UNIVERSITETET I AGDER

-

IS-304: 2020

KnowItAll -Treningsapp

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder	Devendra Bahadur Thapa
Oppdragsgiver:	Knowit Sør

-

Students:

Etternavn	Fornavn
Østrem	Filip
Zakariassen	Robert
Fensbekk	Herman
Jørgensen	Anette
Sjøberg Sveen	Håkon

-

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA _X_	NEI
Kan besvarelsen brukes til undervisningsformål?	JA _X_	NEI
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA _X_	NEI

Preface

It has been a hectic semester with a worldwide pandemic, closed schools and home offices, but we made it to the end with success. There are some people that have helped us succeed with the results we have gotten and we would like to give them some recognition here.

First off we would like to thank Knowit Sør for giving us the chance and opportunities to work on this project with them. Knowit Sør has given the group good guidance and helped us reach the best possible results. They have also given us a good work space, included us in social activities at the office and given us overall a warm welcome and introduction into life at Knowit Sør.

We would like to give an extra thank you to Erling Limm, our supervisor and mentor for product development. He has given us guidance to develop good quality front-end code with positive and structural feedback, as well as helping the group out when we were stuck or unsure of best practises in front-end development.

We would also like to give recognition to our mentor and the man responsible for this bachelor project, Tor Oskar Wilhelmsen. Mentors and project leaders with his competence is hard to come by and we appreciate everything he has done for the group.

Our supervisor from UiA, Devendra Bahadur Thapa, thank you for always being available for steering committee meetings and giving us constructive feedback on what we presented and for what we had done so far in the project.

At last we want to thank our families and significant others for the support through this project. Having a family member or significant other sitting in front of their computer every day for several hours isnt easy and we'd like to thank them for supporting and keeping us sane throughout this semester.

Abstract

The bachelor project is offered by Knowit as a part of a larger project for ØIF Arendal. As part of their improvement strategy, ØIF wants a web application used for collecting data from the players after training sessions and matches. The data is to be used for trend analysis. Knowit Sør, who is sponsoring this project, tasked this bachelor group for creating the user interface of this application.

The aim for this project is to create a user interface that allows the players to receive questionnaires, answer the questions and submit the answers. In addition, the users need to be able to register as users and log in to the application. The application will be used as proof-of-concept before any further development of the entire product.

The project was managed according to the Scrum framework. The project planning phase included creating a project plan which included project description and objectives, schedule and cost management, quality management, risk management and scope. The project design phase resulted in an initial prototype that was approved by ØIF Arendal. The implementation phase involved implementing the user stories into a functional front-end application of high quality. In total, the bachelor group completed 6 sprints over the spring semester 2020.

The technology that was chosen for the project was the programming language Javascript with React as framework and Cascading Style Sheets (CSS) to create a user friendly and interactive prototype. Jest and Enzyme for testing and maintaining proper code standards and functionality.

The result of the bachelor project was a functional proof-of-concept application providing the users with an interface to register as a user, log in to the application and answer the questions. All user stories were implemented, and the product complies with the requirements set as the start of the project. The Scrum framework was followed successfully throughout the entire project.

TABLE OF CONTENTS

1. Introduction

At University of Agder, all students in the final semester of the IT and Information Systems bachelor program participate in planning, implementing and completing a bachelor project offered to the students by a real-life company. The students will organize themselves in teams and collaborate with the company to develop a product or an application that will be presented to UiA at the final exams.

The project takes place at Knowit Sør (https://www.knowit.no/), which is an IT consultancy firm with offices at Kristiansand and Arendal. Knowit Sør offers software systems development services to a variety of customers both in the public and private sector. Knowit Sør have been asked to develop an application to be used for gathering data related to training and matches undertaken by players at the ØIF handball team. The idea is that after every match and every training session, the handball players will use the app to receive a set of questions chosen by the administration and coaches, answer these questions and submit the answers to a database. This data will later be used for trend analysis to improve the team's and players individuals overall performance.

The bachelor project consists of creating the user interface for the answer-questions functionality of the app. The interface should allow players to register a user, log in to the app and find the questions that needed to be answered. After answering all questions, the players should be able to submit the answers and go back to the front page of the app as well as logging out of the application. The time taken to answer each questionnaire should be less than a minute.

The bachelor group for this project consists of five students, all with interest in software systems development and programming and a desire to improve as developers throughout the bachelor project. For these reasons, it was natural to apply for a bachelor project at a software systems development company.

2. Central Decisions

At the start of the project, there were several central decisions that had to be made to be able to manage and run the project successfully. These decisions included project methodology, quality management, the technology stack, the communication and collaboration tools.

2.1. Project Methodology

It was a requirement from Knowit that the project should be agile and run according to Scrum or Kanban. We chose to manage the project according to Scrum because we found that the Scrum framework would be the most suitable for this project. This is because Scrum is a more structured agile approach with timeboxed sprints, sprint events and artefacts compared to Kanban that has fewer rules and is based upon a continuous workflow (Rehkopf, n.d.) The more structured approach of Scrum would be helpful given that the group did not have a lot of experience with software development projects.

The Scrum team consisted of a Scrum Master, a Product Owner and the Development Team. The Scrum master responsibilities were to make sure the Scrum process was followed, arrange sprint planning meetings, sprint retrospective meetings and sprint review meetings. In addition, the scrum master was responsible for the burndown chart.

The work process consisted of sprint planning meetings at the start of every sprint, a time-boxed sprint and sprint retrospective meetings and sprint review meetings at the end of the sprints. During the sprint, the group members participated in standup meetings.

2.2. Quality Management

For this bachelor project, it was important for the group to develop a product of high quality to the end-users and to the client. To plan for quality, a project quality plan was created as part of the project planning process.

2.2.1. Quality

A product with high quality for the end-users was defined as a product that the users find easy and quick to use and which offers the desired functionality to the users. In addition, the users should be able to answer the questions given in less than a minute. To be able to know if the product complied with this definition, a set of quality requirements were established. These requirements were based upon information given by ØIF and Knowit. The group also created a strategy on how to ensure these requirements were met.

A product with high quality for Knowit was defined as a product which complied with the requirements given by Knowit in the bachelor project description. These requirements included testing, version control and the predefined technology stack to be used. To ensure the group would meet these requirements, standards on testing, version control and coding was created and followed throughout the project. The project quality plan can be found in appendix 1.

2.2.2. Version Control

For this project, the group used Git (https://www.git-scm.com/) as version control software. The reason for choosing Git was that Git was well integrated with Bitbucket and Jira. In Jira, the group member would choose a task from the current sprint and create a new branch in Git for that specific task. Bitbucket is automatically updated, meaning all group members are allowed to pull the new branch created onto a local machine and start working on the task. When a group member is done with a task, the code is pushed onto Bitbucket for code review. If approved, the branch will be merged into the master branch.

To ensure that all group members did proper and correct version control, the group decided to create a Git guide to be followed. This was necessary because none of the group members had a lot of experience in using Git and there was a high risk of making mistakes. The group knew from experience in other projects that these mistakes could be difficult to fix and could potentially delay work progress. In the worst case scenario, a severe mistake could destroy the entire code and the group would have had to start coding from scratch. The idea was that by having a procedure to follow, each group member would know what to do in various version control scenarios and therefore avoid making mistakes due to lack of knowledge. The version control guide can be found as part of the Project Quality plan in appendix 1.

Establishing routines for version control was a very important contribution to the quality of the end-product and the progress of the project. There was a significant risk for making time-consuming mistakes because of Git inexperience, and the routine helped the group avoid the risk altogether.

2.2.3. Code Standards

To ensure quality in code, the group decided to create a code standard as part of the project planning process. The code standard covered topics such as variable naming conventions and code structure. As a result, the code written was uniform, readable for all group members and of correct structure. In addition, the code standard was used as reference during code review, ensuring code quality in the project master branch. The code standard can be found as part of the project quality plan in appendix 1. Examples of code according to code standard can be found in appendix 2

2.2.4. Unit testing

A requirement given by Knowit was that all production code needed to have corresponding unit tests. Unit testing can be defined as “a level of software testing where individual units or components of a software are tested” (Malikkides, n.d.). Unit tests ensure that each unit of the program performs as intended, which in turn increases the quality of the end-product. In addition to unit testing, each user story needed to pass an acceptance test to be defined as finished. These acceptance tests communicated to the developer what needed to be complete in order for the user story to be complete.

As a part of the project planning, the group decided to add a test procedure-section to the project quality plan. The procedure defined the testing requirements for this project, the technology to be used for testing and a list of what to test in the components. The reason for creating the procedure was that the group wanted to make sure everyone agreed upon the testing requirements and knew what types of unit tests that were expected to be made. The procedure can be found as part of the project quality plan in appendix 1. Examples of unit tests can be found in appendix 3

The test routines were useful to the project, because they made the group focus on testing very early in the project. At the start of the project, all group members were busy learning React and JavaScript, and learning Jest and Enzyme could easily have been postponed until later in the project. If so, there would have been higher risks of not detecting bugs until late in the project, not detecting bugs introduced to the code through code changes and a higher risk of forgetting to test certain code units. The result was a code with good test coverage that complied with the requirements given by Knowit.

2.4. Technology Stack

The product were built on the following technology stack:

- JavaScript as a programming language.
- React as front-end framework.
- Cascading Style Sheets (CSS) as design language for UI design.
- Jest and Enzyme for testing

Using JavaScript and React was a requirement given by Knowit, since this was the technology they would be using when completing the application. However, the project group would probably have made the same choice, given that there are not many proper alternatives to JavaScript on the market. According to Rouz, JavaScript is the one-of-a-kind client side programming language (Rouz, 2019). There are alternatives such as TypeScript and Dart, but these have drawbacks of having limited resources online which would make it difficult to find solutions to problems encountered. Also, given that Knowit has expertise in JavaScript and React, it would be natural to choose a technology stack where Knowit would be able to offer help and support if needed.

CSS was chosen as the design language because it is a well-known language which is reusable and easy to maintain (Bradley, 2006). A change to an element in CSS would lead to a change in all identical elements in the web application. In addition, there are numerous tutorials and learning resources online which is a major benefit given that the project group has little experience with UI design.

Jest and Enzyme were chosen as testing tools as they are well suited for applications made with React. Both Jest and Enzyme are specifically designed to test React applications. Jest can be used with any other Javascript app but Enzyme only works with React (Fraser, 2018). Jest was used to run tests on the application with Enzyme which made it easier to assert and manipulate test data. There were several resources online which were used, as no group members had any prior experience to testing. Knowit could also help and provide tips which was a huge advantage.

2.5. Other central decisions

To complete the bachelor project successfully, it was important for the project group to make use of project tools that would help us plan, execute and control all aspects of the project process. These tools included a project management software and tools for collaboration and communication.

2.5.1. Project Management Software - JIRA

A project management software is a software used for project planning, resource allocation and scheduling of work tasks. In addition, it facilitates budget control, quality management and team collaboration (Haije, 2019). For this project, we used Jira (https://www.atlassian.com/software/jira), which is a software management tool created to support agile software development projects.

Jira offers a variety of tools to help manage projects, and it was not necessary to use them all for this project. We used the following features of Jira:

- Scrum board with customizable workflows.
- Time estimation and work logging.
- Project backlogs.
- Burndown charts.
- Collaboration with BitBucket

A picture of the scrum board can be found in appendix 4. This is the scrum board from sprint 6, and it did therefore not include any major programming tasks.

2.5.2. Communication Tools

The group also agreed upon the tools to be used for working on assignments and long-distance communication. These tools were:

- Slack for messaging and text communication
- Google Drive to collaborate on writing reports and other documents
- Zoom for online meetings

3. Running the Project

In total, one pre-sprint and six sprints were completed during this project. This section of the report will describe the most important aspects of this process.

3.1. Planning and Analysis

The project began with a pre-sprint where the group focused on defining project scope in the form of user stories and creating a project plan.

3.1.1. Project Scope - User stories

The bachelor project is part of a larger project, and because of this, the information gathering and requirements analysis had already been undertaken by Knowit. The group was given this information at the start of the project, and defined the project scope based upon this knowledge.

The project scope was defined by a set of user stories. A user story is a short story that describes a goal that the user has when he or she is using the system (Satzinger, Jackson & Burd, 2016, p.71) In total, seven user stories were created during project planning. Each of these user stories were given acceptance criteria and an acceptance test. To ensure that the project group would always be working on the most important feature, the user stories were prioritized according to the Moscow method. The Moscow method is a technique for defining and managing requirements and tasks in a project (Madsen, 2019). The user stories were added to the project's product backlog. Examples of the user stories for this project can be found in appendix 5.

3.1.2. Project Plan

The group created a project plan to ensure the project would be well-managed and result in a product according to client expectations. The project plan included a project schedule and a project budget, risk management, quality management, the user stories and an initial product backlog. The product backlog was added to Jira during the first sprint planning meeting, and was later maintained in Jira and not in the project plan. An extract of the project plan can be found in appendix 6.

By creating a project plan, the group reached a common understanding of important aspects such as project scope, quality and risk management. The common understanding was one of the success factors for this project because by having a plan and strategies to follow, we were all able to work in the same direction to achieve the desired outcome.

3.3. Design

The aim during the design phase of the project was to create a user interface with high usability so that it would be easy and quick to use for the handball players. In addition, a design that would appear modern and desirable for the users. To achieve this, it was important to create a functional prototype to be shown to ØIF so that ØIF could provide feedback.

3.3.1 Sketches

In early stages of the project, sketches were made to display how the application could look before getting into any prototyping or coding. Sketching is the basic of user interface and helps to translate the idea to interface in the minimum possible time. (Costa, 2019).
There were sketches created for every aspect of the application from registering a user, login page, the main page with the questionnaires and how the questionnaires were supposed to look like when a user goes through the questions. All the necessary pages to have a working application for our project in sketches. Examples of the sketches made can be found in appendix 7.

3.3.2 User experience(UX) Prototype

A prototype is a "A simulation or sample of version of a final product, which is used for testing prior to launch. "The goal of a prototype is to test products (and ideas) before sinking lots of time and money into the final product. (Uxpin, n.d) Based on the sketches, a functional UX prototype was made with the web application UXPin. Pictures of the UX prototype can be found in appendix 8.

With this tool it was possible to create a prototype where the users are able to navigate and have an understanding on how the application could look like. The goal with this prototype was to show it to ØIF Arendal and to see if the design was functional with the aim to answer the questions within a minute.

This prototype was useful when the group had a meeting with the client Knowit and ØIF Arendal which are the consumers that are going to use the application in the end. In this meeting our group showed the prototype and got feedback from both parties on what they thought about it and some minor improvements to the application. Overall they were satisfied with the prototype and it was approved.

3.4. Implementation

Once the UX prototype was approved by ØIF and Knowit, the implementation phase of the project began. None of the group members had any past experience with JavaScript or React, and to overcome this obstacle everyone had studied tutorials on their own to be prepared for the implementation phase. Despite these measures, the group soon realised that we still had a lot to learn, and most of sprint 2 was spent studying React and JavaScript instead of doing any actual coding.

At the start of sprint 3, the group had reached sufficient knowledge about the technology and the implementation was progressing as planned. At the end of the first week, the coronavirus caused Norway to shut down and as a consequence the group had to continue working from home offices instead of meeting face to face. Despite this challenge, the group managed to continue collaborating as a team and managed to complete all the tasks in the sprint backlog.

Sprint 4 continued with implementing the user stories. The group did not face any particular problems and managed to keep a steady progress throughout the sprint. At the end of the sprint, the group asked Knowit for feedback on the code written and on the application itself. The feedback received was added to the product backlog and further to the sprint backlog in sprint 5. Sprint 5 focused on finalizing the remaining must-have user stories and improving the application according to the feedback received from Knowit. The result of sprint 5 was a completed user interface for receiving and answering questions, and the bachelor project task could be considered completed.

The final sprint was sprint 6, which was dedicated to project wrap-up, including finalizing bachelor report and minor improvements to the code and application.

3.5. Schedule and Cost Management

The project start date was 13 January 2020, and the end date was determined to be 29 May 2020. The project also had milestones that had to be met in order to complete the project successfully. These milestones were:

- week 7: steering committee meeting
- week 14: steering committee meeting
- week 18: steering committee meeting
- week 21: project report hand - in

To get an overview of the project, the group created a timeline visualising the sprints for this project. The first timeline version consisted of 9 sprints, with a duration of two weeks on each sprint. The duration of sprints were changed from two weeks to three weeks. This change was based upon feedback given on Steering Committee Meeting no. 1. The revised timeline consisted of 6 sprints, with a duration of 3 weeks on each sprint. The timelines can be found in appendix 9.

The timeline provided the group with a rough estimate on what activities should take place each sprint. For example, would sprint 3 include implementation activities only while sprint 5 would include both implementation and preparation of the bachelor report.

For this project, the costs will consist of labour hours for each group member. It was estimated that each group member would have 5 hours a day available for the IS-304 subject, giving the project a total of 2500 working hours available.

The estimated hours for each task and actual hours spent was monitored via Jira. During the sprint planning meeting, the group would give a time estimate on each task added to the sprint backlog. During the sprint, each group member would log working hours spent on the various tasks, providing the Scrum master with a burndown chart that would give an overview of the status.

3.6. Communication

For the bachelor project to be well-managed, it was important for the group to ensure a good communication process to avoid misunderstandings and delays. This included both the internal communication processes within the team and the external communication processes between the team and the various project stakeholders.

Based upon experience from former projects, the group members knew that good internal communication flow was a very important successfactor for the project. Good communication would ensure that everyone was working in the same direction, that everyone knew the project’s progress and that any problems or potential delays would be discovered as soon as possible. Organizing the project according to the Scrum framework had a high influence on the internal communication processes within the group. The Agile Manifesto emphasise the value of individuals and interactions over processes and tools. (Martin.R, 2014, p.4). For this reason, the group decided that the main communication strategy for internal communication would be face-to-face communication while working together as a team at the office. Meeting face-to-face gave the opportunity to discuss problems as they occurred, to bring up new ideas and provide each other instant feedback.

The coronavirus-outbreak caused all internal communication to take place online instead of face-to-face. This did not have any major impact on the internal communication process itself, as the group still focused on having daily communication with each other through tools like Slack and Zoom. However, even with focus on good communication, the group still found it challenging not having the opportunity to meet face-to-face. In particular, all collaboration regarding working with code and debugging became more time consuming and less efficient compared to working face-to-face. For example, if one group member was having trouble with a bug, he or she could no longer ask someone to quickly come over and take a look. Instead, he or she would need to ask the other team member to join a Zoom meeting, start screen sharing and explain the trouble through the screen. The other team member would then have to give advice through the screen without being able to point a finger at the code where the bug might be. This was a rather cumbersome way to collaborate, but there was no other option than to make the best of it and take the extra time into consideration when estimating time for work tasks.

The external communication process has consisted of meetings with the various stakeholders, informal communication at Knowit offices and presentations in class regarding the project. The meetings with the stakeholders included:

- meeting with ØIF and Knowit for information and prototype review
- steering committee meetings with Knowit and UiA supervisor
- status meetings with IS-304 class

The communication with ØIF and Knowit greatly contributed to developing an application that complied with the quality and functionality requirements determined at the start of the project. The communication between Knowit and the bachelor group was usually informal because of shared office space, and this made it easy to get answers or guidance as needed. After the coronavirus shutdown, the communication with Knowit became less frequent and any questions needed to be asked via email. In addition, Knowit employees became very busy taking care of their own clients who needed their expertise for shutdown-related problems. Fortunately, at this point in the project, the group had increased their knowledge and experience about the technologies used, and did not need much guidance for Knowit to get the work done.

3.6. Risk Management

Project risk management is about identifying, analyzing and responding to risks that occur throughout the project lifecycle. A risk could be anything that has a potential to cause impact on the project's timeline, performance or budget (Ray, 2017). Since all projects will have to deal with risks of some kind, was it important for the project's success to plan for risk management when creating the project plan. The risk management process for this project involved risk identification, risk analysis and risk evaluation with the purpose of preparing for identified risks, estimating the impact and defining response strategies to the risks.

