BACHELOR'S THESIS

FinnFram i Østfold - Orienteering maps Management system for orientering in Østfold

Jonas Vestgarden
Adis Jasarevic
Michael M. Simon

16.04.2018

Østfold University College
Information Systems, Department of Computer Science

ØSTFOLD UNIVERSITY COLLEGE

Department of Computer Science, Remmen
1757 Halden || Phone: 69 21 50 00 || www.hiof.no

Project category: Bachelor Thesis	**X**	Freely available
Credits: 20		Freely available after:
Field of study: Information Systems		Made available upon contractor's consent

Title FinnFram i Østfold - Orienteering maps Management system for orientering in Østfold	Date: 15. May 2018
Authors: Jonas Vestgarden, Michael M. Simon, Adis Jasarevic	Thesis advisor: Børre Stenseth
Faculty / Program: Department of computer science	Group: BO18-G34
Contractor: Østfold Orienteering Association	Contractors contact person: Svend Sondre Frøshaug

Excerpt:
The use of digital technologies and its importance when creating, updating and sharing information has been vital for an effective process and increasing performance in different fields.

Østfold Orienteering Association (ØOA) planned to make a system for sharing maps in Østfold, enhance the competence of the teachers in this field, and increase interests in the orienteering sport amongst the pupils.

This document is prepared to provide information on the process of developing the system.

3 keywords:	
	Web application
	Development
	Orienteering

Abstract

The orienteering sport can contribute to developing navigational skills as well as map and compass experience. Several schools in Østfold teaches the sport for the mentioned purposes. However, most of those schools struggles when teaching the sport, mainly from a lack of access to updated maps, drawing new courses and writing activity recommendations.

On the other hand, Østfold Orienteering association (ØOA) together with different sport clubs has access to updated orienteering maps for several locations in Østfold. ØOA wishes to share these maps with the schools and the community in general. The goal is to make a platform with access to digital orienteering maps, introduce the sport and promote the World Orienteering Championship (WOC) across Østfold county.

This project is an effort of achieving the aforementioned goals. The team developed a web application that can be used for sharing activity recommendations, orienteering maps and allow users to draw their own orienteering courses, or use verified ones, and print the maps along with the courses. Ultimately, the application will contribute to resolving the challenges that are access to orienteering maps and activity recommendations by connecting the association and the community.

Acknowledgment

We would first like to thank our thesis advisor, Mr. Børre Stenseth at Østfold University College. Mr. Stenseth guided and supported the team by giving advice whenever needed through consistent meetings held on a weekly basis.

We would also like to thank Mr. Jens Erik Mjølnerød, Mrs. Elsie Brenne, Mr. Per Bergerud, at Østfold Orienteering Association, Mr. Svend Sondre Frøshaug at the World Orienteering Championship and all teachers we spoke to for giving their valuable time to help the group.

Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 The group

All members of the team are third year Information systems students with a diverse interest and competence in IT.

Name: Jonas Vestgarden | Birth: 1993 | Location: Halden

Jonas is information systems student. He is interested in programming, modelling and consultation. After completed his bachelor's degree, he wishes to pursue a career as a consultant, software or game developer.

Name: Adis Jasarevic | Birth: 1995 | Location: Sarpsborg

Adis is interested in programming, administrating and modeling databases, 3D modeling applications (Maya and cad) and as well as some marketing and music production. He wishes to specialize in machine learning. He will study further for master's degree and collaborate with his brother who also graduated from Østfold University College.

Name: Michael M. Simon | Birth: 1994 | Location: Halden

Michael is interested in human computer interaction(HCI), user experience, Front-End development. He's currently managing his own private business which mainly deliver services in graphic design for print and digital platforms. He wishes to continue developing his business with a variety of services in web technology and continue to study at the University of Oslo in Interaction design.

1.2 Project owner

Østfold Orienteering Association (ØOA) is an active association hosting multiple official competitions and training runs each year. ØOA is an affiliate of the Norwegian Orientation Association (NOA), and was established in 1940 (Jens Erik Mjølnerød, 2016, s.7).
The association is surrounded by active and engaging community and wish to increase the participation of youths and childrens.

1.3 Task description

A majority of schools in Østfold include the orienteering sport in their course plans. The purpose of introducing the sport is to teach the pupils in use of maps and increase their map reading ability by applying it as an enjoyable and inclusive activity.

