the group a lot earlier than they would without the application. To detect even more groups with bad ratings, we have explored multiple ways of highlighting relevant data within the app. This leads us to our second research question, RQ1.1, how can we improve the design of Team Accelerator to better find groups that are struggling. This has been done through multiple iterations of prototype user testing, but if the design is actually improved where the aim is to detect struggling groups is up to future testing to prove.

7.2 Getting more participants

As pointed out by several of the user test subjects presented in chapter 5, if the application had more students involved, it could be a highly useful product. When only a fraction of the students are registering ratings, and the majority of them are satisfied with their group, the value of an application such as Team Acceleration gets diminished. One measure we did to get more people involved and active in using the app was handing out gift cards as a reward. It worked to some extent, but for a tool such as Team Accelerator to improve group work, and especially capture the groups that are having issues, more participants are needed. During the specialization project, we discussed different methods to get more students to be active, one of them is handing out gift cards. The other one was including the use of the app as part of the grade in the course. This is more difficult to get approved by the institute at the university, but might be a possible step to further increase the number of users in the future. Another solution may be to increase the value of the gift cards, but even then, it might not be enough for students to participate voluntarily. One implementation that should be done in the future is to add a reminder for students to register a rating every week. It can be easy to forget, and getting a notification will most likely increase the activity of users.

7.3 Suggestions for future work

Based on our final user test and the feedback we got, we have some suggestions for changes, or implementations, for future development of this application.

- Adding a weekly reminder for students to register a rating. This can be a large improvement to get more students to use the app.
- Adding a step and providing feedback when exporting weekly scores in the Manage Course page, as now it downloads immediately the file without any notification to the user.
- Highlighting changes when uploading a new .csv-file to update the list of teams or TAs. Can be color-coded, with green color for the new students on the list, red color for those that have dropped out, and yellow color for those who have changed groups.
- Specifying that the number of ratings on the Overview page represents the number of students, as more than one user was a bit confused by what the numbers were.
- Change the standard view for Weekly Ratings in the Overview page as the average ratings, not the number of ratings. This information might be more important to visualize first, but this is up to further testing to evaluate.
- Adding an indicator for the current week in the Weekly Ratings on the Overview page. For example, this can be done either by making the column for the current week larger or by changing the week number to a bold text.
- Implementing a way of communicating between students and teaching assistants in the application.
- According to [7] and [6], having a plan, and roles in a group help participants to work more efficiently and without conflicts. Hence adding a README.txt-like page for each group and roles to have someone to talk

with TAs would help students to organize their work.

7.4 Database Backups

Instead of manually backing up the database files every day, a more efficient and improved solution would be to automate this process. A data collection could be set every morning at 8 am or every week at the time of ratings refresh to gather weekly data. The database dump files should be stored more in than one location, to account for losses if one of the locations crashes.

7.5 More Data - New Data

The current state of the application only allows students to rate their group once per week. According to our usability tests feedbacks, it would be very valuable to add a group message functionality and a direct message system to communicate with their assigned TA. In addition, to make the application useful for students, a README.txt-like document would help TAs to get more involved in their group progress. An edition mode for students, and a reading mode for TAs with a comments functionality, would give more information about the meaning of unusual ratings.

Considering that this application will be improved and used over several years at NTNU, analyzing the data between different semesters would offer great feedback on the impact of course modification on group work. Hence, the possibility to import data from a previous dump according to the app version, to check on the previous year's data is a future functionality with great potential.

7.6 Connection to Blackboard

The way Team Accelerator works now is to manually upload a .csv-file to register new groups and TAs in the app. The .csv-file is exported from Blackboard and is ready to be uploaded as-is. However, a better solution would be to integrate this process entirely, by connecting Team Accelerator to the Blackboard API. Moreover, linking Team Accelerator to the BlackBoard API would allow automating most aspects of the current prototype like course creation, course deletion, groups change, assigned TAs, and Instructor.