The risk identification process identified the different kinds of risks that could occur during the project's lifecycle. The result was a list with 16 identified risks that could have a potential negative impact on the project results. An example of identified risk was the risk of losing work or documents due to computer theft or damaged computer. Another risk identified was the risk of delivering a product that the client would not be content with. All risks were listed in a risk register. An extract of the risk register can be found in appendix 10. The risk register was reviewed and revised as necessary at every sprint planning meeting.

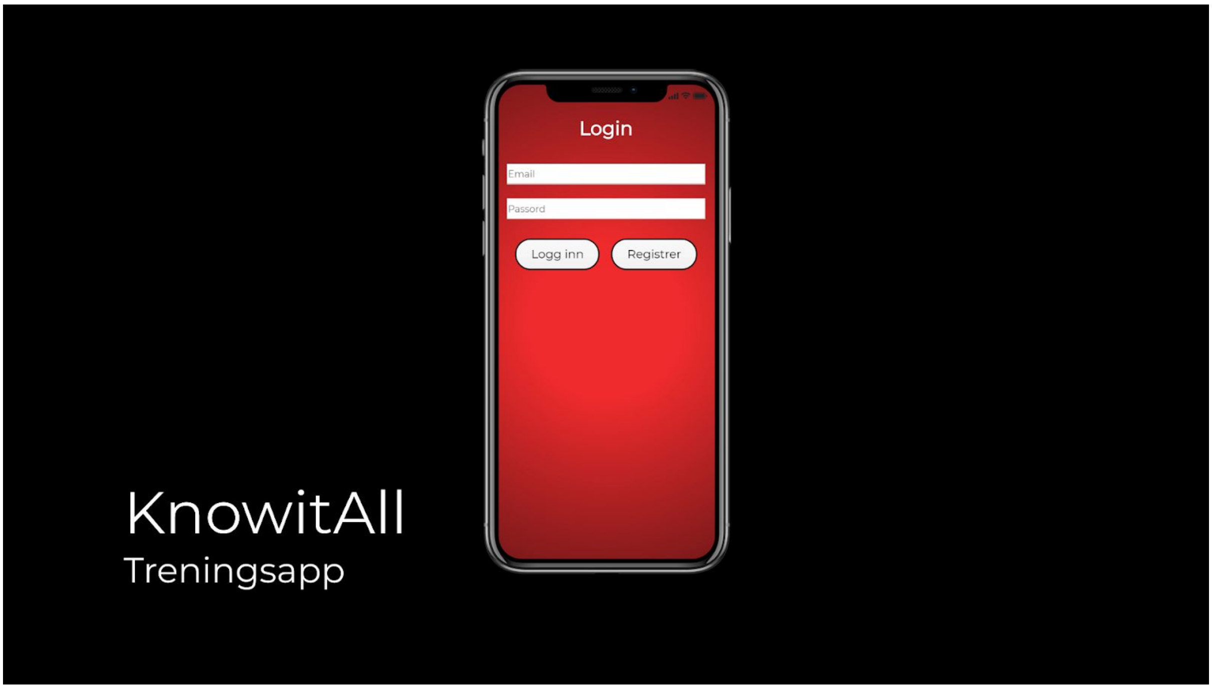
The risks identified were listed into a risk diagram and assessed with a likelihood-, and consequence scale from one to five. Based upon this risk diagram, the group concluded that the risks that needed to be monitored closely where the following:

- Risk 5: Disagreement within the group.
- Risk 9: Lose motivation.
- Risk 10: Customer not satisfied with product.
- Risk 14: Failure to use Git/version control properly.
- Risk 15: Failing to test and have poor test coverage.
- Risk 16: Failing to deliver a working product.

To avoid these risks from happening, the group followed the risk strategies created, and by sprint 5 these risks had been reduced significantly. The risk diagrams created are found in appendix 11.

4. The Product

Below is a link to a video that demonstrates the application.



Link to the video: https://vimeo.com/420351521

5. Reflections

This section of the report will reflect upon several aspects of the bachelor project, including the project management process, the quality of the end product and the risk management process.

5.1. Project Management

Following the Scrum principles, the project group was self-organized and did not have a formal project manager appointed. For each sprint planning meeting, the group planned ahead for the next three weeks, time estimated each task and made sure the estimated labour hours did not exceed the budgeted hours. The work tasks added to the sprint backlog were the tasks with the highest priority in the product backlog, making sure the group was always working on the most important functionalities.

After every sprint, the Scrum master held a sprint review meeting and a sprint retrospective meeting, allowing the group to review the work done during the sprint and provide each other feedback on what went well during the sprint and what could be improved. This ensured a continuous improvement of work process and quality throughout the project.

The group experienced several benefits of using Scrum:

- planning for only three weeks at a time made the planning more accurate. It was easier to estimate how much work we could accomplish for the next three weeks compared to estimating the entire project work tasks at once.
- looking at the burndown charts throughout the sprint ensured that worked hours did not exceed estimated hours, ensuring cost control with regards to labour hours.
- the sprint events highly influenced the communication processes within the group. Especially the sprint retrospective meetings were very useful because of the focus on giving feedback on positive and negative aspects of the past sprint.
- it became easy to respond to changes and challenges. Only planning for three weeks at a time made it possible to make changes as necessary every three weeks, and ensured that we did not waste any time on planning something that in the end did not happen. This became particularly important at the coronavirus shutdown, since the group had to change work routines altogether and had to find new ways to collaborate to ensure progress in the project.

During this bachelor project, the group has gained a deeper understanding of the Scrum framework and how it can be used to manage software development projects. All group members had experience with Scrum from other projects at UiA, but this was the first project where the group was completely self-organized working for a real client that had expectations and requirements for the end product.

5.2. Quality of End Product

For this bachelor project, it was important to develop a product of high quality to the end-users and to the client. To plan for quality, a project quality plan was created as part of the project planning process.

A product with high quality for the end-users was defined as a product that the users find easy and quick to use and which offers the desired functionality to the users. To be able to know if the product complied with this definition, a set of quality requirements were established. These requirements were based upon information given by ØIF and Knowit. The group also created a strategy on how to ensure these requirements were met. Overall, the group considers the quality of the end product to be of high quality to the end-users. The product complies with all requirements listed in the project quality plan, contains all desired functionalities and is quick and easy to use.

A product with high quality for Knowit was defined as a product which complied with the requirements given by Knowit in the bachelor project description. These requirements included testing, version control and the predefined technology stack to be used. To ensure the group would meet these requirements, standards on testing, version control and coding was created and followed throughout the project. The result was a product that complied with the requirements given by Knowit. Throughout the project, Knowit has been asked to review both the code written and the design of the product and provide the group with feedback on how to improve. This feedback has been overall good, with only minor suggestions for improvements. The group believes that the end product is of good quality to the client. This because the product complies with all requirements given at the start of the project, and because the feedback from Knowit was positive both on code quality and on design.

There are two main reasons for successfully achieving good quality in the end-product. The first reason is that the group planned for quality at the beginning of the project and then stayed focused on quality throughout the project. The second reason is that the group focused on close communication with our clients and end-users. This communication led to a good understanding of the user needs and the client expectations. The feedback received from Knowit helped the group refactor and improve the product to better suit the client.

5.3. Risk Management

The group considers the risk management process for this project to have been executed in a good manner. By having discussions on possible risks early in the project, the group reached an awareness about what to do to handle the risks. This was particularly important when the virus-outbreak occurred, because even though the shutdown came very sudden and as a surprise to us all, we still had some idea on how to handle the situation and we avoided panic and chaos. It was also useful to revise the risk register at every sprint planning meeting, because even though we did not always make changes to the register, it ensured that the project risks were evaluated on a regular basis.

When planning the risk management, the group identified six risks that needed to be monitored closely. During the project, these risks were mitigated by increased experience in the technology used, procedures and standards to follow, by focusing on good communication with the client and by doing our best to have a good working environment within the group. Without the risk management planning, we believe there would have been less awareness about these risks and avoiding the risks would have happened by chance and not by risk management control.

In retrospect, we see that some of the risks first identified were not really realistic risks that had to be dealt with. For example, one of the risks was the risk of Knowit going bankrupt. Considering that Knowit is a well established company with big profit, this risk was highly unlikely to happen and did not really need to be listed in the register.

5.4. Challenges

A bachelor project should be challenging by nature, and this project is no exception. Throughout the project, there were in particular three major challenges that had to be dealt with. These challenges were lack of experience with technology stack, sprint planning and time estimation and the coronavirus outbreak with the following shutdown.

5.4.1. Lack of Experience with Technology Stack

None of the group members had any prior experience with JavaScript and React, which was the technology stack to be used for this project. In addition, none of the group members had done much front-end programming overall. This was a challenge in the first and second sprint because we did not have the basic knowledge to evaluate whether the programming decisions taken were good ones or not. For example, without knowledge of React it was difficult to structure the code according to React best practises. React (https://reactjs.org/) is a component-based JavaScript-library, which means that the user interface is composed of several components that are able to manage their own state and render a view based upon data received and events occurred. This meant that, very early in the implementation phase, the group had to decide on what components should be created, what relationship should exist between the components and the functionality of each component. Throughout the project, these decisions had to be reconsidered and the code structure needed refactoring as we learned better ways of doing it.

At the beginning of the project, the group took several measures to meet this challenge:

- It was decided that every group member had to take on responsibility to learn and understand JavaScript and React basics. To arrive at the best solutions, it was important for everyone to be able to contribute to the discussions and the actual code writing.
- During sprint planning it was allocated time for learning React and JavaScript.
- It was decided that if a group member got stuck, he or she would not spend weeks trying to solve the problem him/herself, but instead ask for help either from other group members or from Knowit.

These measures did help us minimize the consequences of the challenge. Every group member spent a lot of time at the beginning of the project watching React tutorials on Youtube and Udemy, which again increased the programming skills across the group. This leads to everyone being able to participate in technical discussions and contribute to solutions. Also, we were able to help each other out as we ran into problems with the code. With more knowledge and experience, it became easier to see what code structure would be preferable for our project and we made changes accordingly.

However, despite these countermeasures, we still experienced problems. One problem was that it was difficult to know when to ask for help from Knowit and when to keep trying to figure things out ourselves. For example, in sprint 1, a group member was tasked to write code for the login function in the application. To be able to do this task, the group member had to watch several tutorials on the subject. However, the tutorials gave different instructions on how to write the code, and with no prior knowledge of the subject matter it was impossible to know which method would be the most suitable one for this project. As a consequence, the code had to be scratched three times before the end result was correct. A benefit of this trial and error was that the group member gained a lot of experience in React and debugging, while the drawback was a lot of time wasted on code that had to be scratched. If he instead had asked Knowit about which method was the correct one to use at the very beginning, he would have spent less time on the task, but he would have had a smaller learning outcome. All group members had similar dilemmas from time to time, and choosing the right solution was not always obvious.

Another issue was that even though we allocated hours to be spent on tutorials and learning, we found that we had not estimated enough hours. This became very obvious during sprint 2, when progress stagnated altogether because no one knew how to proceed. In the end, we had to stop working on the code and dedicate all remaining hours to watching tutorials and experimenting with code to fully understand all principles behind React. As a consequence, we did not complete any of the tasks we set out to do in sprint 2.

5.4.2. Sprint Planning and Time Estimation

Throughout the first two sprints, it became clear that the group had a challenge with planning the sprints and estimating the time it would take to complete each task in the sprint. The main problem was that the group underestimated how much time it would take to learn React and Javascript and failed to see the complexity of the tasks to be done. As a result, we failed to complete all the tasks we planned to do during the sprints.

Since this problem was closely linked to lack of prior experience, at first the group hoped that increased experience would be enough to tackle this problem as well. Because of this, the group did not take any specific action to improve the situation at first. However, as sprint 2 stagnated completely, it became clear that we needed to improve the planning and time estimation for sprint 3.

To improve the sprint planning and time estimation for sprint 3, the group did the following:

- We asked for advice from Knowit.
- We estimated less tasks to be completed.
- For each task, we estimated approximately 80% of the time to be spent on research and 20% of the time to be spent on actual coding. These numbers were given to us by a Knowit advisor, as these were the numbers they used when planning.
- We started pair programming, meaning that each task was assigned to two group members.

The steps to improve the planning and estimation process were considered to be working well. All tasks in the sprint backlog were completed on time. In addition, the learning progress for each group member was better and faster than expected. More time was spent coding than doing research which resulted in spending less time than planned to finish the tasks. There was still potential for improvement in time estimation. Therefore the group decided to go with the same tactic in the fourth sprint. This time all tasks were finished and the hours estimated were more accurate.

5.4.3. Coronavirus Outbreak

Sprint no. 3 started out very well. The first two days of the sprint, the group managed to complete the login-function, the question schema-function and the switch-view function, and also merge these functions together to a functioning application. On day 4 in the sprint, the Norwegian prime minister declared shutdown of all schools, kindergartens and all other educational institutions in the country to try to mitigate the risks of the Covid-19 outbreak in the country. In addition, every citizen was encouraged to work from home if possible.

This caused several consequences for the project:

- The group was prevented from meeting face to face, instead it was spread throughout the country where the group members worked from home offices.
- We lost face-to-face contact with advisors from Knowit, and all guidance had to be done over the internet.
- UiA got closed, and we were left to communicate with teachers via email.

Based upon these consequences, the project was faced with several challenges:

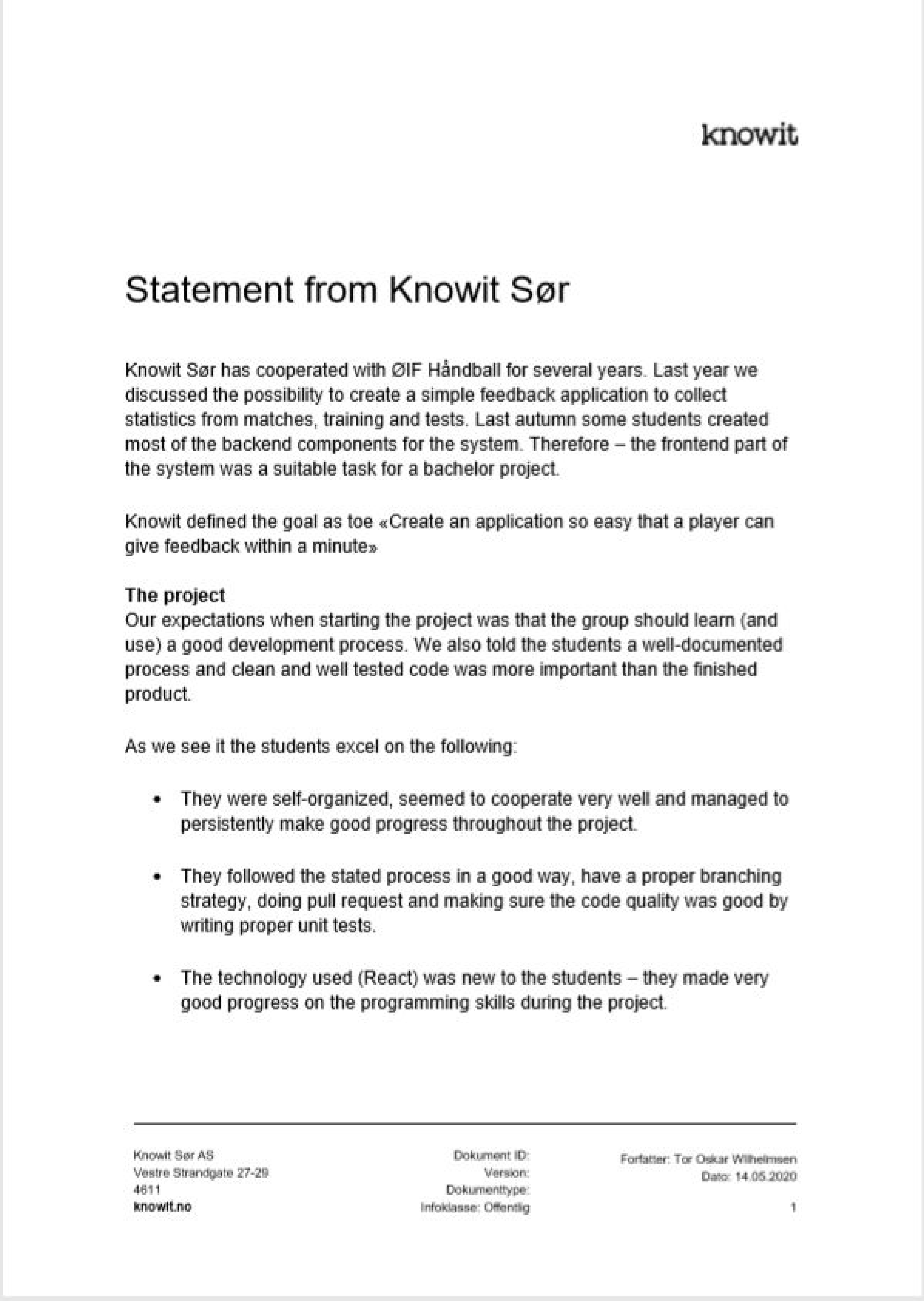
- we had to find a new work routine to be able to cooperate and communicate as a team. Since we were used to meeting and working face-to-face, it was now a challenge to keep up the pace and progress at distance.
- schools closing led to one of the group members having three kids at home for homeschool. As a result, this group member was unable to meet or work on the project during the day, and all meetings with the group had to take place during the afternoons and evenings.
- losing face-to-face contact with our advisor from Knowit caused the coding progress to slow down during the sprint. Being able to talk face to face gave quicker feedback than being dependent on online communication.

To face these challenges, the group had a crisis meeting on March 16th to evaluate the situation and make a plan for the weeks ahead. We performed a risk analysis of the situation, and agreed upon the three major risks for the project where the risks of quarantine, the risk of family members being critical ill/hospitalized and the risk of group members being infected by the virus preventing the infected person from working.

To try to mitigate the risk, we decided that we would all try our best to follow the advice given by FHI to not get infected by the virus. In addition, we decided to have daily scrum every day to make sure we would communicate and know each other’s status on a daily basis. A third strategy was to make sure that no one in the group would be the sole responsible for a task. The idea was that if one person fell ill and was unable to work, then the rest of the group would be able to complete the work instead. This way, progress would be ensured and would not be dependent on the recovery of the infected.

The strategy to mitigate the risks was followed by all team members throughout the project. At report delivery date, no group members have been ill or absent due to the virus. It is however not possible to evaluate for sure whether this is pure luck or because of the strategies laid at the crisis meeting. The consequences of the outbreak was that the work progress of sprint 3 was delayed while the group rearranged work routines and got used to the new everyday life.