The purpose of the project is to initiate physical activity among pupils, expand interest and spread awareness of the World Orienteering Championship which is going to be held in 2019.

ØOA and the sport clubs in Østfold has high quality digital maps which they wish to provide to elementary schools located in Østfold. The system is designed to solve challenges around access to updated maps and serve as a platform for sharing activities and related information.

Users will be able to upload, share and draw control points and directions on the map. ØOA and approved users will be able to upload the maps, post activity suggestions and other related information.

Furthermore, the map edit feature will also be used to draw control points and write other relevant information on the map. Currently, ØOA uses a third-party application called Purple Pen for drawing on the maps. The drawing functionality was suggested by the group members with the intent of removing the need of third party applications for drawing control points and other information that are required in orienteering maps.

The application runs on the web using compatible browsers with JavaScript enabled. This is known as a web app or web application. "A web-based application is any program that is accessed over a network connection using HTTP, rather than existing within a device's memory. Web-based applications often run inside a web browser. However, web-based applications also may be client-based, where a small part of the program is downloaded to a user's desktop, but processing is done over the internet on an external server." (Technopedia.com, n.d).

1.4 Goals, deliverables and methods

1.4.1 Goal

- **Goal** Develop a platform for sharing and drawing orienteering maps that will be used in different municipalities in Østfold country. Users of the platform will be able to;
- **Sub goal 1** Upload, store, download, edit and print orienteering maps.
- **Sub goal 2** Select a map based on a selected location.
- **Sub goal 3** Suggest activities and materials for supporting the learning process.
- **Sub goal 4** Access to all maps available in the system.

Simple and effective
The application should be simple to use, intuitive and utilize comprehensible linguistics.

User friendly
The application should be developed considering best practices regarding user experience, technology and standards.

Accessible
This project will be developed under the guideline specified in Web Content Accessibility Guidelines(WCAG) 2.0.

Responsive design
The application will be responsive. Which means that the user experience will be identical regardless of access device.

1.4.2 Deliverables

The outcome of this project will be a web-based application that contains a variety of functionality such as uploading, sharing and editing maps.

This application will allow users to have a common place to view and share orienteering maps for several locations in Østfold. All the maps will be saved in a database and the application will be available for everyone.

The team will be following up on the progress of the project from the beginning to the final delivery. Guidelines will be used for project phases to measure accomplishments and set goals. Each of the project phases will be performed iteratively by delivering small features of the application known as sprints.

Overview: Deliverables	
UI sketches	A simple sketch of the application user interface.
UI design	A more complete design of user interface with more features of the application added, that will be improved based on the ØOA's feedback.
UI design prototype	A final mockup with most of the application feature. It will be presented to ØOA to confirm the core features of application and consent on the interface design and navigation.
View maps	Feature of the application where users can access the maps.
Database configuration and setup	Models and configuration and structure of the database when registering users, uploading maps and managing articles.
Drawing feature	A feature that allows users to draw control points and other relevant information on the map.
User register	Registration and login page.

User authorization	Authorizes the users to access features of the application.
Token and session	Metadata stored in the browser.
Admin	Admin page will be used to manage the application, post articles and activities.
Content management feature	Articles and activity suggestions can be organized by an admin and posted from this page.
Upload maps	Users that are responsible for uploading maps can use this functionality.
Save maps	Users that access maps will be able to save the maps to their computer and the application.
Complete drawing functionality	A complete version of the application where users draw on the map.
Final product	The final product that complies with the specification

Table 01: Overviews of the deliverables.