Chapter 8

Acknowledgment

First, we would like to thank our supervisor, George Adrian Stoica, for guiding us through this project. We would like to thank all the participants in the course TDT4180 for using the application throughout the spring semester, and NTNU for letting us conduct such an experiment. Thank you to the ISSE group at IDI for financial support to purchasing gift cards. We would also like to thank every participant in the user testing sessions for taking their time and sharing their thoughts. Thank you to NTNU for letting us host the application at their servers, and to the IT Support for their help and guidance.

Bibliography

[1] A. Lund Eilertsen, ‘Team Accelerator,’ en, 2020. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2777642.

[2] *Resolving group work issues*, en, Oct. 2019. [Online]. Available: https://my.uq.edu.au/information-and-services/student-support/study-skills/group-work/resolving-group-work-issues.

[3] S. Gantasala, ‘Collaborative team work in higher education: A case study,’ *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, vol. 9, p. 636, Jan. 2015.

[4] S. Daityari, *Angular vs React vs Vue: Which Framework to Choose in 2022*, en-US, Jan. 2019. [Online]. Available: https://www.codeinwp.com/blog/angular-vs-vue-vs-react/.

[5] *Angular vs React vs Vue: Which Framework Is Better? 2022*, en-US, Nov. 2020. [Online]. Available: https://athemes.com/guides/angular-vs-react-vs-vue/.

[6] J. Scarnati, ‘On becoming a team player,’ *Team Performance Management*, vol. 7, pp. 5–10, Feb. 2001. DOI: 10.1108/13527590110389501.

[7] D. W. Johnson and R. T. Johnson, ‘Making cooperative learning work,’ *Theory Into Practice*, vol. 38, no. 2, pp. 67–73, 1999, ISSN: 00405841, 15430421. [Online]. Available: http://www.jstor.org/stable/1477225.

[8] P. Tarricone and J. Luca, ‘Successful teamwork: A case study,’ *Research outputs pre 2011*, Jan. 2002. [Online]. Available: https://ro.ecu.edu.au/ecuworks/4008.

[9] *The Importance of Human-centered Design in Product Design*, en. [Online]. Available: https://www.toptal.com/designers/ux/human-centered-design.

[10] D. C. Design, *What Is Human-Centered Design?* en, Oct. 2020. [Online]. Available: https://medium.com/dc-design/what-is-human-centered-design-6711c09e2779.

[11] V. Korbel, *The unified theory of human-centred design and its success — the wheel of innovation*, en, Feb. 2021. [Online]. Available: https://uxdesign.cc/the-unified-theory-of-human-centred-design-and-its-success-the-wheel-of-innovation-dbafd5ed12b4.

[12] J. Bastien, 'Usability testing: A review of some methodological and technical aspects of the method,' *International journal of medical informatics*, vol. 79, e18–23, May 2009. DOI: 10.1016/j.ijmedinf.2008.12.004.

[13] W. L. i. R.-B. U. Experience, *Usability Testing 101*, en. [Online]. Available: https://www.nngroup.com/articles/usability-testing-101/.

[14] *What is Agile Software Development (Agile Methodologies)?* en. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development.

[15] *Find Market Research Companies, Facilities, Jobs, Articles, More | Quirks.com*, en. [Online]. Available: https://www.quirks.com/articles/9-types-of-research-bias-and-how-to-avoid-them.

[16] *Six types of bias in research*, en-GB, Jun. 2021. [Online]. Available: https://www.experienceux.co.uk/ux-blog/six-types-of-bias-in-research/.

[17] *Acquiescence Bias: What it is & How to Stop it*, en, Dec. 2020. [Online]. Available: https://www.qualtrics.com/blog/avoiding-the-yes-bias/.

[18] C. Heneghan, *Hawthorne effect*, en, Nov. 2017. [Online]. Available: https://catalogofbias.org/biases/hawthorne-effect/.