6. Statement from Client



knowit

Statement from Knowit Sør

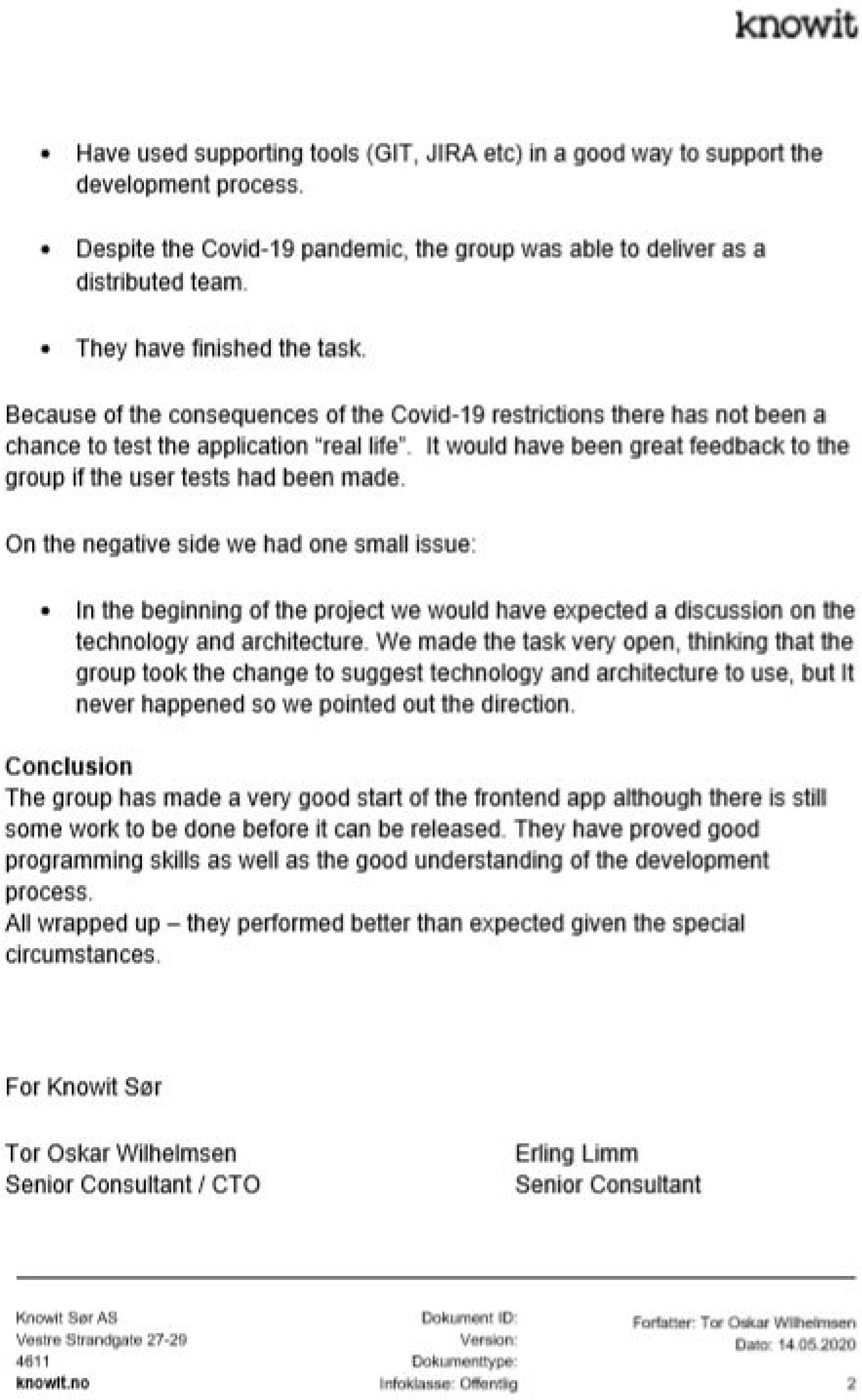
Knowit Sør has cooperated with ØIF Håndball for several years. Last year we discussed the possibility to create a simple feedback application to collect statistics from matches, training and tests. Last autumn some students created most of the backend components for the system. Therefore – the frontend part of the system was a suitable task for a bachelor project.

Knowit defined the goal as toe «Create an application so easy that a player can give feedback within a minute»

The project
Our expectations when starting the project was that the group should learn (and use) a good development process. We also told the students a well-documented process and clean and well tested code was more important than the finished product.

As we see it the students excel on the following:

- They were self-organized, seemed to cooperate very well and managed to persistently make good progress throughout the project.
- They followed the stated process in a good way, have a proper branching strategy, doing pull request and making sure the code quality was good by writing proper unit tests.
- The technology used (React) was new to the students – they made very good progress on the programming skills during the project.



- Have used supporting tools (GIT, JIRA etc) in a good way to support the development process.
- Despite the Covid-19 pandemic, the group was able to deliver as a distributed team.
- They have finished the task.

Because of the consequences of the Covid-19 restrictions there has not been a chance to test the application "real life". It would have been great feedback to the group if the user tests had been made.

On the negative side we had one small issue:

- In the beginning of the project we would have expected a discussion on the technology and architecture. We made the task very open, thinking that the group took the change to suggest technology and architecture to use, but it never happened so we pointed out the direction.

Conclusion
The group has made a very good start of the frontend app although there is still some work to be done before it can be released. They have proved good programming skills as well as the good understanding of the development process.
All wrapped up – they performed better than expected given the special circumstances.

For Knowit Sør

Tor Oskar Wilhelmsen Senior Consultant / CTO	Erling Limm Senior Consultant

7. Self-Evaluation

Anette Jørgensen
At the start of the project, I was involved in all of the project planning activities. I participated in all group meetings and was an active contributor to the creation of the project plan. In particular, I took on responsibility for writing and completing the project quality plan.
During the implementation phase of the project, I learned myself React and JavaScript by watching tutorials and experimenting with code on my own. Together with the other members of the group, I took part in developing the user stories into functional code. In particular, I took responsibility for coding the functionality for how to display the unanswered question-schemas on the front page and the functionality for displaying the questions one-by-one to the user. I also learned unit testing with Jest and Enzyme, wrote several unit tests and ensured good test coverage for the code.
Towards the end of the project, I have contributed to the IS-304 report by writing several sections of the report and by taking responsibility for transcribing the report before the deadline.

Herman Fensbekk
I was involved in the planning of the project and attended all group meetings. In the early phases I was participating in creating user stories and spent a lot of time learning core features of Javascript, React and Node to produce functional code from the user stories. My main responsibility in the implementation phase was to create the backend server / functionality of the web application. I created specific routes with get, post and delete requests with allowed cross-origin resource sharing. This made it possible for us to store users, create users, retrieve and delete questions and save answers from the website that is being interacted with. With the help of Filip we created a login functionality. I also took part in displaying the questions available and what answer has been chosen and then making it possible to be saved and send it to the backend server. I made a functionality in the backend which would either render a textbox or buttons to answers depending on what type of question that would display. I also spent time learning unit-testing with Jest and Enzyme to write unit tests for our code. I have helped with some design elements and writing on the report.

Robert Zakariassen
At the beginning of the project, I was involved in planning the project. I also participated in all the group meetings. I took on responsible for creating a UX prototype of the application that was shown to the client. During the implementation phase I watched courses and tutorials to learn React and Javascript and alongside the tutorials I wrote code. In early phases I participated in creating the user stories, and particularly took responsibility for creating and coding the functionality of the different types of answers opportunities the users had. I also

took the charge of implementing a progress bar for the questionnaire with a counter with help of Anette and Herman to implement it into our application. I also took part in creating and adding the design part with Cascading Style Sheets(css) into the application, both for the web and for mobile phones. I also took responsibility for coding a logout button functionality on the front page that the users could log out. Throughout the project I have also helped with writing the report.

Filip Østrem

At the start of the project, I was involved in all of the project planning activities, group meetings and scrum planning.

Early on i got the role as scrum master and made sure we had sprint planning, sprint retrospective/review and daily scrum.

During implementation I created the login functionality allowing a user to access with a specific key/token only and a routing system that always routes to the right page at the right time as well as route between all the different functionalities such as login, register, dashboard and questionnaire.

Later on I helped with creating various tests around proptypes and focused on creating good and clean Cascading Style Sheets(css)/design and helped write the report.

Håkon Sveen

There have been a lot of different tasks and work during this project and I have participated in almost every group-task along with having the responsibility for a few tasks alone. At the start of the project I took part in the planning process of the project together with the rest of the group and I have attended all of the group meetings. In the planning phase I also took responsibility for sketching out and creating a lo fi prototype for the application on paper. In the implementation of the application I have mainly worked with creating the register page, but I have also worked on other parts of the application like the css, creating type-checking and different tests for the application. I have written some parts in the report along with some proofreading. I have also created a demonstration video of our application that showcases our product.

8. References

Bradley, S (2006, March 2006) The Benefits of Cascading Style Sheets. Retrieved from: https://vanseodesign.com/css/benefits-of-cascading-style-sheets/

Costa, R (2019, Jule) Beginner’s guide to UI sketching. Retrieved from: https://www.justinmind.com/blog/ui-sketching/

Fraser, D (2018, April). Testing React with Jest and Enzyme I. Retrieved from: https://medium.com/codeclan/testing-react-with-jest-and-enzyme-20505fec4675

Haije, E.G. (2019, November 25). Top 20 best project management software: An overview. Retrieved from: https://mopinion.com/top-20-best-project-management-software-an-overview/

Malikkides, O. (n.d). *How to (unit) test in React.*
Retrieved from: https://itnext.io/how-to-unit-test-in-react-72e911e2b8d

Martin, R.C. (2014) *Agile Sjesoftware Development Principles, Patterns and Practices.* (First Edition) Essex: Pearson Education Limited

Ray, S. (2017, October 19) The Risk Management process in Project Management. Retrieved from: https://www.projectmanager.com/blog/risk-management-process-steps

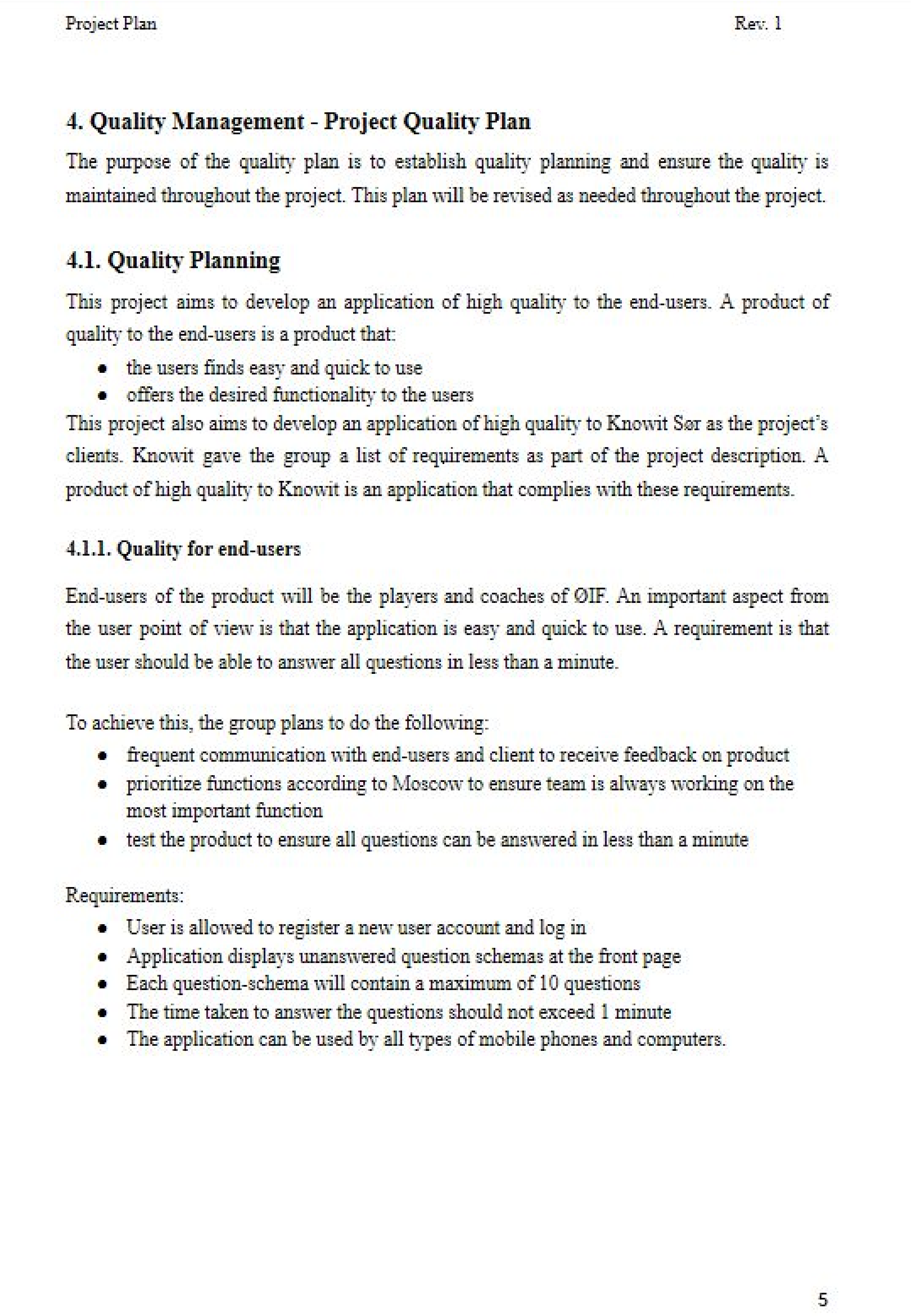
Rehkopf, M (n.d.) Kanban vs Scrum. Retrieved from: https://www.atlassian.com/agile/kanban/kanban-vs-scrum

Rouz, U. (2019, November 21). Is there a viable alternative to javascript?
Retrieved from: https://www.webdesignerdepot.com/2019/11/is-there-a-viable-alternative-to-javascript/

Satzinger, J, Jackson & R, Burd, S (2016) *Systems Analysis and Design in a Changing World.* (7th Edition) Boston: Cengage Learning

Uxpin (n.d) UI Design and Prototyping tool. Retrieved from: https://www.uxpin.com/
Madsen, S (2019, December 3) How to Prioritize with the Moscow Method. Retrieved from: https://www.projectmanager.com/training/prioritize-moscow-technique

Appendix 1 - Project Quality plan



Project Plan Rev. 1

4. Quality Management - Project Quality Plan

The purpose of the quality plan is to establish quality planning and ensure the quality is maintained throughout the project. This plan will be revised as needed throughout the project.

4.1. Quality Planning

This project aims to develop an application of high quality to the end-users. A product of quality to the end-users is a product that:

- the users finds easy and quick to use
- offers the desired functionality to the users

This project also aims to develop an application of high quality to Knowit Sør as the project's clients. Knowit gave the group a list of requirements as part of the project description. A product of high quality to Knowit is an application that complies with these requirements.

4.1.1. Quality for end-users

End-users of the product will be the players and coaches of ØIF. An important aspect from the user point of view is that the application is easy and quick to use. A requirement is that the user should be able to answer all questions in less than a minute.

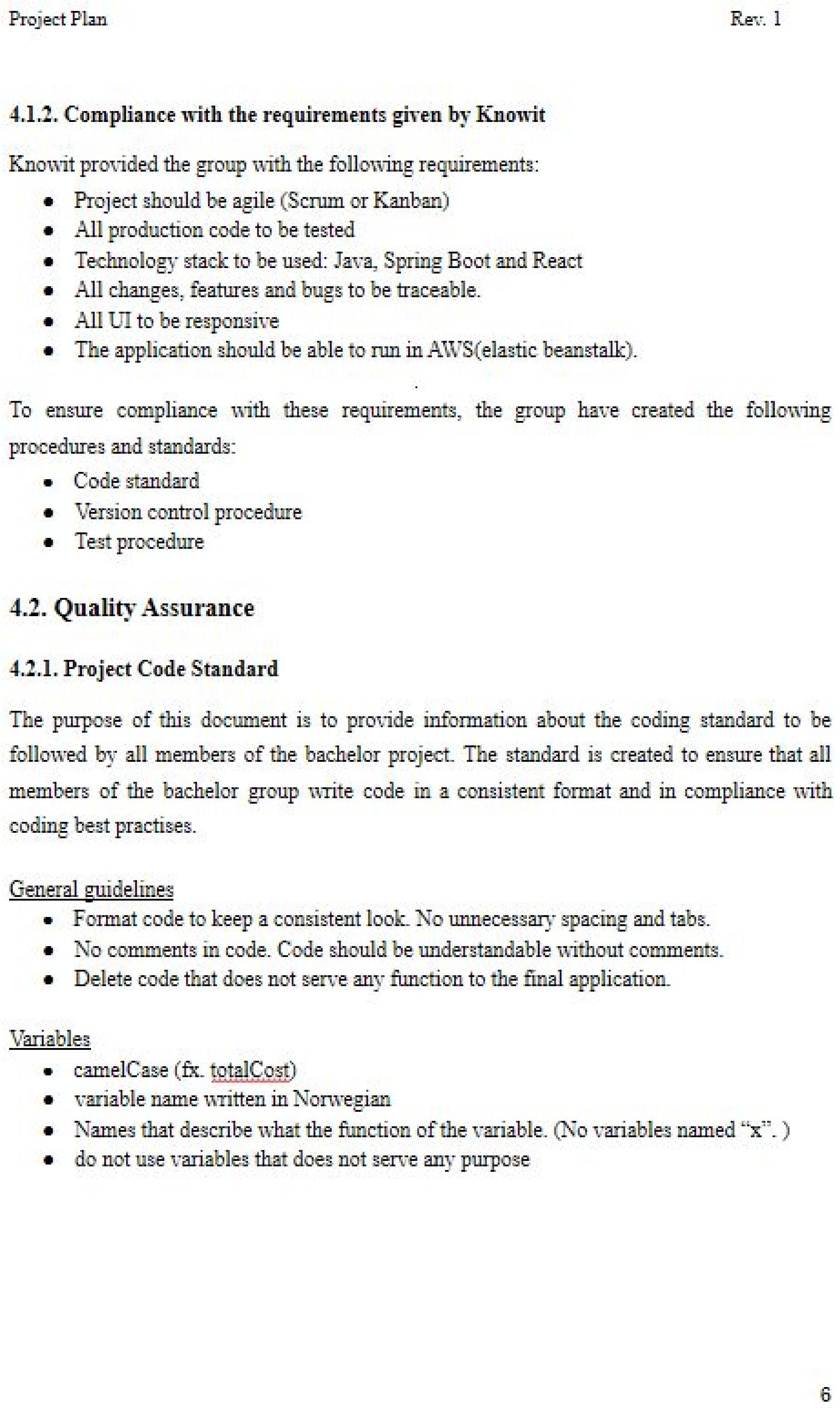
To achieve this, the group plans to do the following:

- frequent communication with end-users and client to receive feedback on product
- prioritize functions according to Moscow to ensure team is always working on the most important function
- test the product to ensure all questions can be answered in less than a minute

Requirements:

- User is allowed to register a new user account and log in
- Application displays unanswered question schemas at the front page
- Each question-schema will contain a maximum of 10 questions
- The time taken to answer the questions should not exceed 1 minute
- The application can be used by all types of mobile phones and computers.

5



4.1.2. Compliance with the requirements given by Knowit

Knowit provided the group with the following requirements:

- Project should be agile (Scrum or Kanban)
- All production code to be tested
- Technology stack to be used: Java, Spring Boot and React
- All changes, features and bugs to be traceable.
- All UI to be responsive
- The application should be able to run in AWS(elastic beanstalk).