1.4.2.1 Project phases

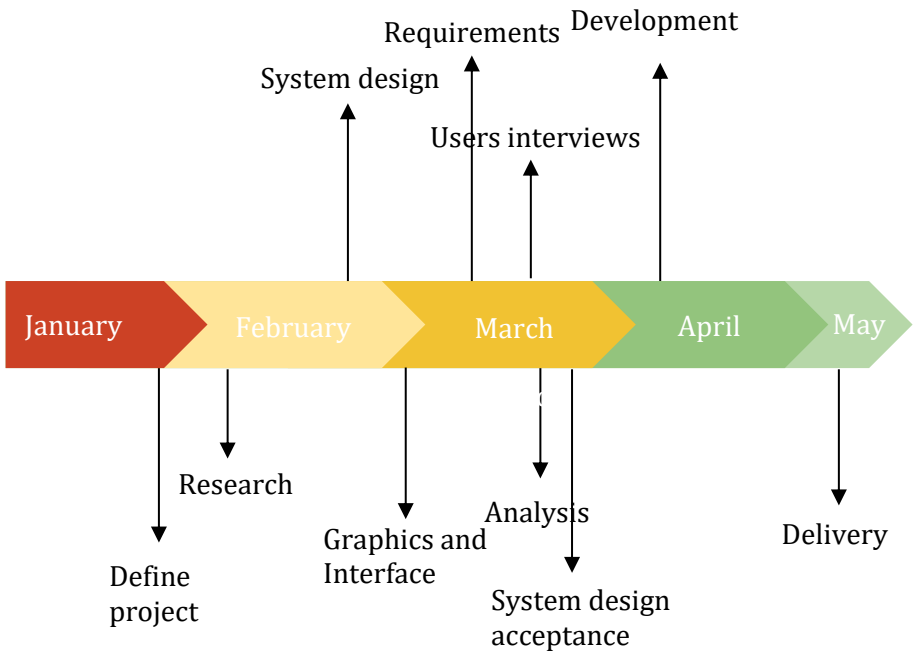


Figure 01: A timeline of the overall project phases.

Defining the project
After receiving the project proposal, the team focused on understanding orienteering sport and the activities performed. They discussed the project goal, specification, outcome and benefits of the project to different stakeholders. These stakeholders are schools, ski clubs and orienteering championship organizers.

Technical review and analysis of the specification was performed at this stage, the team measured the requirements based the following factors:

1. Time
2. Technical feasibility.
3. Importance of the required features.
4. Integration of the features to the required platform(web).
5. Capability and skill of the team to complete the tasks.

Research
The team performed preliminary research on the technologies to be applied and how those technologies would assist solving the goals. Meanwhile, the team tested the technologies by making sub functionalities of the application. This helped the team apply a systematic approach when selecting suitable technologies and tools.

System design

At this stage, the team made some of the use cases and scenarios, completed the iteration of the UI mockups and got confirmations from ØOA. The team started working on the details of the technical and functional requirements, defined schemas for the database and set the development environment. These elements were continuously iterated and improved upon in each sprint.

Development
Development was performed in iterations where the team designed and shipped small features of the application and received feedback accordingly. Based on the feedback the team decided if a rollback was required, or if the feature was complete. Improving the UI design and sample features were also a part of the development phase of the application.

1.4.3 Method

This section will describe the development methods, project management approaches that were used and how we tuned the traditional methods to fit the project and the team. Furthermore, it elaborates on the effectiveness of the approaches utilized.

We agreed to adopt different methodologies and frameworks that can be merged together such that we could apply unorthodox development practices that best fit the team. This decision was taken based on two factors.

First, the team want to use this opportunity to experiment and learn at the same time.

One of the strengths of the learning process is that it allows for experimenting and learning from failures. These are salient for both self-development and creativity.

Secondly, the team had a big project considering the team size and there were several responsibilities and roles that to be filled. The team took care of communication with stakeholders, management, research, planning, reporting, UI sketching and design and programming in several parts of the application from database to back-end to front-end. By taking into consideration the aforementioned factors, the team had to come up with an approach that best fit the team's size and the project.

The team used the scrum framework for development and attempted to combine it with design thinking. "Scrum is an empirical approach applying the ideas of industrial process control theory to systems development resulting in an approach that reintroduces the ideas of flexibility, adaptability and productivity." (Peka abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, 2002, page. 29).

The team used a design thinking approach when solving problems regarding the features of the application, user interactions and decisions related to usability and experience.

"Design thinking is that designers' work processes can help us systematically extract, teach, learn and apply these human-centered techniques to solve problems in a creative and innovative way – in our designs, in our businesses, in our countries, in our lives." (Dam & Siang, 2018, n.p)

Scrum

We focused on planning out the features and improve on them while iterating. Each iteration includes planning, building, testing and reviewing based on their performance and compliance to the specification.