[19] W. D. Perreault, 'Controlling order-effect bias,' *The Public Opinion Quarterly*, vol. 39, no. 4, pp. 544–551, 1975. [Online]. Available: http://www.jstor.org/stable/2748507.

[20] *Response Bias: Definition, 6 Types, Examples & More (Updated)*, en-US, Jun. 2019. [Online]. Available: https://www.nextiva.com/blog/response-bias.html.

[21] *Blackboard Learn*. [Online]. Available: https://ntnu.blackboard.com/?new_loc=%2Fultra%2Finstitution-page.

[22] *Trello Guides: Help Getting Started With Trello | Trello*. [Online]. Available: https://trello.com/guide.

[23] *Planifiez vos projets de A à Z*. fr. [Online]. Available: https://monday.com/lang/fr/project-management.

[24] *Piazza*. [Online]. Available: https://piazza.com/.

[25] *Team compass by weekdone*. [Online]. Available: https://teamcompass.co/.

[26] H. Snyder, 'Literature review as a research methodology: An overview and guidelines,' en, *Journal of Business Research*, vol. 104, pp. 333–339, Nov. 2019, ISSN: 0148-2963. DOI: 10.1016/j.jbusres.2019.07.039. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0148296319304564.

[27] *Researching Information Systems and Computing by Oates, Briony J. 9781446235447R180. Digital bok - 2015 | Akademika.no*. [Online]. Available: `https://www.akademika.no/9781446235447r180/researching-information-systems-and-computing`.

[28] B. Tognazzini, *Tog on Interface*. Addison-Wesley, 1992.

[29] *Why you only need to test with 5 users*. [Online]. Available: `https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/`.

Appendix A

NSD Application

NSD NORSK SENTER FOR FORSKNINGSDATA

Meldeskjema

Referansenummer

126171

Hvilke personopplysninger skal du behandle?

- Navn (også ved signatur/samtykke)
- E-postadresse, IP-adresse eller annen nettidentifikator
- Bilder eller videoopptak av personer
- Lydopptak av personer
- Gps eller andre lokaliseringsdata (elektroniske spor)
- Bakgrunnsopplysninger som vil kunne identifisere en person

Beskriv hvilke bakgrunnsopplysninger du skal behandle

Navn på personene som benytter seg av applikasjonen, samt en score (tall fra 1 til 5 om status på gruppen).

Prosjektinformasjon

Prosjekttittel

Team Accelerator

Prosjektbeskrivelse

En web-applikasjon for å monitorere status på grupper som jobber i prosjekt relatert til akademisk arbeid. Dette er en masteroppgave ved NTNU hvor det skal utprøves ved et emne til våren 2022. Studentene blir bedt om ukentlig å gi en score fra 1 til 5, som beskriver hvor bra ting går i gruppen. Målet med applikasjonen vil være å fange opp grupper hvor ting ikke går så bra, så tidlig som mulig. Av personlige data som samles inn, så vil det være navn (brukernavn fra Feide), og denne scoren studentene avgir. Det vil også bli gjennomført brukertesting av applikasjonen, hvor man samler inn data av intervjuene i form av video med lyd, skjermopptak, og tilbakemeldinger gitt av testerne. Prosjektet vil kun vare fra januar til mai/juni 2022. Etter dette vil dataen bli anonymisert. Det som ikke er mulig å anonymisere vil bli slettet.

Begrunn behovet for å behandle personopplysningene

Det vil være helt nødvendig for applikasjonen å teste ut prosjektet med personopplysninger ettersom det skal brukes i en reell setting ved NTNU. For at foreleserne/emneansvarlige skal kunne ta kontakt med den aktuelle gruppen, må man ha samlet inn data på hvilke studenter som tilhører hvilken gruppe.