To ensure compliance with these requirements, the group have created the following procedures and standards:

- Code standard
- Version control procedure
- Test procedure

4.2. Quality Assurance

4.2.1. Project Code Standard

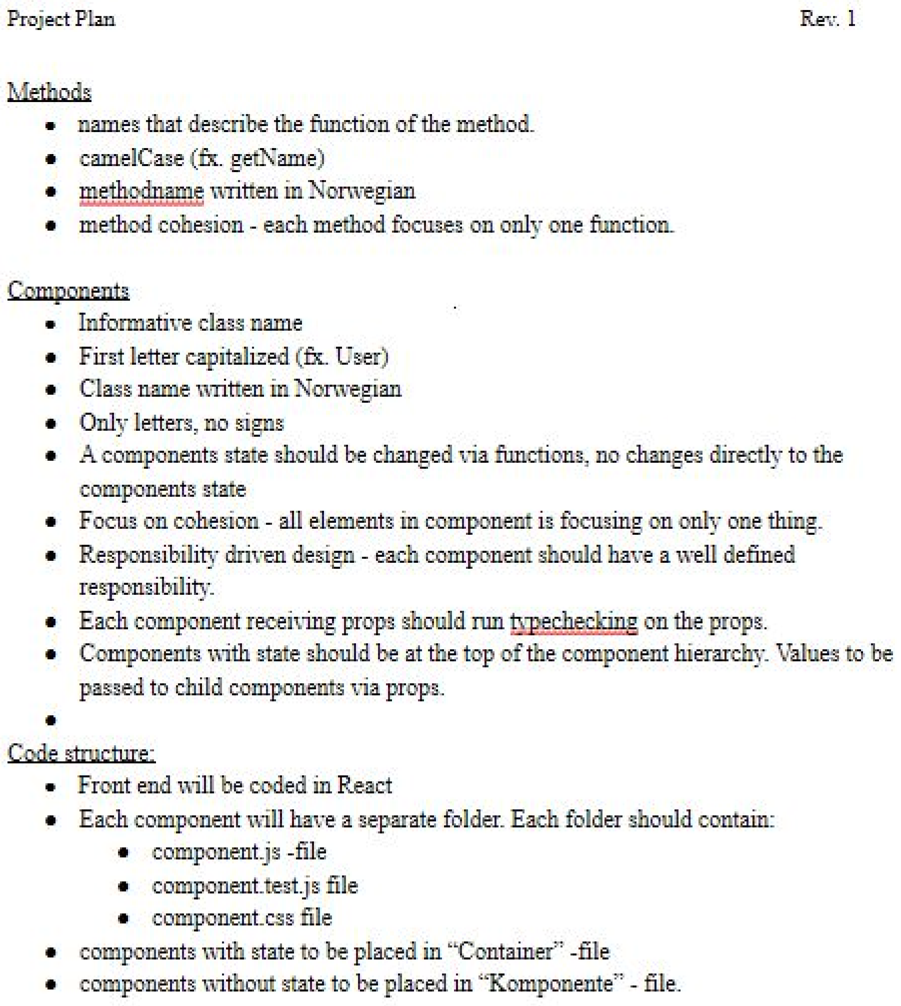
The purpose of this document is to provide information about the coding standard to be followed by all members of the bachelor project. The standard is created to ensure that all members of the bachelor group write code in a consistent format and in compliance with coding best practises.

General guidelines

- Format code to keep a consistent look. No unnecessary spacing and tabs.
- No comments in code. Code should be understandable without comments.
- Delete code that does not serve any function to the final application.

Variables

- camelCase (fx. totalCost)
- variable name written in Norwegian
- Names that describe what the function of the variable. (No variables named "x".)
- do not use variables that does not serve any purpose



Methods

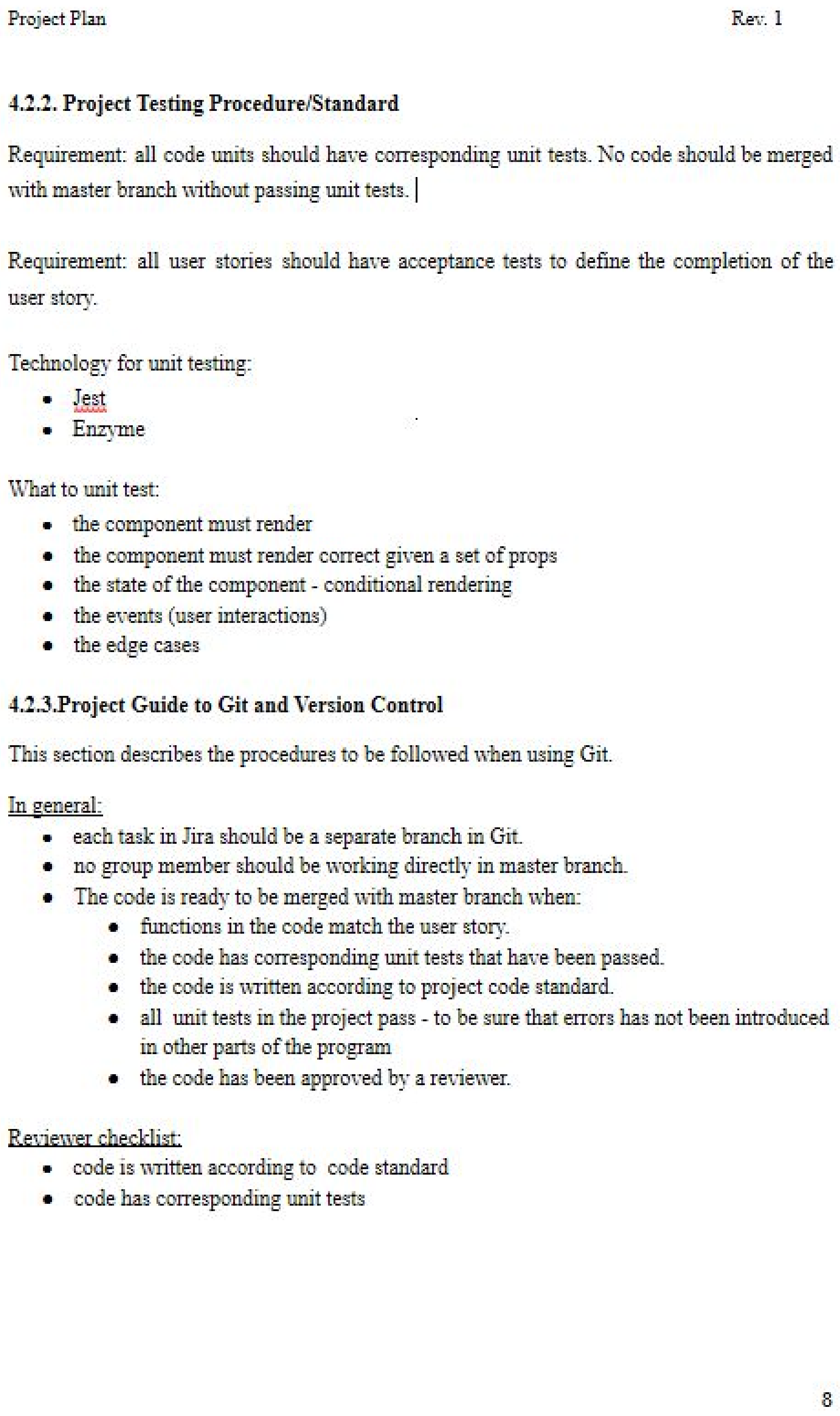
- names that describe the function of the method.
- camelCase (fx. getName)
- methodname written in Norwegian
- method cohesion - each method focuses on only one function.

Components

- Informative class name
- First letter capitalized (fx. User)
- Class name written in Norwegian
- Only letters, no signs
- A components state should be changed via functions, no changes directly to the components state
- Focus on cohesion - all elements in component is focusing on only one thing.
- Responsibility driven design - each component should have a well defined responsibility.
- Each component receiving props should run typechecking on the props.
- Components with state should be at the top of the component hierarchy. Values to be passed to child components via props.
-

Code structure:

- Front end will be coded in React
- Each component will have a separate folder. Each folder should contain:
  - component.js -file
  - component.test.js file
  - component.css file
- components with state to be placed in "Container" -file
- components without state to be placed in "Komponente" - file.



4.2.2. Project Testing Procedure/Standard

Requirement: all code units should have corresponding unit tests. No code should be merged with master branch without passing unit tests.

Requirement: all user stories should have acceptance tests to define the completion of the user story.

Technology for unit testing:

- Jest
- Enzyme

What to unit test:

- the component must render
- the component must render correct given a set of props
- the state of the component - conditional rendering
- the events (user interactions)
- the edge cases

4.2.3.Project Guide to Git and Version Control

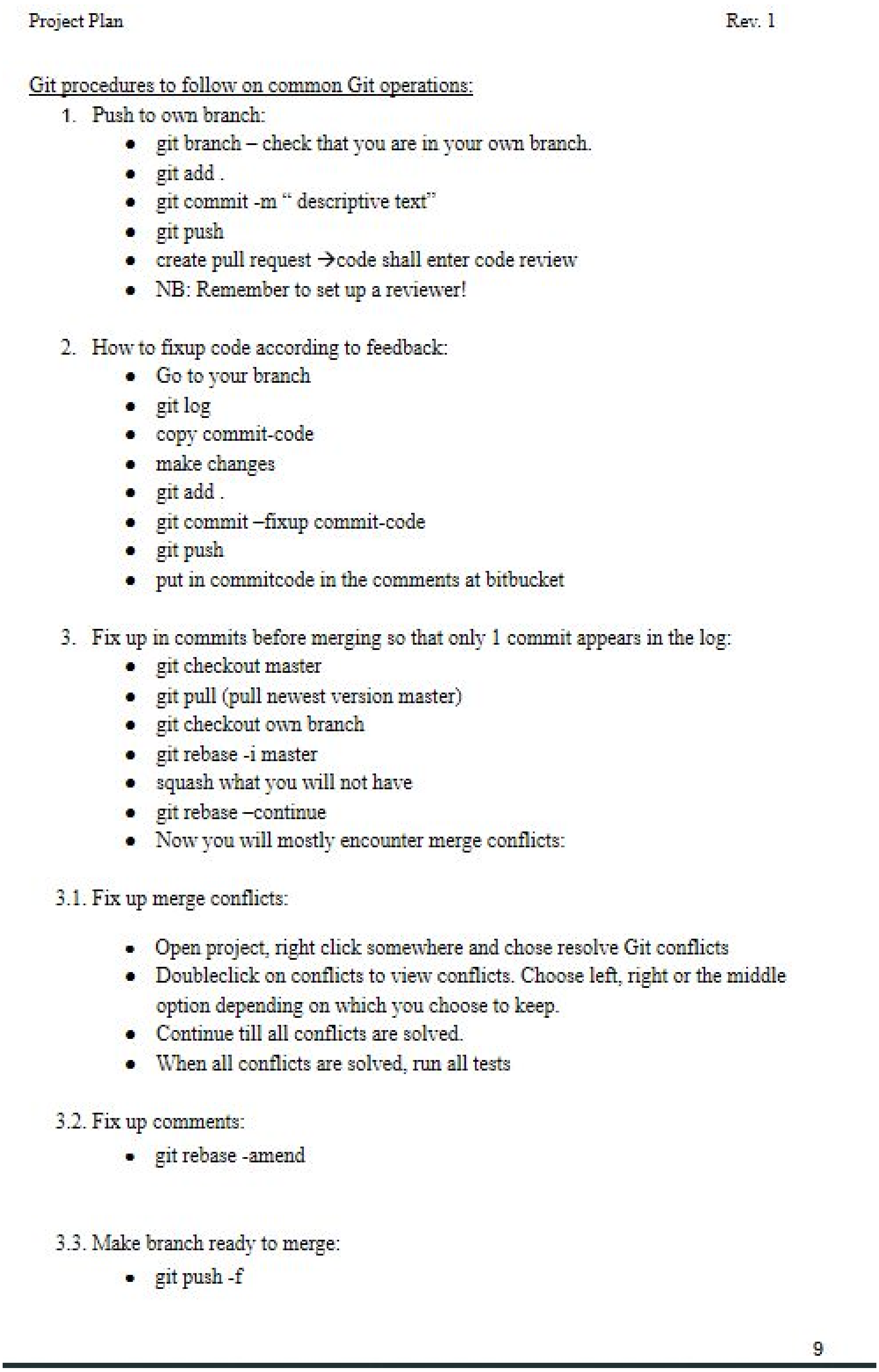
This section describes the procedures to be followed when using Git.

In general:

- each task in Jira should be a separate branch in Git.
- no group member should be working directly in master branch.
- The code is ready to be merged with master branch when:
    - functions in the code match the user story.
    - the code has corresponding unit tests that have been passed.
    - the code is written according to project code standard.
    - all unit tests in the project pass - to be sure that errors has not been introduced in other parts of the program
    - the code has been approved by a reviewer.

Reviewer checklist:

- code is written according to code standard
- code has corresponding unit tests



Git procedures to follow on common Git operations:

1. Push to own branch:
   - git branch – check that you are in your own branch.
   - git add .
   - git commit -m " descriptive text"
   - git push
   - create pull request →code shall enter code review
   - NB: Remember to set up a reviewer!

2. How to fixup code according to feedback:
   - Go to your branch
   - git log
   - copy commit-code
   - make changes
   - git add .
   - git commit –fixup commit-code
   - git push
   - put in commitcode in the comments at bitbucket

3. Fix up in commits before merging so that only 1 commit appears in the log:
   - git checkout master
   - git pull (pull newest version master)
   - git checkout own branch
   - git rebase -i master
   - squash what you will not have
   - git rebase –continue
   - Now you will mostly encounter merge conflicts:

3.1. Fix up merge conflicts:

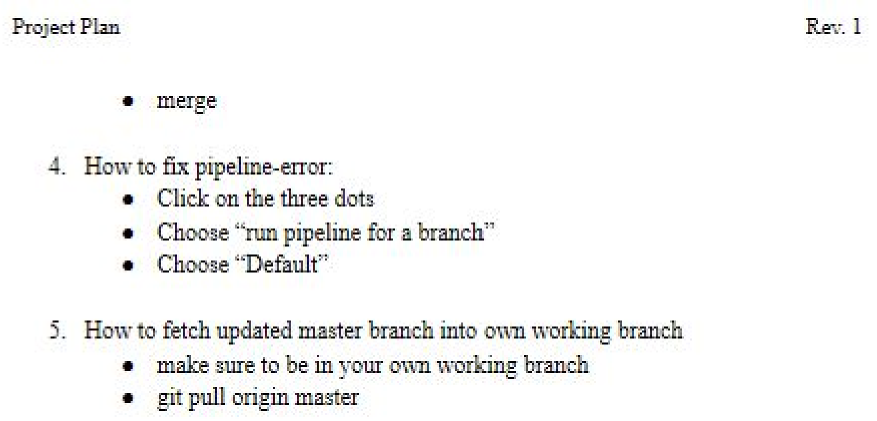
- Open project, right click somewhere and chose resolve Git conflicts
- Doubleclick on conflicts to view conflicts. Choose left, right or the middle option depending on which you choose to keep.
- Continue till all conflicts are solved.
- When all conflicts are solved, run all tests

3.2. Fix up comments:

- git rebase -amend

3.3. Make branch ready to merge:

- git push -f



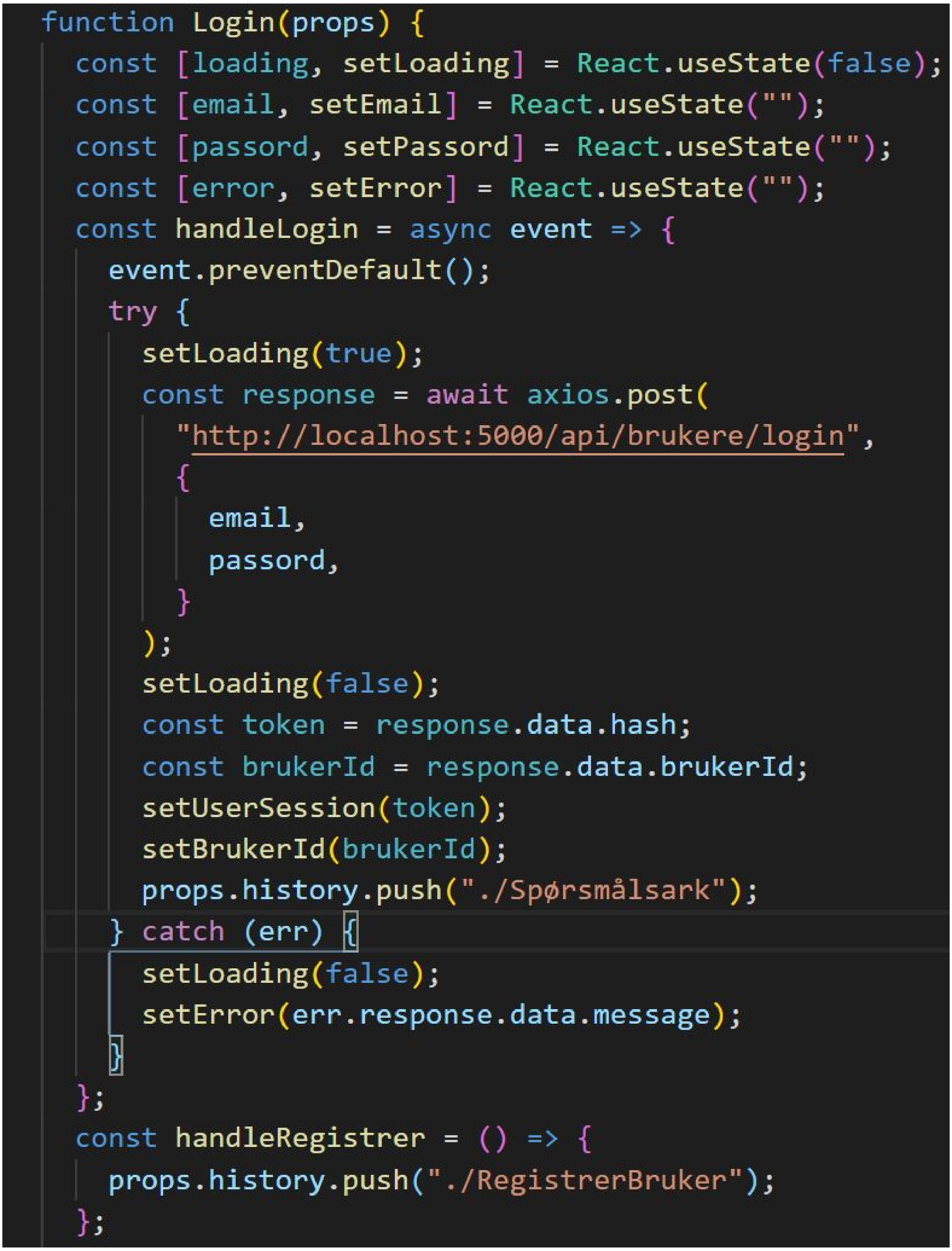
- merge

4. How to fix pipeline-error:
   - Click on the three dots
   - Choose “run pipeline for a branch”
   - Choose “Default”

5. How to fetch updated master branch into own working branch
   - make sure to be in your own working branch
   - git pull origin master