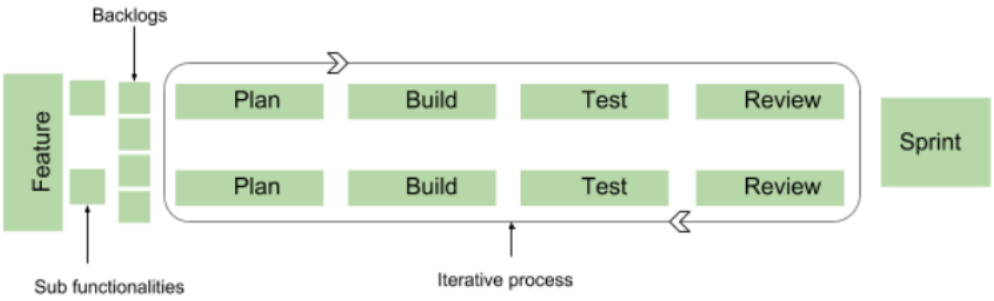


Figure 02: Illustration of the development process

A set of functions and modules together make up a complete feature. These portions are referred in figure 02 as sub-functions. The sub-functions contain a variety of tasks that should be built referred to as backlogs in figure 02.

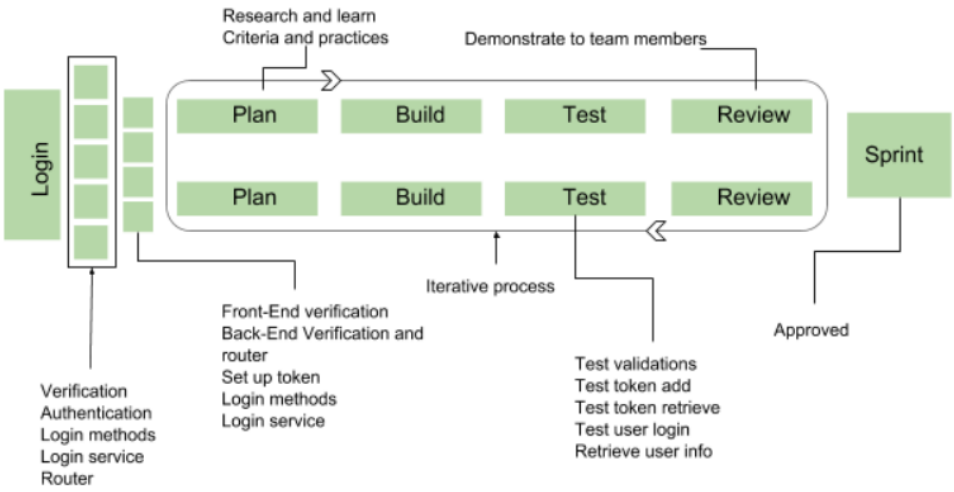


Figure 03: Development process example

Design Thinking

Design thinking is a user-oriented design strategy that can be applied when designing a solution with the purpose to solve a problem. The design thinking process contains five-stages; empathize, define, ideate, prototype and test. The team iterate through these stages when finding out and denoting the users pain points.

Figure 04: The five stages of design thinking

1.5 Report structure

This document is structured based on the template provided by HiOF and further modified based on the team's needs. The team decided to write the document in English because of the multicultural background of its members.

The overall structure of the document is composed of 9 chapters. These chapters discuss the entire project process and the result of the project under their distinctive sections.

Chapter 1 - Introduction: An introduction of the group members and their affiliations, the project owner and their organization, the task, goals and methods used to achieve the result.

Chapter 2 - Terminology: Different terminology used throughout the document can be found here.

Chapter 3 - Project process: This chapter details how the group worked during the project period. What decisions was made, what were they based on and ultimately how they were implemented.

Chapter 4 - Analysis: An analysis of the research performed prior and during the project as well as the interviews, and the results.

Chapter 5 - Design: The system design, process, implementation and related decisions taken based on the requirements specification.

Chapter 6 - Documentation: Documentation of the most relevant code written and the application's architecture.

Chapter 7 - Testing: Test plans and criteria.

Chapter 8 - Discussion: A discussion of deliverables, implementation, the process, the final product and further work required.

Chapter 9 - Conclusion: A summary and conclusion of the project.