Ekstern finansiering

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Henrik Backer, henrikmb@stud.ntnu.no, tlf: 97109963

Behandlingsansvar

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for informasjonsteknologi og elektroteknikk (IE) / Institutt for datateknologi og informatikk

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

George Adrian Stoica, stoica@ntnu.no, tlf: 73412088

Skal behandlingsansvaret deles med andre institusjoner (felles behandlingsansvarlige)?

Nei

Utvalg 1

Beskriv utvalget

Studenter som gjennomfører emnet 'Human Computer Interactions' ved NTNU våren 2022.

Rekruttering eller trekking av utvalget

Alle i utvalget er studenter som gjennomfører et fag som undervises av veillederen ved masterprosjektet.

Alder

19 - 35

Inngår det voksne (18 år +) i utvalget som ikke kan samtykke selv?

Nei

Personopplysninger for utvalg 1

- Navn (også ved signatur/samtykke)
- E-postadresse, IP-adresse eller annen nettidentifikator
- Bilder eller videoopptak av personer
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Hvordan samler du inn data fra utvalg 1?

Ikke-deltakende observasjon

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Samtykke (art. 6 nr. 1 bokstav a)

Felteksperiment/feltintervensjon

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Samtykke (art. 6 nr. 1 bokstav a)

Personlig intervju

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Samtykke (art. 6 nr. 1 bokstav a)

Informasjon for utvalg 1

Informerer du utvalget om behandlingen av opplysningene?

Ja

Hvordan?

Skriftlig informasjon (papir eller elektronisk)

Tredjepersoner

Skal du behandle personopplysninger om tredjepersoner?

Nei

Dokumentasjon

Hvordan dokumenteres samtykkene?

- Elektronisk (e-post, e-skjema, digital signatur)

Hvordan kan samtykket trekkes tilbake?

Ved å gi beskjed enten via epost eller muntlig til enten masterstudent eller veilleder.

Hvordan kan de registrerte få innsyn, rettet eller slettet opplysninger om seg selv?

I applikasjonen vil de registrerte ha innsyn i alle opplysningner om seg selv og ha muligheten til å slette dem.

Totalt antall registrerte i prosjektet

100-999

Tillatelser

Skal du innhente følgende godkjenninger eller tillatelser for prosjektet?

Behandling

Hvor behandles opplysningene?

- Maskinvare tilhørende behandlingsansvarlig institusjon
- Mobile enheter tilhørende behandlingsansvarlig institusjon

Hvem behandler/har tilgang til opplysningene?

- Prosjektansvarlig
- Student (studentprosjekt)

Tilgjengeliggjøres opplysningene utenfor EU/EØS til en tredjestat eller internasjonal organisasjon?

Nei

Sikkerhet

Oppbevares personopplysningene atskilt fra øvrige data (koblingsnøkkel)?

Nei

Begrunn hvorfor personopplysningene oppbevares sammen med de øvrige opplysningene

Personopplysningene som lagres i dette prosjektet er kun navn og en gitt score (fra 1 til 5). Både størrelsen på prosjektet, tidsperspektivet og graden av sensitiv informasjon er grunnen til dette.

Hvilke tekniske og fysiske tiltak sikrer personopplysningene?

- Adgangsbegrensning
- Opplysningene anonymiseres fortløpende

Varighet

Prosjektperiode

03.01.2022 - 01.06.2022

Skal data med personopplysninger oppbevares utover prosjektperioden?

Nei, data vil bli oppbevart uten personopplysninger (anonymisering)

Hvilke anonymiseringstiltak vil bli foretatt?

- Lyd- eller bildeopptak slettes
- Personidentifiserbare opplysninger fjernes, omskrives eller grovkategoriseres

Vil de registrerte kunne identifiseres (direkte eller indirekte) i oppgave/avhandling/øvrige publikasjoner fra prosjektet?

Nei

Tilleggsopplysninger

H. Backer, R. Lamy

Team Accelerator 2.0

NTNU
Norwegian University of
Science and Technology