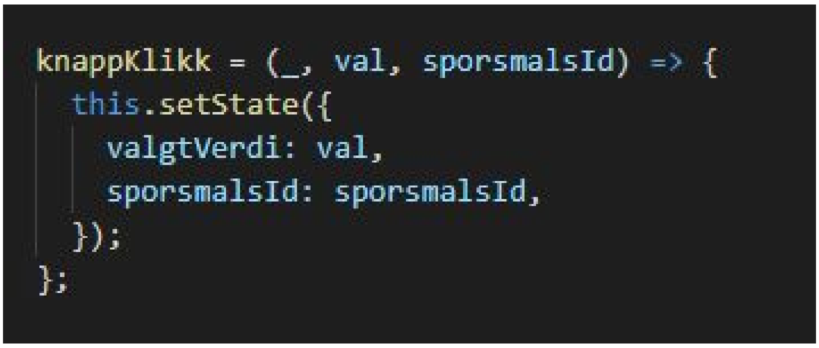
Appendix 2 - Code Examples

Example 1: login component

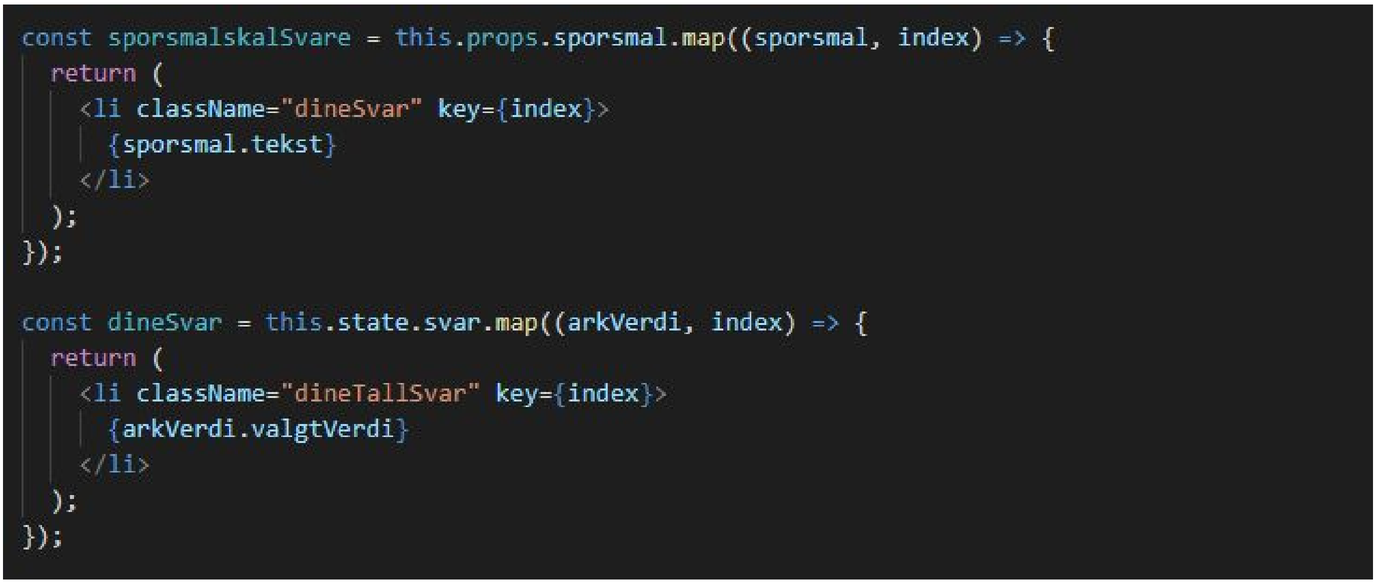


```
function Login(props) {
  const [loading, setLoading] = React.useState(false);
  const [email, setEmail] = React.useState("");
  const [passord, setPassord] = React.useState("");
  const [error, setError] = React.useState("");
  const handleLogin = async event => {
    event.preventDefault();
    try {
      setLoading(true);
      const response = await axios.post(
        "http://localhost:5000/api/brukere/login",
        {
          email,
          passord,
        }
      );
      setLoading(false);
      const token = response.data.hash;
      const brukerId = response.data.brukerId;
      setUserSession(token);
      setBrukerId(brukerId);
      props.history.push("./Spørsmålsark");
    } catch (err) {
      setLoading(false);
      setError(err.response.data.message);
    }
  };
  const handleRegistrer = () => {
    props.history.push("./RegistrerBruker");
  };
```

Example 2: variable and methods named according to standard



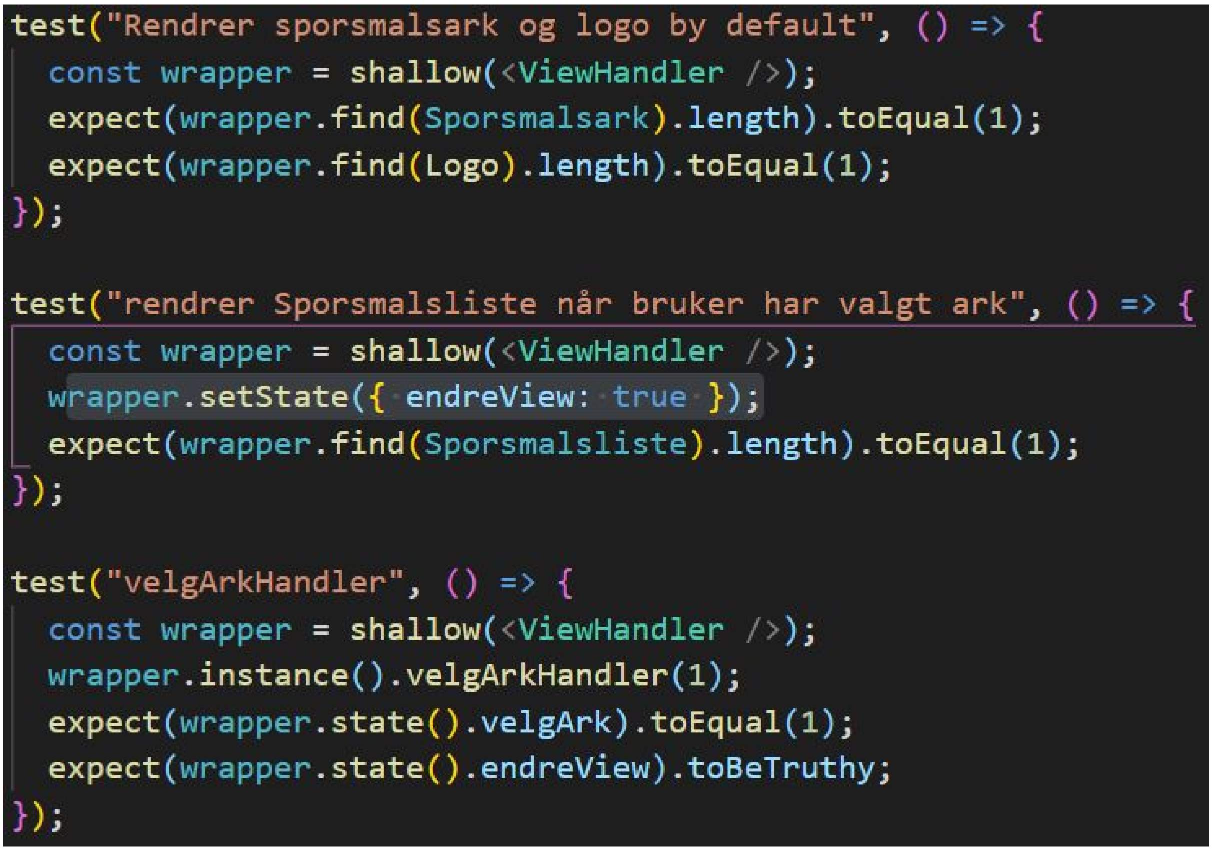
```
knappKlikk = (_, val, sporsmalsId) => {
  this.setState({
    valgtVerdi: val,
    sporsmalsId: sporsmalsId,
  });
};
```



```
const sporsmalskalSvare = this.props.sporsmal.map((sporsmal, index) => {
  return (
    <li className="dineSvar" key={index}>
      {sporsmal.tekst}
    </li>
  );
});

const dineSvar = this.state.svar.map((arkVerdi, index) => {
  return (
    <li className="dineTallSvar" key={index}>
      {arkVerdi.valgtVerdi}
    </li>
  );
});
```

Appendix 3 - Example of Unit Test



```
test("Rendrer sporsmalsark og logo by default", () => {
  const wrapper = shallow(<ViewHandler />);
  expect(wrapper.find(Sporsmalsark).length).toEqual(1);
  expect(wrapper.find(Logo).length).toEqual(1);
});

test("rendrer Sporsmalsliste når bruker har valgt ark", () => {
  const wrapper = shallow(<ViewHandler />);
  wrapper.setState({ endreView: true });
  expect(wrapper.find(Sporsmalsliste).length).toEqual(1);
});

test("velgArkHandler", () => {
  const wrapper = shallow(<ViewHandler />);
  wrapper.instance().velgArkHandler(1);
  expect(wrapper.state().velgArk).toEqual(1);
  expect(wrapper.state().endreView).toBeTruthy;
});
```

Appendix 4 - Project Scrum Board

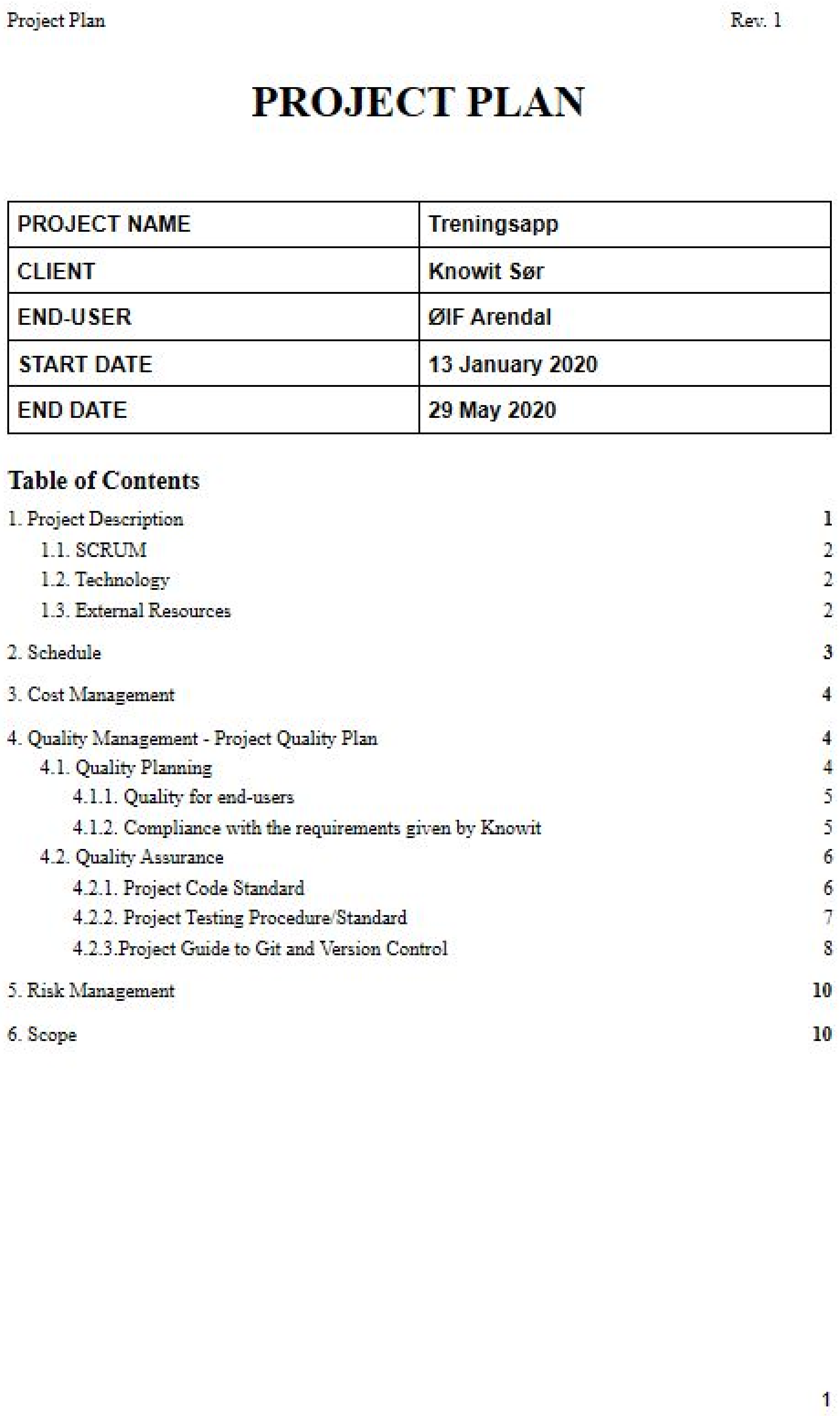
Scrum board with the columns: To-do, In-progress, In-Peer-Review, Quality Assurance (QA) and Done

Appendix 5 - Examples of User Story

User story 1	As a user, I want a login page so I can log in
Priority	Must-have
Arguments	Without login page, user will not be able to log in and use the system
Acceptance Criteria	field for user name (email -address) field for password log-in button unit test
Acceptance test	User fills in all login details and is allowed to log in

User story 3	To be able to find my unanswered question-schemas, as a user I want to find the schemas at the front page
Priority	Must-have
Arguments	The question schemas are necessary to get access to the questions.
Acceptance Criteria	unanswered question schemas displayed as list on front page each schema function as a link to the question site code to have corresponding unit tests
Acceptance test	user enters front page of app and finds his unanswered question schemas when clicking on the schema-icon, user accesses the questions.

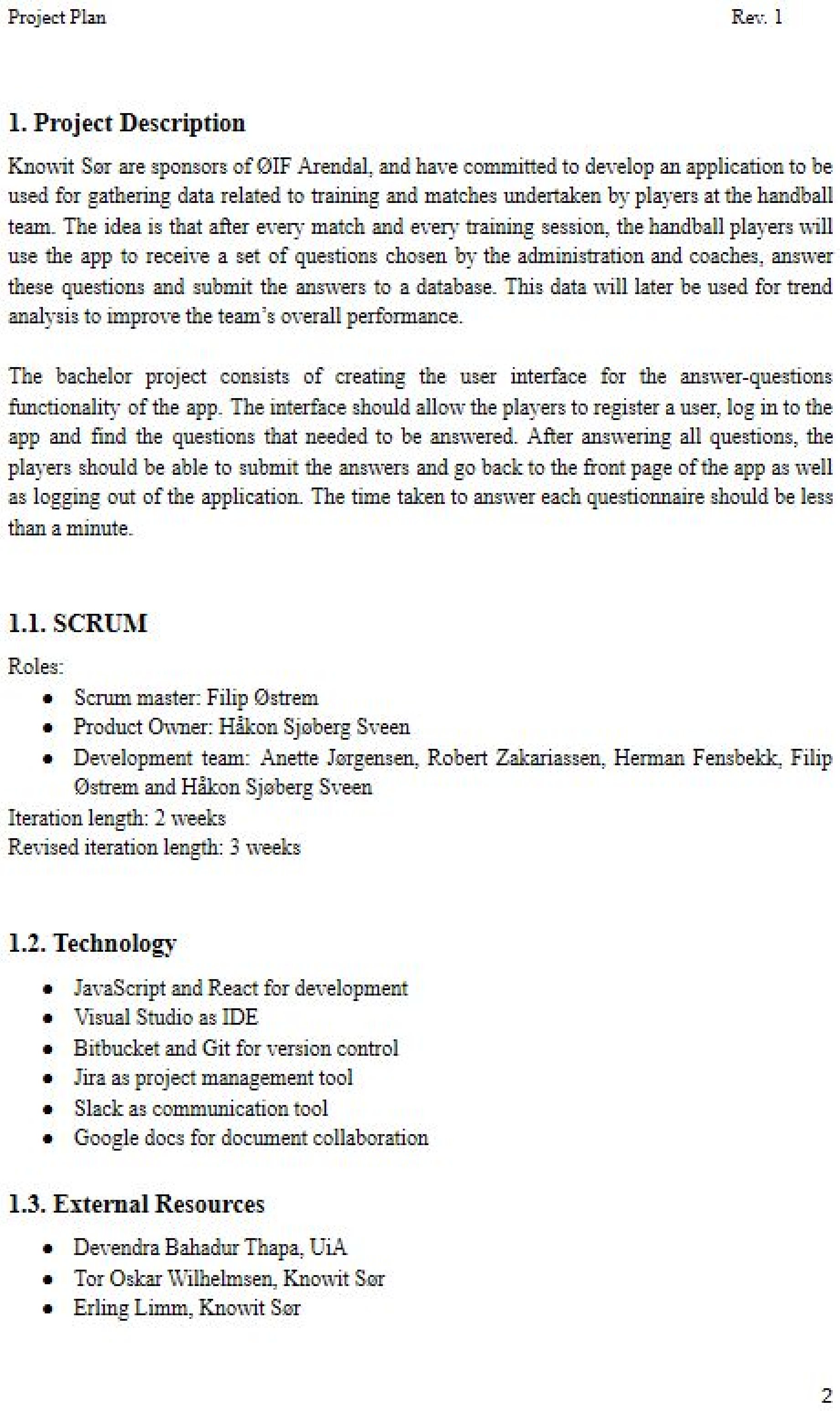
Appendix 6 - Extract of Project Plan



PROJECT PLAN

PROJECT NAME	Treningsapp
CLIENT	Knowit Sør
END-USER	ØIF Arendal
START DATE	13 January 2020
END DATE	29 May 2020

Table of Contents



1. Project Description

Knowit Sør are sponsors of ØIF Arendal, and have committed to develop an application to be used for gathering data related to training and matches undertaken by players at the handball team. The idea is that after every match and every training session, the handball players will use the app to receive a set of questions chosen by the administration and coaches, answer these questions and submit the answers to a database. This data will later be used for trend analysis to improve the team's overall performance.

The bachelor project consists of creating the user interface for the answer-questions functionality of the app. The interface should allow the players to register a user, log in to the app and find the questions that needed to be answered. After answering all questions, the players should be able to submit the answers and go back to the front page of the app as well as logging out of the application. The time taken to answer each questionnaire should be less than a minute.

1.1. SCRUM

Roles:

- Scrum master: Filip Østrem
- Product Owner: Håkon Sjøberg Sveen
- Development team: Anette Jørgensen, Robert Zakariassen, Herman Fensbekk, Filip Østrem and Håkon Sjøberg Sveen

Iteration length: 2 weeks
Revised iteration length: 3 weeks

1.2. Technology

- JavaScript and React for development
- Visual Studio as IDE
- Bitbucket and Git for version control
- Jira as project management tool
- Slack as communication tool
- Google docs for document collaboration

1.3. External Resources

- Devendra Bahadur Thapa, UiA
- Tor Oskar Wilhelmsen, Knowit Sør
- Erling Limm, Knowit Sør

2. Schedule

Project start date: 13 January 2020
Project end date: 29 May 2020

Project phases:

- Pre-sprint/project planning
- Design phase
- Implementation
- Project wrap-up/closure

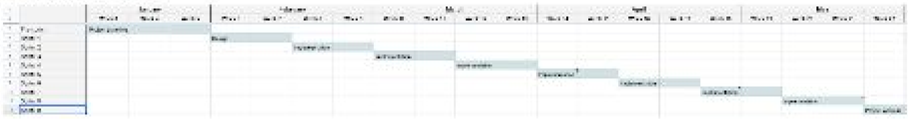
Note: Since this project is part of a bigger project,the analysis phase is already completed by Knowit

Milestones:

- week 7: steering committee meeting
- week 14: steering committee meeting
- week 18: steering committee meeting
- week 21: project report hand - in

Timeline version 1
The timeline consists of 9 sprints, with a duration of 2 weeks on each sprint

Timeline Rev. 1:



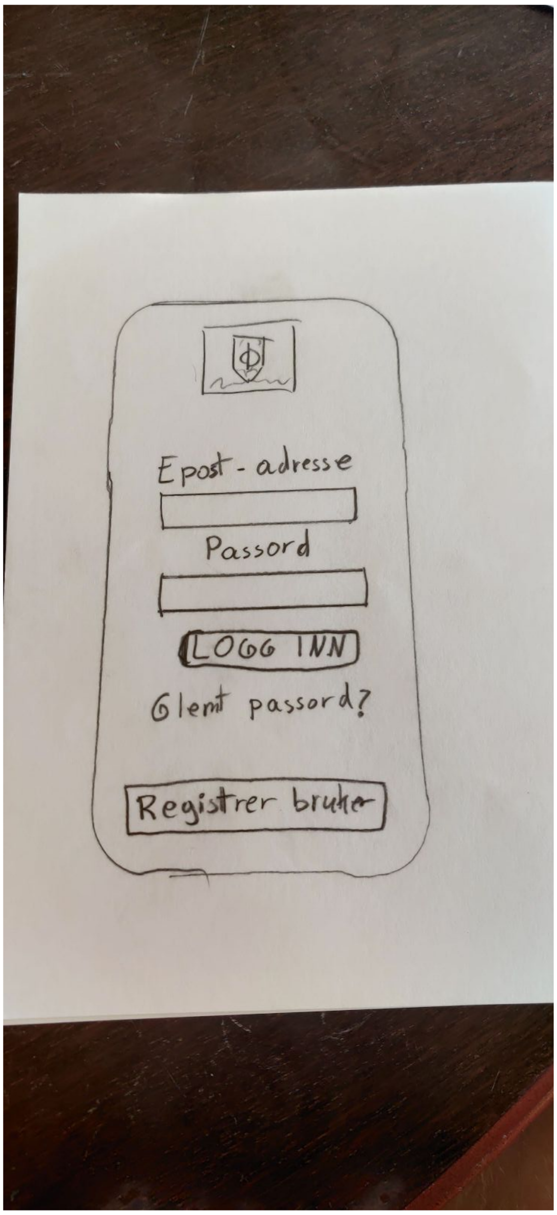
Timeline version 2
The duration of sprints were changed from 2 weeks to 3 weeks. This change was based upon feedback given on Steering Committee Meeting no. 1t

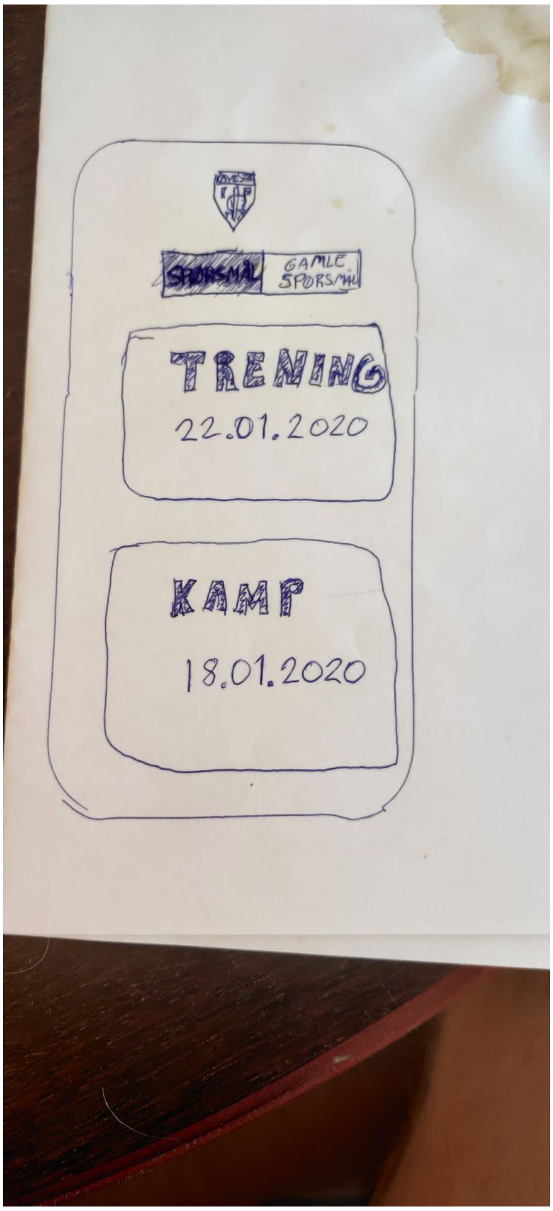
Revised timeline consists of 6 sprints, with a duration of 3 weeks on each sprint.