Chapter 2 - Terminology

These are definitions and terminology used inside this document. The document contains several technical terminologies that may require additional research to understand.

2.1 Programming languages and data formats

CSS
CSS stands for Cascading Style Sheets and is primarily used to modify the attributes and rules of HTML elements. This changes how a standard HTML page is displayed to the user and is a powerful tool when creating intuitive and accessible solutions.

HTML
HTML stands for Hypertext Markup Language and is used to describe the layout of a web page. This is the bottom layer of a web page and is accessed by other languages by using the tags associated with the desired block.

JavaScript
JavaScript is an object-oriented programming language and is one of the foundations for most web development solutions, usually used in conjunction with CSS and HTML.

JSON
JSON stands for JavaScript Object Notation and is a text based, simple way to store JavaScript-like objects. The simple syntax is to allow human readability and easy access for JavaScript objects to be stored and retrieved for later use.

SVG
SVG stands for Scalable Vector Graphics and is an XML-based file format to describe two-dimensional vector graphics. It's a basic part of HTML, but it can't be modified in the same way as other tags using CSS for example.

Typescript
Typescript is open-source programming language written on top of JavaScript. Most notably, it features the possibility to declare variables as types, much like in java and other object-oriented programming languages.

2.1.1 Libraries

Svg.JS
Svg.JS is a "lightweight library for manipulating and animating SVG." (Svg.js, n.d). It offers additional functionality otherwise unavailable in the standard SVG format, to allow for easier customization, manipulation and creation of SVG elements.

2.2 Framework

The application is mainly written in Typescript using the MEAN stack in NodeJS.
It uses a NoSQL database called MongoDB to store data in JSON-like documents, and Angular to display the client. MEAN stands for **M**ongoDB, **E**xpressJS, **A**ngular, **N**odeJS.

MongoDB
MongoDB is an open source NoSQL distributed database using JSON-like documents. NoSQL databases were made to deal with limitations in the SQL databases, mainly scalability, multi-structured data, geo distribution and agile development sprints (What Is MongoDB?, n.d.).

ExpressJS
ExpressJS is a lightweight framework built on top of NodeJS to help organize the web app into an MVC architecture. ExpressJS har various features built in to help manage requests, routes, views, API etc. (Express, n.d.).

Angular
Angular is an open source JavaScript framework developed by google. It is "what HTML would have been, had it been designed for applications". The current version and the one we're using, Angular 5, is written in typescript. (Angular, n.d.).

NodeJS
NodeJS is a JavaScript runtime built on Chrome's V8 JavaScript engine. NodeJS uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. NodeJS facilitates the use of JavaScript in front-end and back-end of the application structure and built using C# and JavaScript at its core. (NodeJS, n.d.).

V8
V8 (or Chrome V8) is a JavaScript library written by Google for chromium-based browsers. It's an incredibly fast engine as it compiles JavaScript code directly into native machine code prior to executing it (Chrome V8, n.d.).

Material Light
Material Light is a library that is developed based on the android material design principles. Material light is developed with a purpose of creating new interactions of user and elements by transforming the behavior of real world objects, and their natural features like shadow and distance, into the digital world.

2.3 Design

WCAG 2.0
The web content accessibility guideline sets a standard for web contents accessibility. The guidelines contain twelve points that is a minimum requirement when developing a website that are accessible to different types of users despite their physical and cognitive ability (W3C, 2008).

Adobe XD
A desktop application aimed for producing, editing and sharing sketches and UI mockups.

User Interface (UI)
The application interface that users interacts with.

2.4 Orienteering

Control point
A control point (CP, also called control and checkpoint) is a marker used to determine directions and destinations used in the orienteering sport.

Goal
The final control point in an orienteering course.

Start
The starting control point in an orienteering course.

Orienteering
Orienteering is a sport that is performed by combining map and compass for navigation. It's easy to learn, but always challenging. "The object is to run, walk, ski, or mountain bike to different points displayed in the map, choosing routes, both on and off trail, that will help you find all the required points and get back to the finish in the shortest amount of time." (Orienteering, n.d.).

"The points on a course are marked with orange and white flags and pouches or electronic devices, so you can prove you've been there. Each "control" marker is located on a distinct feature, such as a stream junction or the top of a knoll" (Orienteering, n.d.).