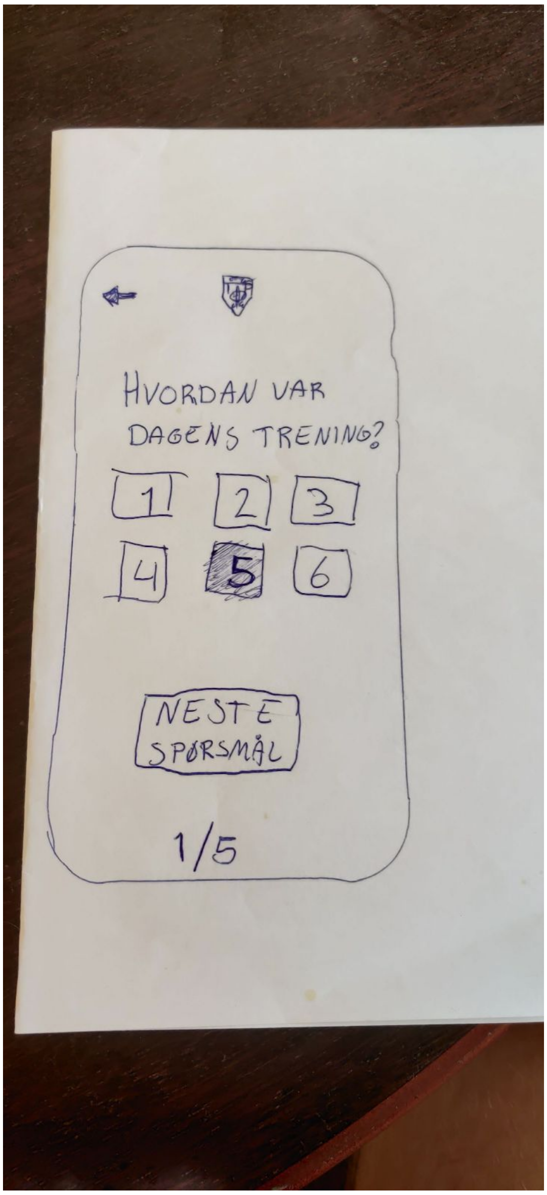
Timeline Rev 2:

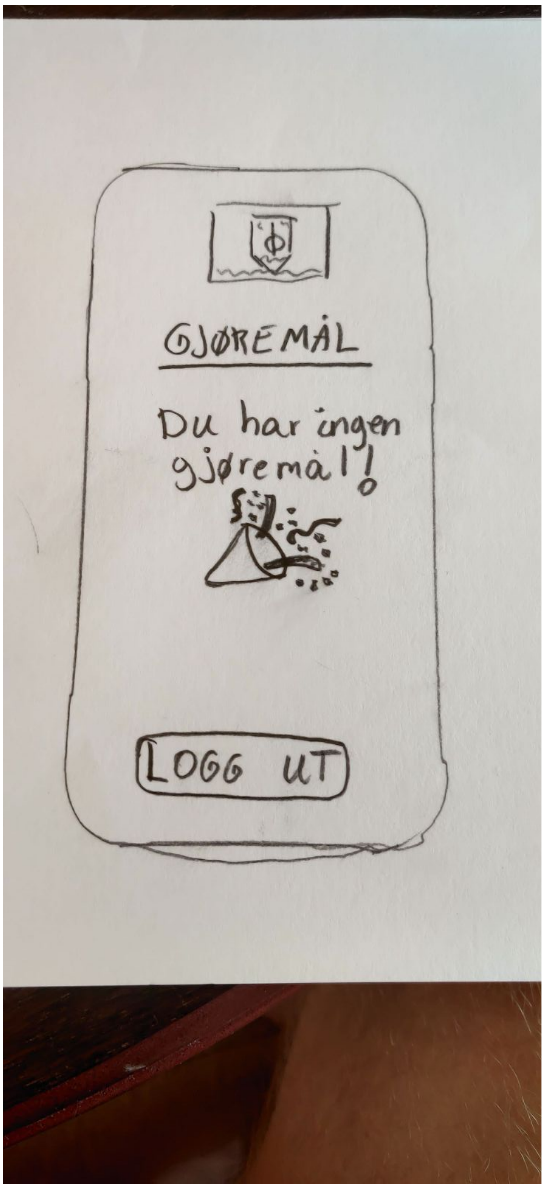
January
February
March
April
May

Appendix 7 - Sketches



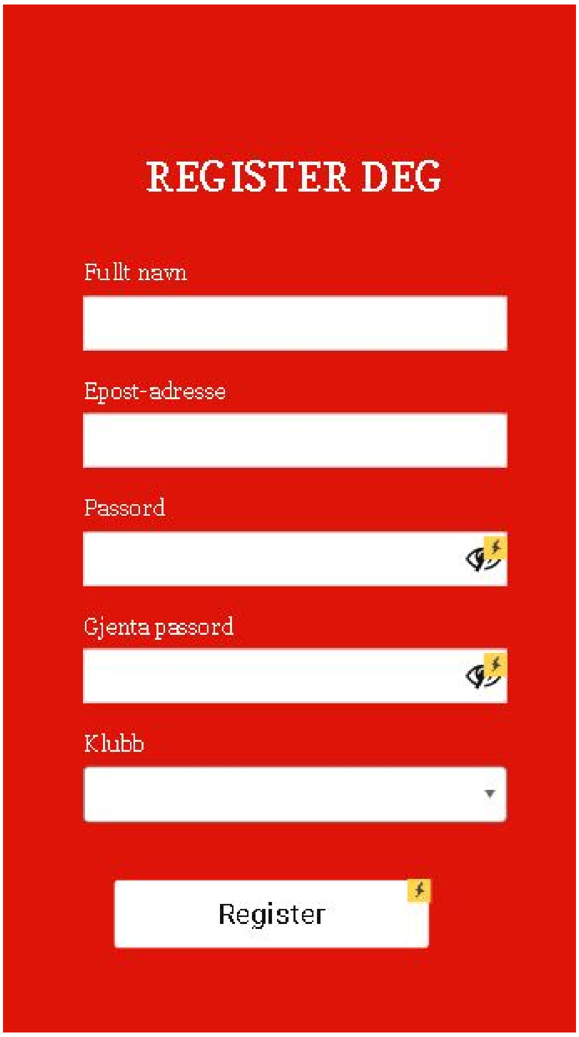


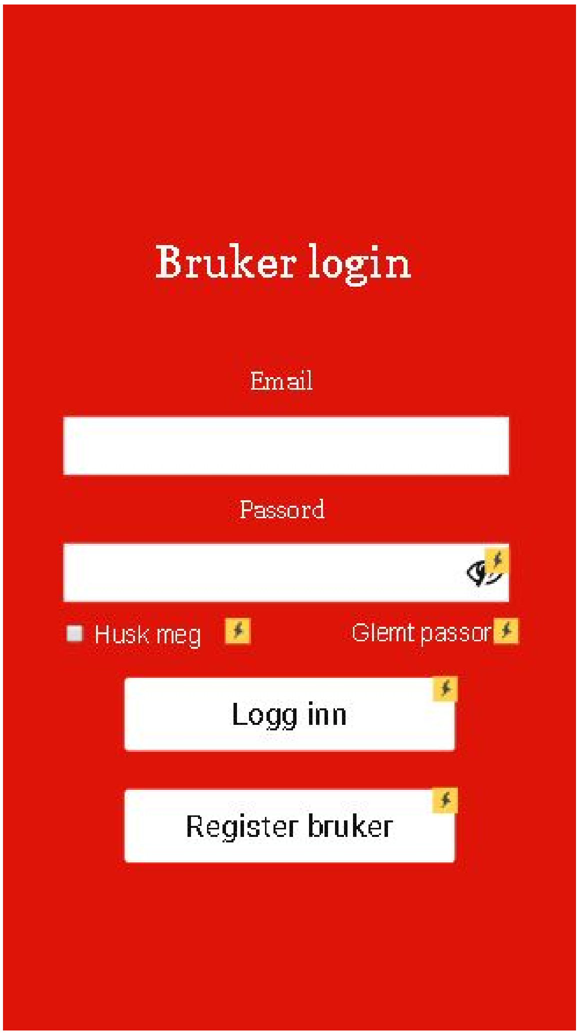

HVORDAN VAR DAGENS TRENING?
1
2
3
4
5
6
NESTE SPØRSMÅL
1/5

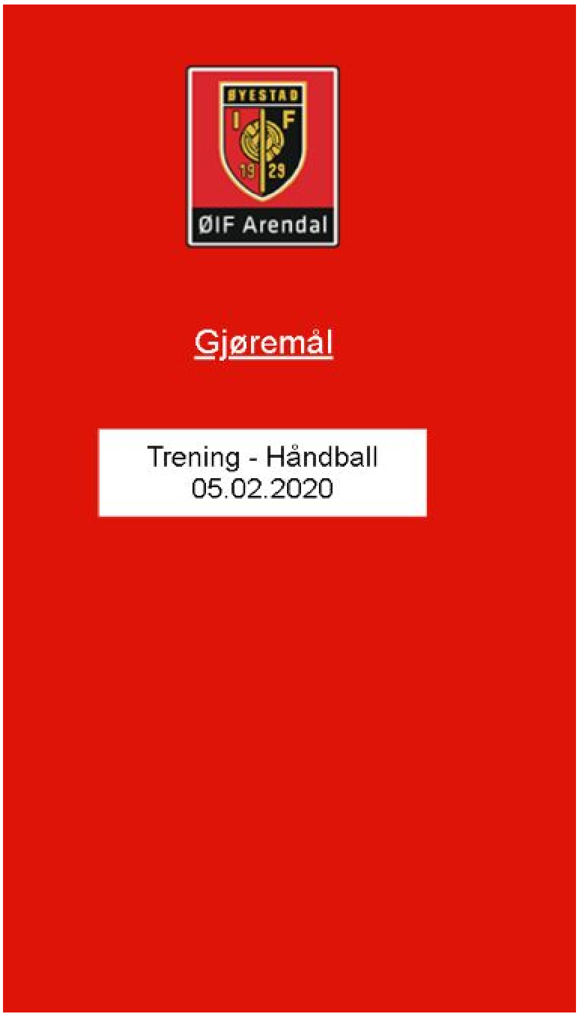

GJØREMÅL
Du har ingen gjøremål!
LOGG UT

Appendix 8 - UX prototype

Prototype available at:
https://preview.uxpin.com/7fb204007fc97420378cd5c8e987a91e9a28d1bf#/pages/122748838/simulate/sitemap?mode=i

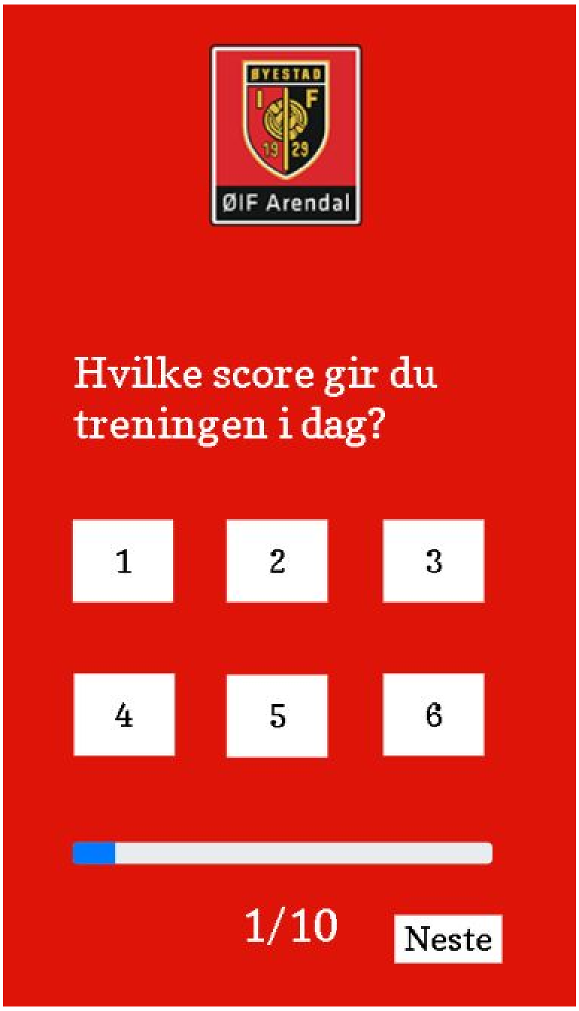






Gjøremål

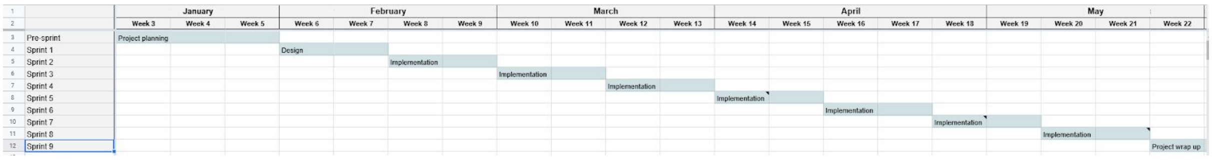
Trening - Håndball
05.02.2020



Hvilke score gir du treningen i dag?

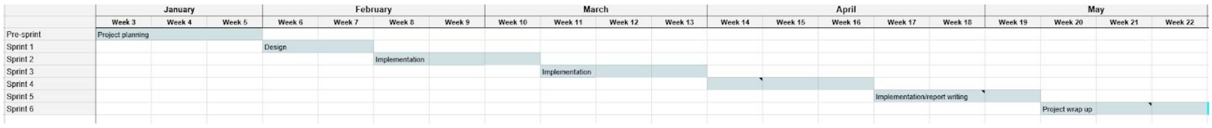
Appendix 9 - Timeline

Timeline rev. 1



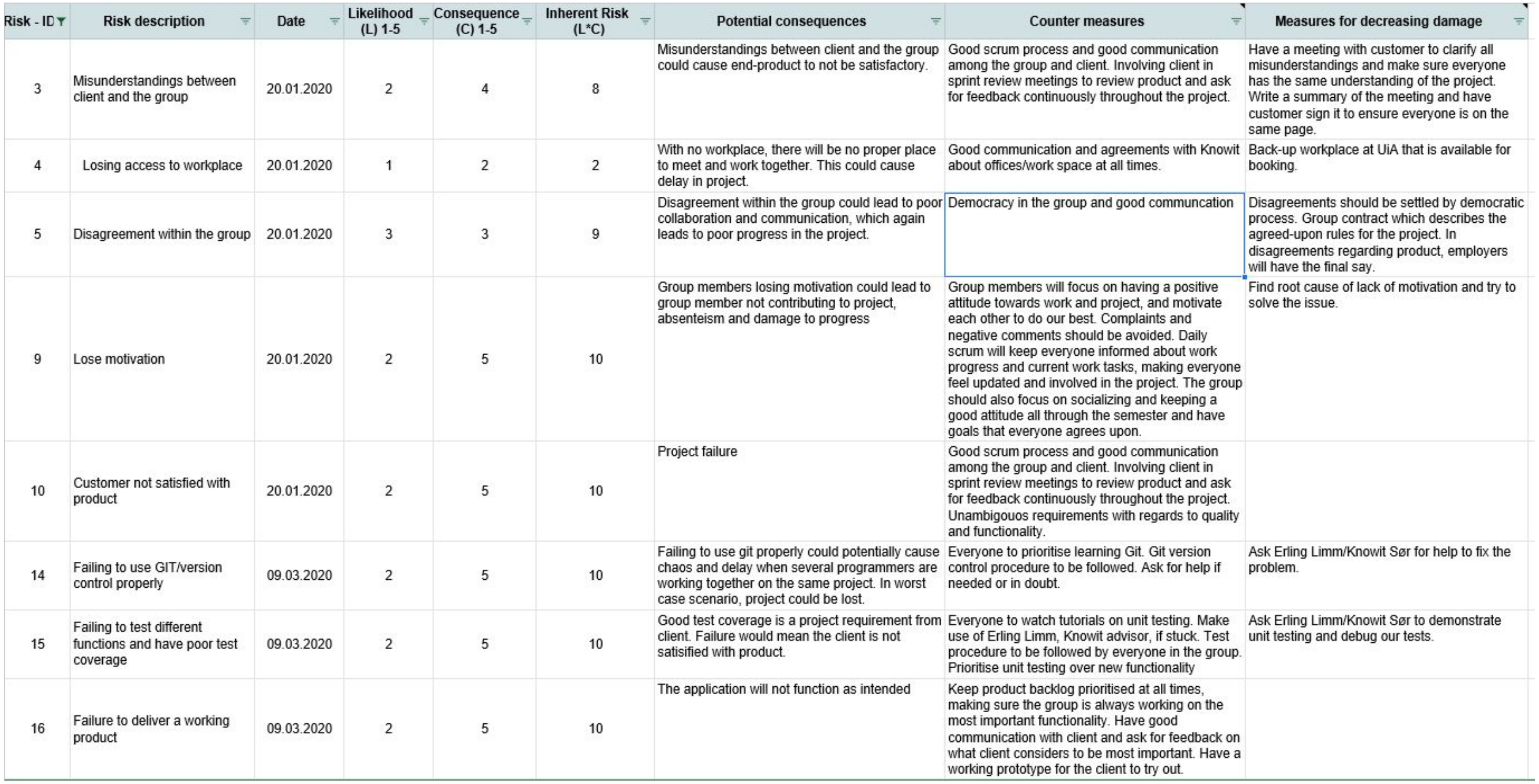
1		January			February				March				April					May			
2		Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22
3	Pre-sprint	Project planning	▨	▨																	
4	Sprint 1				Design	▨															
5	Sprint 2						Implementation	▨													
6	Sprint 3								Implementation	▨											
7	Sprint 4										Implementation	▨									
8	Sprint 5												Implementation	▨							
9	Sprint 6														Implementation	▨					
10	Sprint 7																Implementation	▨			
11	Sprint 8																		Implementation	▨	
12	Sprint 9																				Project wrap up