2.5 Miscellaneous

Front-End
The part of the application users interacts with.

Back-End
The server side of the application which is responsible for sending and receiving http requests and store data.

Encrypt
Converting data into obfuscated codes with a purpose of making data secret from unauthorized parties.

Decrypt
The reverse action of encrypting, converting obfuscated code into readable data.

Token
A set of characters stored inside user's local storage in the browsers. The characters can be used to track whether users are logged in and logged out.

Validation
Making sure the user's inserted information meets the criteria.

Schema
Models to a set of data to be stored.

Component
A set of elements that together make up a web page. In angular the html, CSS, JavaScript etc. files are together referred to as components.

Karma
Auto test runner for JavaScript applications.

Single page application (SPA)
A web app that can navigate to other parts of the application, retrieving data and performing actions without having to reload the page.

Route
For the application to be SPA, routes are required. Routes are responsible for sending the data from one part of the application to another.

Content delivery network (CDN)
A CDN is a service that provide fast delivery of external web content. They are commonly used in conjunction with HTML to provide additional functionality to your website.

Chapter 3 - Project process

This section elaborates on how the team planned, executed those plans and the process of developing the application. It includes several decisions the team took through discussions held with the stakeholders, how the team scheduled, delegated responsibilities and organized the project by applying different practices and methods.

3.1 Overall process

After receiving the project, the team focused on gathering necessary information related to the orienteering sport. Furthermore, they commenced studying the domain, the purpose of the sport and the expectations of ØOA and World Orienteering Championship (WOC).

The team later held a meeting with ØOA where their goals and expectations for the final product were explained. During one of these meetings the initial project description was discussed, however it was explained that this was more of a brainstorm than an actual project description. Considering the deadline of the project and estimated workload, the group suggested prioritizing features based on their value the achieving the goals. Together with the team, ØOA came to an agreement to reconsider the complexity of the specification.

While waiting for the new specification the group planned out sub features of the application, for instance the map drawing feature, and tested out SVG as a drawing method. The samples were published on frigg(HiOF's server) and was available for all stakeholders to follow up on. The goal of making the sub features available was to give continuous access to the project so that the stakeholders could view, evaluate and give feedback on the progress of the application.
The sample application helped the team experiment on several development tools and programing languages. Subsequently, the team had no issues selecting the ones that would best suit the application.

Because the team valued the stakeholder's involvement in the project, it was imperative that they maintained continuous communication with them. Research and discussions were held frequently to figure out which technology and management methods were essential.

The interviews with the teachers helped the team understand the users' challenges, gave an insight as to how the orienteering sport is applied in schools, prioritizing functionalities and supporting design decisions.

Some of the obstacles that occurred implementing the scrum framework was the amount of team members, the work environment and the setting. Therefore, scrum was not fully implemented, and modifications were made by applying other elements that were suitable, for example Design Thinking.

Details on implementation of the aforementioned processes are discussed in chapter 3.2 and further in chapter 4.

3.2 Planning and Research

Planning and research was essential for the project and the decisions taken.
"Plans help us know who needs to be available to work on a project during a given period. Plans help us know if a project is on track to deliver the functionality that users need and expect. Without plans we open our projects to any number of problems."
(Cohn, 2005)

The project phases in figure 01 is an estimated overview used for defining the approximate deliveries to be achieved in their respective time periods. It's an overview that helps the team and stakeholders answer questions like what the team is currently working on, what's coming after and when the project is to be delivered.

According to Cohn, plans helps us reduce risk, reduce uncertainty, support better decision making and establishing trust. Correspondingly, the stakeholders always knew what the team was doing which established better communication with ØOA.

Release plan

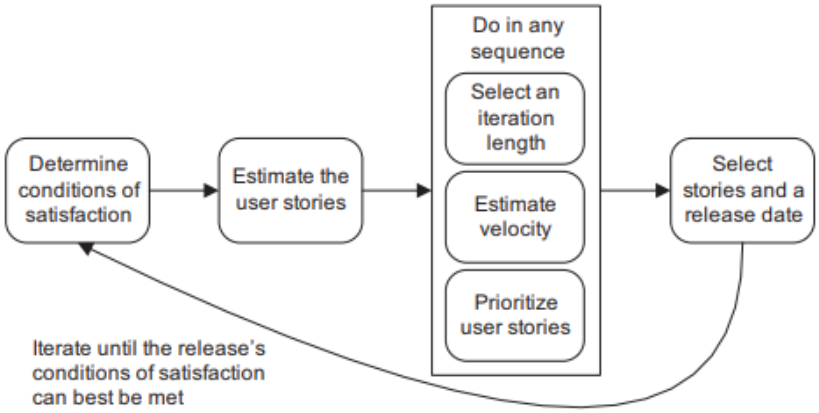


Figure 05 Release plan suggestion by Cohn (Cohn, 2005, p. 131)