Timeline rev. 2



	January			February				March				April					May			
	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22
Pre-sprint	Project planning	▨	▨																	
Sprint 1				Design	▨															
Sprint 2						Implementation	▨	▨												
Sprint 3									Implementation	▨	▨									
Sprint 4												▨	▨	▨						
Sprint 5															Implementation/report writing	▨	▨			
Sprint 6																		Project wrap up	▨	▨

Appendix 10 - Risk register



Risk - ID	Risk description	Date	Likelihood (L) 1-5	Consequence (C) 1-5	Inherent Risk (L*C)	Potential consequences	Counter measures	Measures for decreasing damage
3	Misunderstandings between client and the group	20.01.2020	2	4	8	Misunderstandings between client and the group could cause end-product to not be satisfactory.	Good scrum process and good communication among the group and client. Involving client in sprint review meetings to review product and ask for feedback continuously throughout the project.	Have a meeting with customer to clarify all misunderstandings and make sure everyone has the same understanding of the project. Write a summary of the meeting and have customer sign it to ensure everyone is on the same page.
4	Losing access to workplace	20.01.2020	1	2	2	With no workplace, there will be no proper place to meet and work together. This could cause delay in project.	Good communication and agreements with Knowit about offices/work space at all times.	Back-up workplace at UiA that is available for booking.
5	Disagreement within the group	20.01.2020	3	3	9	Disagreement within the group could lead to poor collaboration and communication, which again leads to poor progress in the project.	Democracy in the group and good communcation	Disagreements should be settled by democratic process. Group contract which describes the agreed-upon rules for the project. In disagreements regarding product, employers will have the final say.
9	Lose motivation	20.01.2020	2	5	10	Group members losing motivation could lead to group member not contributing to project, absenteism and damage to progress	Group members will focus on having a positive attitude towards work and project, and motivate each other to do our best. Complaints and negative comments should be avoided. Daily scrum will keep everyone informed about work progress and current work tasks, making everyone feel updated and involved in the project. The group should also focus on socializing and keeping a good attitude all through the semester and have goals that everyone agrees upon.	Find root cause of lack of motivation and try to solve the issue.
10	Customer not satisfied with product	20.01.2020	2	5	10	Project failure	Good scrum process and good communication among the group and client. Involving client in sprint review meetings to review product and ask for feedback continuously throughout the project. Unambigouos requirements with regards to quality and functionality.	
14	Failing to use GIT/version control properly	09.03.2020	2	5	10	Failing to use git properly could potentially cause chaos and delay when several programmers are working together on the same project. In worst case scenario, project could be lost.	Everyone to prioritise learning Git. Git version control procedure to be followed. Ask for help if needed or in doubt.	Ask Erling Limm/Knowit Sør for help to fix the problem.
15	Failing to test different functions and have poor test coverage	09.03.2020	2	5	10	Good test coverage is a project requirement from client. Failure would mean the client is not satisified with product.	Everyone to watch tutorials on unit testing. Make use of Erling Limm, Knowit advisor, if stuck. Test procedure to be followed by everyone in the group. Prioritise unit testing over new functionality	Ask Erling Limm/Knowit Sør to demonstrate unit testing and debug our tests.
16	Failure to deliver a working product	09.03.2020	2	5	10	The application will not function as intended	Keep product backlog prioritised at all times, making sure the group is always working on the most important functionality. Have good communication with client and ask for feedback on what client considers to be most important. Have a working prototype for the client to try out.	

Appendix 11 - Risk diagram

Risk diagram 20.01.2020 - at the start of the project

Likelihood	Very High	🟨	🟨	🟥	🟥	🟥
	High	🟨 1	🟨	🟨	🟥	🟥
	Possible	🟩	🟨 7	🟨 5	🟨	🟥
	Low	🟩 12	🟩 2	🟨	🟨 4,11,13,17	🟨 9,14,15,16,10
	Very Low	🟩	🟩 3	🟩	🟩 8	🟨 6
		Very Low	Low	Significant	High	Catastrophic
Consequence						

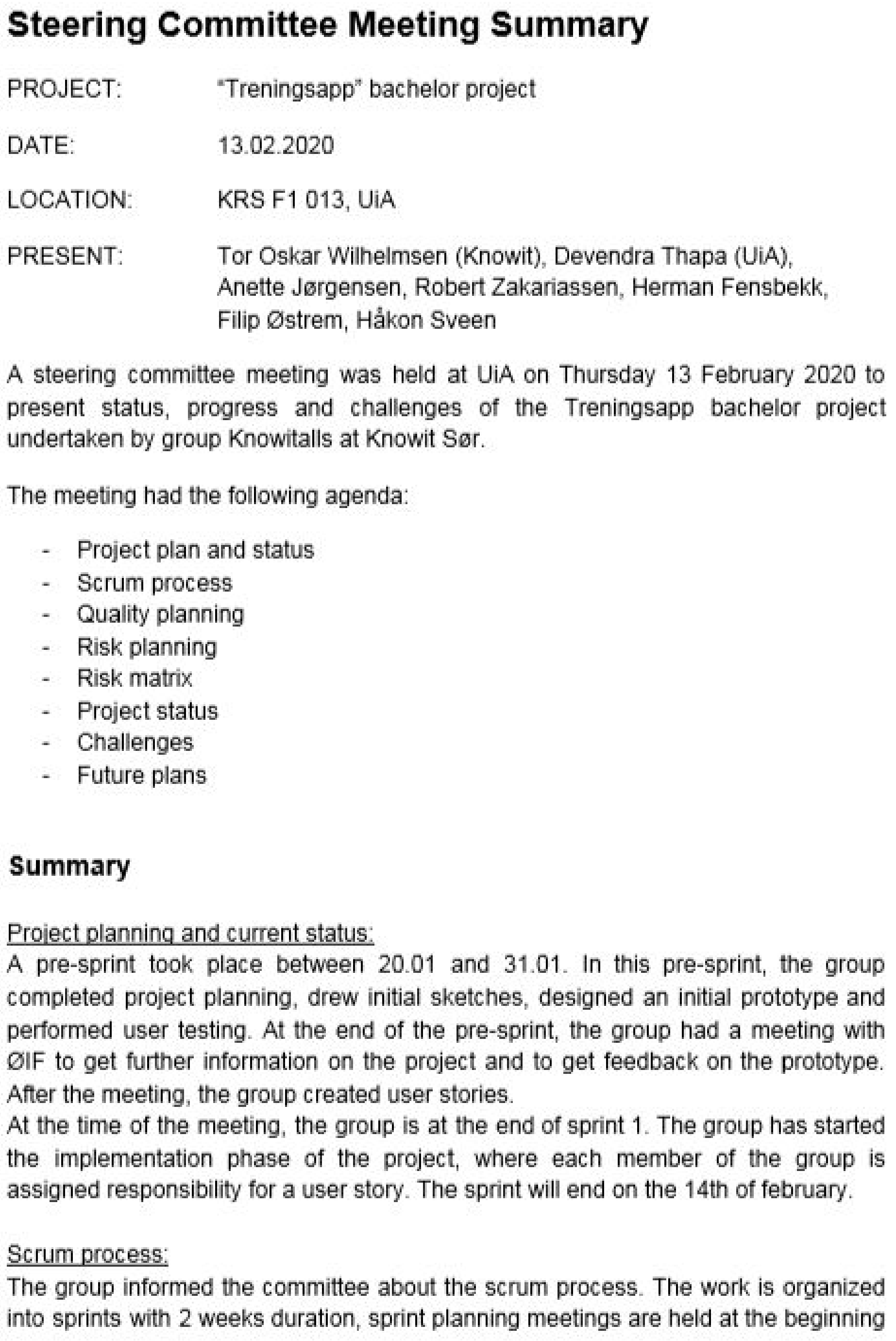
Risk diagram 16.03.2020 - crisis meeting due to coronavirus shutdown

Likelihood	Very High	🟨 14	🟨	🟥	🟥	🟥
	High	🟨 1	🟨	🟨	🟥	🟥 15
	Possible	🟩	🟨 7	🟨 5	🟨	🟥 16
	Low	🟩	🟩 2	🟨	🟨 4	🟨 9,10,11,12,13
	Very Low	🟩	🟩 3	🟩	🟩 8	🟨
		Very Low	Low	Significant	High	Catastrophic
Consequence						

Risk diagram 20.04.2020 - start of sprint 5

Likelihood	Very High	🟨 4	🟨	🟥	🟥	🟥
	High	🟨 1	🟨	🟨	🟥	🟥 15
	Possible	🟩	🟨 7	🟨	🟨	🟥 16
	Low	🟩	🟩 2,5,10	🟨 11	🟨	🟨
	Very Low	🟩 9,13	🟩 3	🟩	🟩 8,12	🟨 6,14
		Very Low	Low	Significant	High	Catastrophic
Consequence						

Appendix 12 - Steering Committee Meeting 13.02.20



Steering Committee Meeting Summary

PROJECT:	"Treningsapp" bachelor project
DATE:	13.02.2020
LOCATION:	KRS F1 013, UiA
PRESENT:	Tor Oskar Wilhelmsen (Knowit), Devendra Thapa (UiA), Anette Jørgensen, Robert Zakariassen, Herman Fensbekk, Filip Østrem, Håkon Sveen

A steering committee meeting was held at UiA on Thursday 13 February 2020 to present status, progress and challenges of the Treningsapp bachelor project undertaken by group Knowitalls at Knowit Sør.

The meeting had the following agenda:

- Project plan and status
- Scrum process
- Quality planning
- Risk planning
- Risk matrix
- Project status
- Challenges
- Future plans

Summary

Project planning and current status:
A pre-sprint took place between 20.01 and 31.01. In this pre-sprint, the group completed project planning, drew initial sketches, designed an initial prototype and performed user testing. At the end of the pre-sprint, the group had a meeting with ØIF to get further information on the project and to get feedback on the prototype. After the meeting, the group created user stories.
At the time of the meeting, the group is at the end of sprint 1. The group has started the implementation phase of the project, where each member of the group is assigned responsibility for a user story. The sprint will end on the 14th of february.

Scrum process:
The group informed the committee about the scrum process. The work is organized into sprints with 2 weeks duration, sprint planning meetings are held at the beginning

of every sprint and sprint retrospective and sprint review meetings are held at the end of every sprint. Filip Østrem has the role as scrum master.

Quality planning:
To ensure quality in the process and in the final product, the group has created a project quality plan. This plan includes definitions for what is quality in the project, a code standard to ensure everyone is writing code according to agreed upon principles, a test procedure to ensure good test coverage and a guide to git to ensure everyone performs version control in a correct way. At the time of the meeting, the test procedure was not yet completed due to lack of knowledge.

Risk planning:
The group created a risk matrix with a lot of different types of risk that can happen during the project. As a group we created risk and rated the consequence and likelihood of these risks from 1-5 and then we got the risk assessment. The counter measures were then created to prevent these risks for happening and measures for decreasing damage. Everyone in the group participated in creating these risks and rated them. All risk where then put into a risk matrix from very low to very high in likelihood and very low to catastrophic in consequence with the color green, yellow and red from the outcome.

Challenges:
The group stumbled upon a few challenges during the pre-sprint and sprint 1 which were also informed to the committee. There were three main challenges that the group faced. These are unfamiliarity with front-end (React and Javascript), lack of experience which causes estimation problems and different experience between the different group members. Solutions to these challenges were that each team member had to take responsibility to learn React and Javascript at work, but it was also expected to put some time into learning in their spare time. With experience, the time estimation is more likely to be more precise. A good scrum process and helping each other when help is needed.

Future plans:
The group plans to continue implementing the user stories.

Discussion/desired feedback:

The group asked for feedback on the following points:

- the scrum process - any room for improvements? Is the duration of sprints sensible?
- the Jira setup and user stories - any room for improvements?

- Is the group at a good place altogether?

Feedback received

The Scrum process:

The group was advised to prolong the duration of the sprints from 2 weeks to 3 weeks. This is because with only two weeks, it is more difficult to produce anything of value to customer. The goal of every sprint should be to produce something of value.

The group was advised to use planning poker in the time estimation process. There are many apps one can use for this purpose. It is important to make sure the group members are not influenced by each other while estimating.

There is room for improvement when it comes to time estimation of each work task. In addition to log hours spent working on a task, it was important to estimate remaining hours as well.

The group should add a QA step to the workflow process. So far, the workflow has consisted of "To do", "In progress" "In peer-review" and "Done". Before a task is done, it should be deployed and tested manually with real data to make sure there are no errors. The group is advised to create a test script for this purpose.

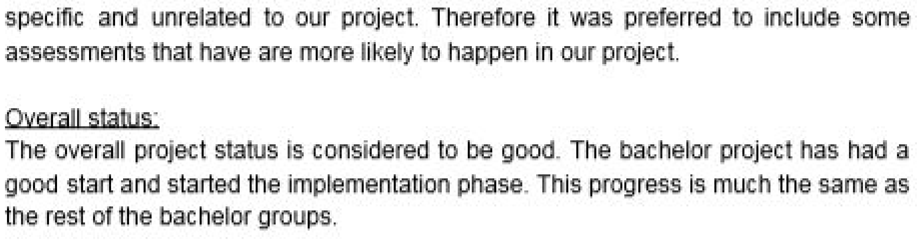
Jira setup/user stories:

The group received feedback on how to improve the user stories. Some of the user stories lacked acceptance criteria and acceptance tests, and this needed to be added to the stories.

In addition, no administration tasks should be part of Jira. One task in Jira should equal one branch in git, and should include criteria for knowing when the task is finished. An admin task called "Learned React basics" would not equal a branch in git and would not have criteria for when the task is done and should therefore not be included in Jira. The group is advised to follow the work processes and routines of the company (Knowit) rather than what we learn at UiA.

Risk analysis

The group received some mixed feedback on the risk matrix. Normally a project has a list of six risk assessments, while we had 18. The committee suggested that we reduced the number of risk assessments. Some of the assessments were also very

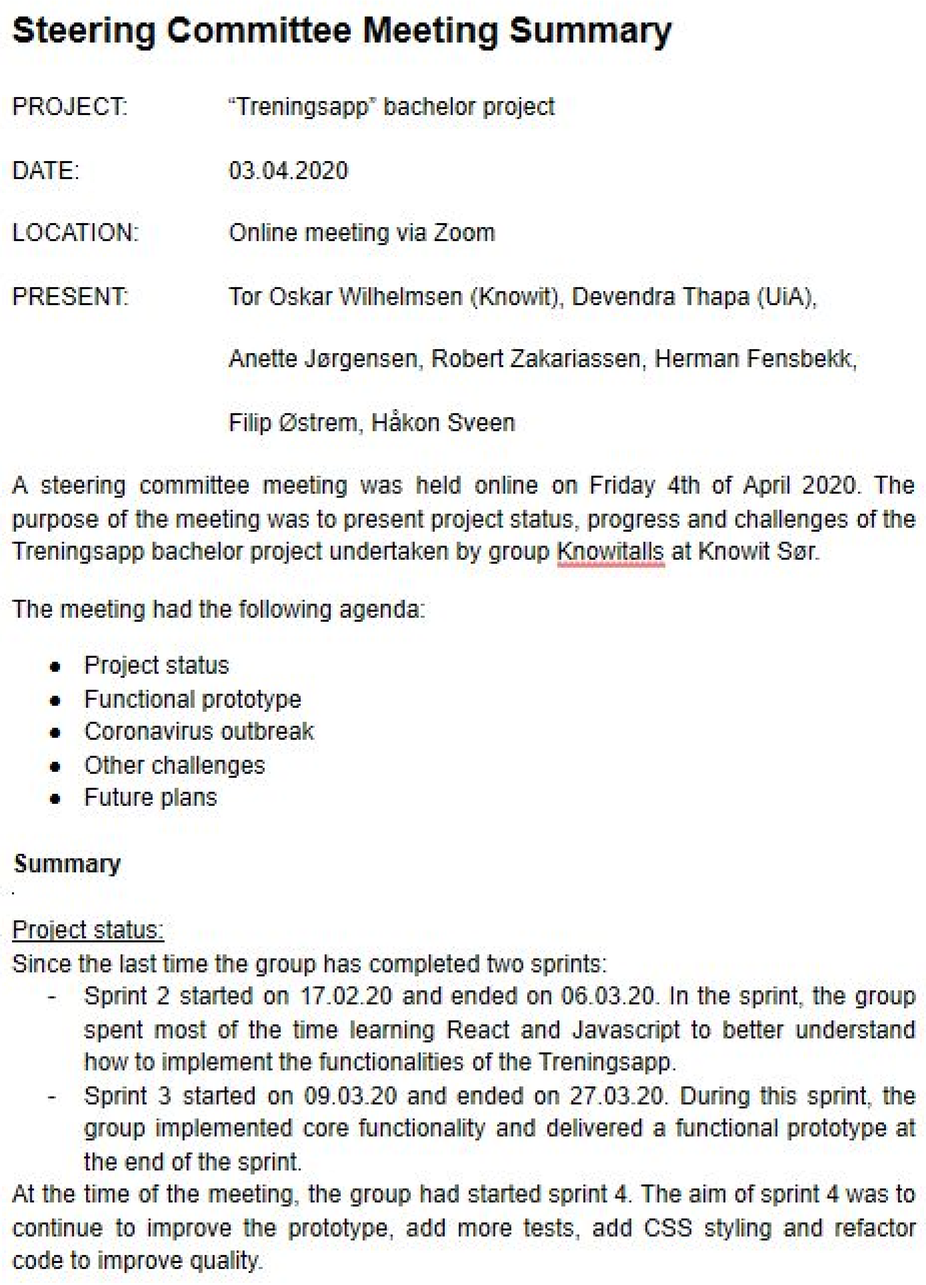


specific and unrelated to our project. Therefore it was preferred to include some assessments that have are more likely to happen in our project.

Overall status:

The overall project status is considered to be good. The bachelor project has had a good start and started the implementation phase. This progress is much the same as the rest of the bachelor groups.

Appendix 13 - Steering Committee Meeting 03.04.20



Steering Committee Meeting Summary

PROJECT:	“Treningsapp” bachelor project
DATE:	03.04.2020
LOCATION:	Online meeting via Zoom
PRESENT:	Tor Oskar Wilhelmsen (Knowit), Devendra Thapa (UiA), Anette Jørgensen, Robert Zakariassen, Herman Fensbekk, Filip Østrem, Håkon Sveen

A steering committee meeting was held online on Friday 4th of April 2020. The purpose of the meeting was to present project status, progress and challenges of the Treningsapp bachelor project undertaken by group Knowitalls at Knowit Sør.

The meeting had the following agenda:

- Project status
- Functional prototype
- Coronavirus outbreak
- Other challenges
- Future plans

Summary

.

Project status:
Since the last time the group has completed two sprints:

- Sprint 2 started on 17.02.20 and ended on 06.03.20. In the sprint, the group spent most of the time learning React and Javascript to better understand how to implement the functionalities of the Treningsapp.
- Sprint 3 started on 09.03.20 and ended on 27.03.20. During this sprint, the group implemented core functionality and delivered a functional prototype at the end of the sprint.

At the time of the meeting, the group had started sprint 4. The aim of sprint 4 was to continue to improve the prototype, add more tests, add CSS styling and refactor code to improve quality.