Figure 5 illustrates the release plan suggested in Cohn's book "Agile estimating and planning"
The first step of release plan according to Cohn is to determine conditions of satisfaction, this is stated as criteria in our design specification, system design (chapter 5), conceptual design and the function requirements (chapter 5.3) are examples of conditions that must be mate to approve release.

We choose to make use cases and that are more detailed and indicative to the user stories (chapter 5.2.2). The reason for that is a lack of knowledge to mean stack can be troublesome after we began with the development, to have a detail use cases gave us a clear understanding of the elements required, helped us search tools and achieve the goals of each iteration.

Sprint plan
After started with coding, we applied significantly different approach than the overall project plan illustrated in figure 01. The plans were a sprint based known as sprint plans. Sprint plans focus on what will be delivered and when it will be delivered. They are only made for the small features of the application. These features were available on Trello in sprint backlog.

Because the team was working with new technology, and this technology was not part of the curriculum, they had to learn while developing. A simple strategy to solve this was to integrate learning and research in the sprint iterations, as illustrated in figure 06.

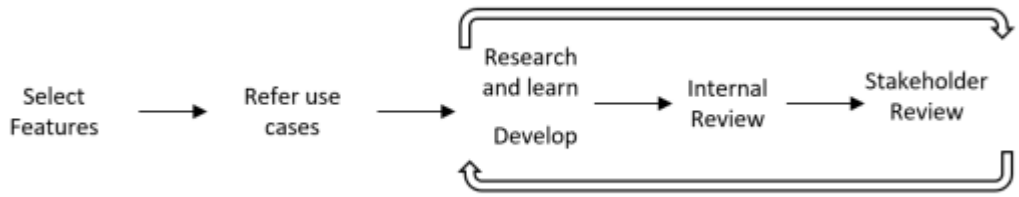


Figure 06 Iterative process plan of sprints.

Research
Researching was critical in all aspects of the process of the project. As mentioned earlier we didn't had experience with angular, typescript, MongoDB and Nodejs environment. All the team members had to read the documentation, articles and watch several tutorials while building the application.

The other aspect of the research performed is the selection of appropriate technologies that both meet the requirements set for tools and technologies (chapter 4.4) and interesting to learn.
The researches focused in three main areas, these areas are:

1. Development methodologies and technology research
2. Programming related researches
3. Interviews for defining user needs

3.3 Methodologies

Merriam-Webster defines a methodology as "a body of methods, rules, and postulates employed by a discipline : a particular procedure or set of procedures" (Definition of Methodology, n.d.).

3.3.1 Design Thinking

Design thinking helped the team define the users' needs and desires that are required to make a complete solution. By integrating the principles of design thinking, the team designed the features required and created the UI. The initial design process was

performed using papers and pen. The team discussed the design features and iterated throughout the process by improving the design elements.

3.3.2 SCRUM

Scrum is a framework for structuring tasks and processes in a development team. Scrum can be utilized to deliver software product in medium sized to small teams. Sprints are small portions of the application that the team produces, they are features that are shipped in a one to two weeks of an interval in an iterative development process. Each sprint can be reviewed and if they are not confirmed by stakeholders or do not meet the requirements, the iterative process can be performed repetitively.

Scrum development suits a team of three to nine persons, and the team members are delegated tasks grouped in three. These groups are scrum master, product owner and development team. Scrum increases the effectivity of project processes and development activities, but it may cause overlapping of roles in a team with as few members as ours. The team decided to implement the suitable parts of scrum while dropping the incompatible ones.

3.4 Implementation of methodology

A variety of tools such as Git, Appear, Trello, Google doc, Facebook and Discord were used during development. These tools were used for task management, documentation and communication inside the team and with stakeholders.