Functional Prototype:

We have created the core functionality with the application. Now we have a functional login page with username and password that redirect the user to the dashboard page with the questionnaires. We have added some css, there is also a navigation bar at the top of the screen, and we also have a register page, where the users can register an new user for the application, an stores the users in a fake backend server that we also have created.
In the questionnaire page we list the different questionnaires that are stored in the backend server where the user can click on any of the questionnaire and then the questions will display one by one with a scale of 1-6. At the end of the questionnaire there is a finish button that will refresh the questionnaire component.
There is also a logout page where the user can log out and their session will be done.

Coronavirus - outbreak:

During sprint 2 we had face-to-face meeting on a daily basis at the office spaces at KnowIt. Everyone on the group were comfortable with having meetings there, and being at the office in general and see nice people. We hoped we could continue to do that.

Unfortunately at sprint 3 the government announced that everyone should stay inside due to the corona virus outbreak and that was when we realised that we had to change our meetings routines. Three from the group decided to travel home as it was not advised to go outdoors and not to meet at the office space at Knowit.

Everyone in the group now had their own home office and meetings face-to-face was no longer possible. Instead meetings and discussions were carried online via Discord and Slack. It worked fine, but it came with some challenges.

First challenge was that we lost face-to-face communication. It was very strange to not meet each other more as we used to. It feels a bit strange to lose the social interaction with each other. Second challenge was that we lost access to the work space at Knowit and UiA, where we have done all our meetings face-to-face. It also means we cannot use the equipment that we are allowed to use like the monitors. It is not a major problem, but it makes it more difficult if we are trying to look at tutorials and code. The third challenge relates to the second challenge where we cannot access work space, which causes some technical issues. I for instance have horrible network connection at home, especially when the rest of the family is in meetings. Fourth problem is that we lose face-to-face communication with Erling which is the supervisor at Knowit. Not to big of a deal as he replies to his emails quite fast and helps us when needed, but it would be nicer to communicate in person at feels more social. Last challenge we come across is that we have different work schedules mainly due to Anette having homeschooled kids. So far it has worked fine as everyone on the group are able to work on what they are responsible for.

During the second sprint the group added the Coronavirus to the Risk Assessment Plan. This was almost added as a joke because we did not think it would blow up to be this big and affect Norway the way it has.

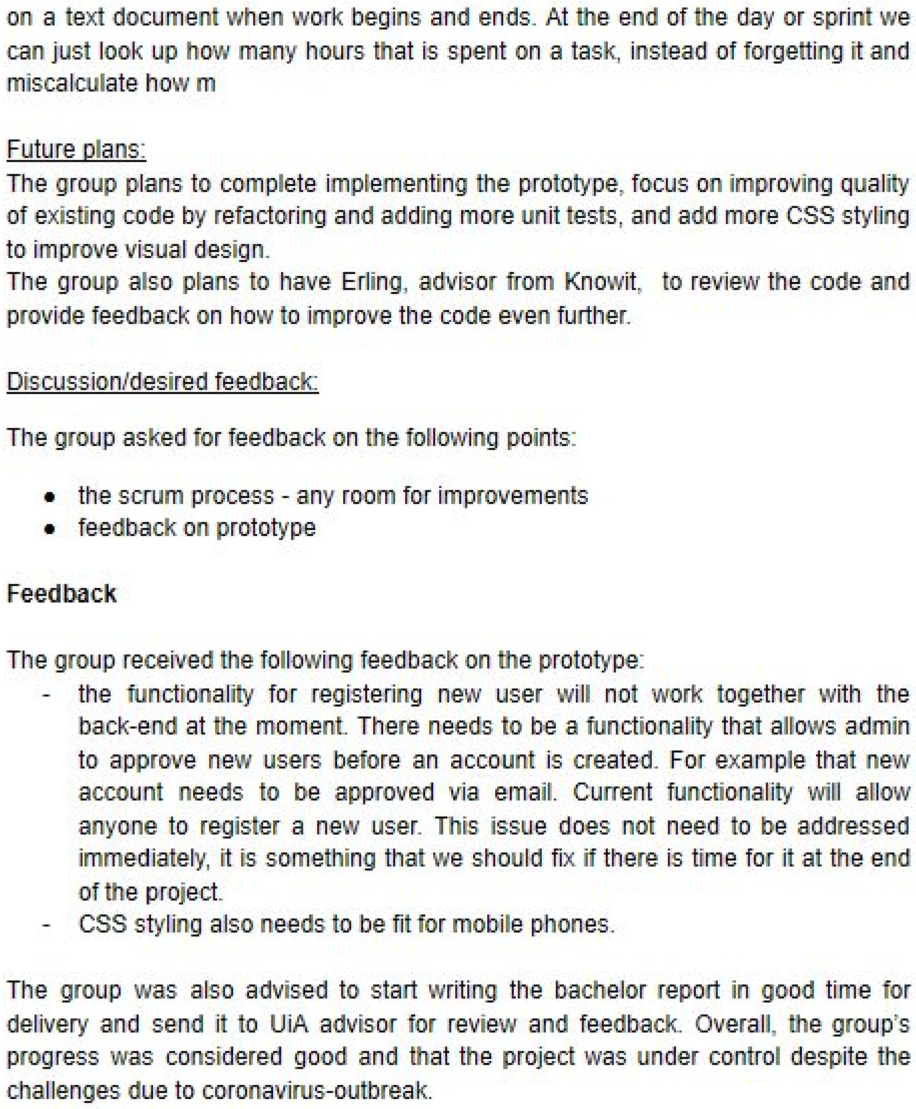
During the third sprint it actually did blow up, the school closed, and the authorities advised everyone to work from home. We stopped meeting down at Knowit's office and everyone in the group set up home offices. On March 16th the group had a voice call crisis meeting to assess the situation and create a plan for how we were going to deal with the situation. We revised the Risk Assessment Plan with updated risks related to Corona. We agreed on having voice call meetings on Mondays, Wednesdays and Fridays at 14.00. We also agreed that we needed daily scrums and to focus on good communication flow.

Other Challenges:
The group spent quite a lot of time trying to understand what testing is, how it works and what to actually test. There are many great video tutorials and articles related to testing regarding how to test and what to test. Most of the time our code looks different than the code in the tutorials. The problem with this is that the tutorials explain their specific code very well, but their explanations do not cover exactly what we are trying to test. With some trials and errors, we have managed to implement some tests on each sprint task.

Some of us forget how much time is spent on a sprint task, this can be a problem as the burndown chart will not get accurate. It is easy to say at the start of a sprint that "I remember how much I have worked on this task, I will write it down later." but then forgetting it, and trying to figure out how much time was spent three weeks later.

Solutions for the first challenge is to ask the supervisor as he is often available. We experience that he answers quickly by email, therefore it is no problem for us to ask him about anything. In the last three weeks we have asked him how to make some tests and what to test. He responded very quickly and made some for us. With some tests in place, it is easier for us to make more as we can look at previously made tests and make similar ones.

To the second challenge, we should use a stopwatch or write down on either paper



on a text document when work begins and ends. At the end of the day or sprint we can just look up how many hours that is spent on a task, instead of forgetting it and miscalculate how m

Future plans:

The group plans to complete implementing the prototype, focus on improving quality of existing code by refactoring and adding more unit tests, and add more CSS styling to improve visual design.

The group also plans to have Erling, advisor from Knowit, to review the code and provide feedback on how to improve the code even further.

Discussion/desired feedback:

The group asked for feedback on the following points:

- the scrum process - any room for improvements
- feedback on prototype

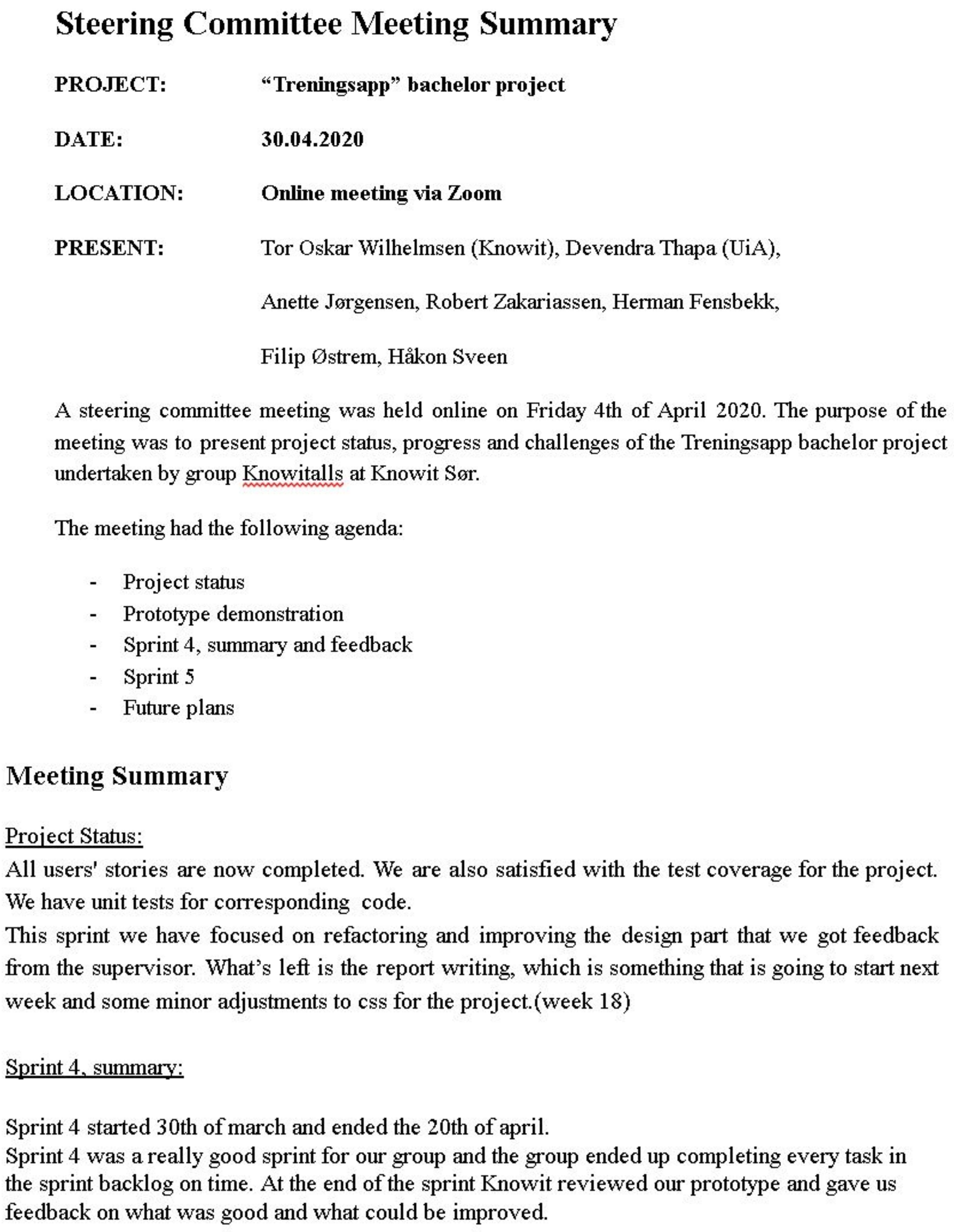
Feedback

The group received the following feedback on the prototype:

- the functionality for registering new user will not work together with the back-end at the moment. There needs to be a functionality that allows admin to approve new users before an account is created. For example that new account needs to be approved via email. Current functionality will allow anyone to register a new user. This issue does not need to be addressed immediately, it is something that we should fix if there is time for it at the end of the project.
- CSS styling also needs to be fit for mobile phones.

The group was also advised to start writing the bachelor report in good time for delivery and send it to UiA advisor for review and feedback. Overall, the group's progress was considered good and that the project was under control despite the challenges due to coronavirus-outbreak.

Appendix 14 - Steering Committee Meeting 30.04.2020



Steering Committee Meeting Summary

PROJECT:	**"Treningsapp" bachelor project**
DATE:	**30.04.2020**
LOCATION:	**Online meeting via Zoom**
PRESENT:	Tor Oskar Wilhelmsen (Knowit), Devendra Thapa (UiA), Anette Jørgensen, Robert Zakariassen, Herman Fensbekk, Filip Østrem, Håkon Sveen

A steering committee meeting was held online on Friday 4th of April 2020. The purpose of the meeting was to present project status, progress and challenges of the Treningsapp bachelor project undertaken by group Knowitalls at Knowit Sør.

The meeting had the following agenda:

- Project status
- Prototype demonstration
- Sprint 4, summary and feedback
- Sprint 5
- Future plans

Meeting Summary

Project Status:

All users' stories are now completed. We are also satisfied with the test coverage for the project. We have unit tests for corresponding code.
This sprint we have focused on refactoring and improving the design part that we got feedback from the supervisor. What's left is the report writing, which is something that is going to start next week and some minor adjustments to css for the project.(week 18)

Sprint 4, summary:

Sprint 4 started 30th of march and ended the 20th of april.
Sprint 4 was a really good sprint for our group and the group ended up completing every task in the sprint backlog on time. At the end of the sprint Knowit reviewed our prototype and gave us feedback on what was good and what could be improved.

The group really improved on the time estimation in sprint 4, and the burndown chart was much easier to read and understand than the burndown charts from the first sprints.
During sprint 4 the group worked a lot on tests and the test coverage was improved a lot during sprint 4. After sprint 4 we had tests on every component in our application.
Even though we had to work separately due to the corona-virus our communication was very good and so was our teamwork.
The group did not really meet any huge challenges during this sprint.

Feedback given at end of sprint 4:

After sprint 4 the group sent in the progress we had made so far to be reviewed byTor Oskar and Erling, knowits front end guy. The goal of the review was to get feedback on what we have done right and what we need to change in the two last oncoming sprints. The feedback we got was good, there were some new tasks and fixes. Some of these you can see below here, but overall I think the group feared there was more to be fixed than the actual feedback revealed.
Overall the feedback was more positive than expected in our opinion.

Feedback on code:

- Overall good feedback
- Minor changes regarding css functionality, code formatting and variable names.
- Validation of user registration is needed.

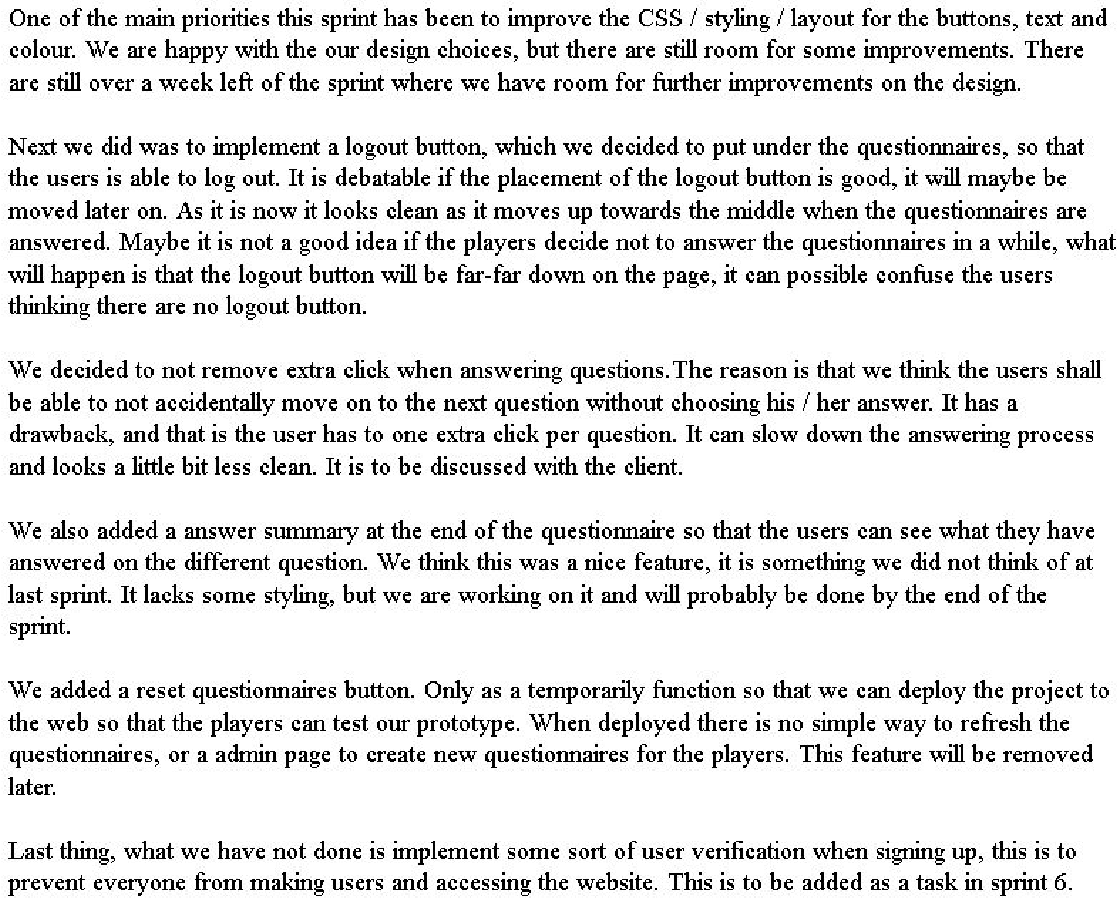
Feedback on design:

- Improve CSS design - current look is a bit “ingeniør design”.
- Add user verification functionality - now everyone is able to log in
- Add logout button
- When answering questions, there is one unnecessary click
- Add a summary of answers at the end of the questionnaire
- Add proper questions

Sprint 5

Sprint 5 started on the 20th of April and ends on the 8th of May. We have to this date been focusing on refactoring code and improving quality according to feedback received from our supervisor Erling and Tor Oskar

As Robert has walked you through the prototype there are some noticeable changes to previously implemented features and new ones that was made accordingly to the feedback received before the start of this sprint.



One of the main priorities this sprint has been to improve the CSS / styling / layout for the buttons, text and colour. We are happy with the our design choices, but there are still room for some improvements. There are still over a week left of the sprint where we have room for further improvements on the design.

Next we did was to implement a logout button, which we decided to put under the questionnaires, so that the users is able to log out. It is debatable if the placement of the logout button is good, it will maybe be moved later on. As it is now it looks clean as it moves up towards the middle when the questionnaires are answered. Maybe it is not a good idea if the players decide not to answer the questionnaires in a while, what will happen is that the logout button will be far-far down on the page, it can possible confuse the users thinking there are no logout button.

We decided to not remove extra click when answering questions.The reason is that we think the users shall be able to not accidentally move on to the next question without choosing his / her answer. It has a drawback, and that is the user has to one extra click per question. It can slow down the answering process and looks a little bit less clean. It is to be discussed with the client.

We also added a answer summary at the end of the questionnaire so that the users can see what they have answered on the different question. We think this was a nice feature, it is something we did not think of at last sprint. It lacks some styling, but we are working on it and will probably be done by the end of the sprint.

We added a reset questionnaires button. Only as a temporarily function so that we can deploy the project to the web so that the players can test our prototype. When deployed there is no simple way to refresh the questionnaires, or a admin page to create new questionnaires for the players. This feature will be removed later.

Last thing, what we have not done is implement some sort of user verification when signing up, this is to prevent everyone from making users and accessing the website. This is to be added as a task in sprint 6.

Future plans

Towards the exam, the group plans to focus on writing and completing the bachelor report. The group will also continue to work on the application to improve and polish it according to feedback. Sprint 6 will be the last sprint of this project, and the team will spend the time preparing for final exams.

Feedback

The group received feedback on the prototype from Knowit regarding several aspects of the application. Overall the feedback was good, and the quality of code was considered very good. Knowit had questions with regards to test coverage and if all parts of the code was tested. The group confirmed that all parts of the code were tested.

In addition, the group received feedback on design and some suggestions for improvement:

- The text was a bit bold and this could cause problems for the elderly and visually impaired.
- The styling should be consistent
- The buttons should look like buttons.
- The background color is a bit too red and should be graded.
- The input fields - make sure no one can perform sql-injection attacks.
- User validation - add a function so that random people are not able to register a user