Trello were used for posting, controlling completed and undergoing tasks. Tasks that were completed was marked accordingly and they were grouped in five categories, back-end, front-end, administrative, testing and database.

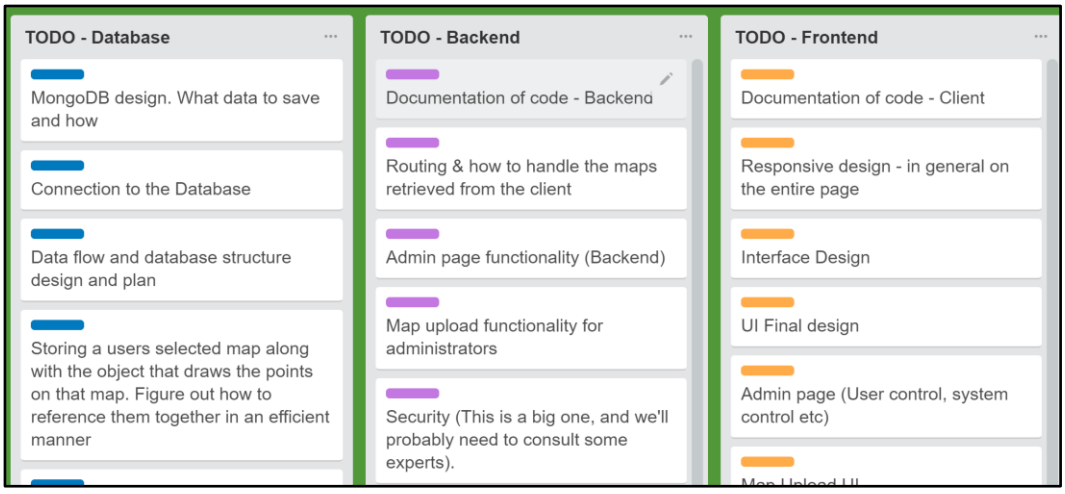


Figure 07: Trello backlog

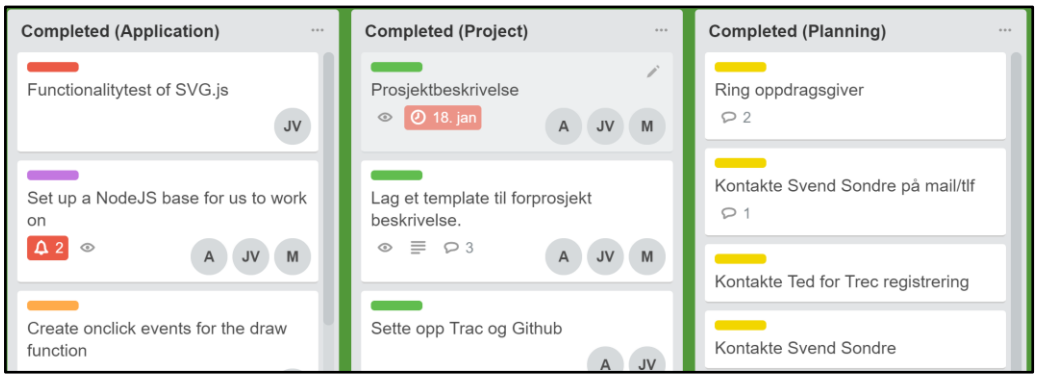


Figure 08: Completed tasks on Trello

Meetings

Weekly meetings were held with our thesis advisor, we discussed issues that occurred which were resolved accordingly. These meetings were also used to brief our thesis advisor on the progress of the project.

In addition to those meetings, the team members had three additional meetings per week. These meetings usually lasted fifteen minutes, in which the group discussed weekly plans, completed tasks, contacts with stakeholders etc. The meetings were also used to discuss a variety of issues group members encountered and to discuss solutions.

Four meetings were held through appear.io with the project leader where the team presented the sprint releases thus far and received feedback accordingly.

Chapter 4 - Research and Analysis

In this chapter we will discuss the choices we took, how we came about selecting the tools and libraries and what implications those choices have. We will also discuss the interviews and analyze the results.

The purpose of this analysis was to further support our choices and decisions when creating the application.

4.1 Interviews

Hereunder we will discuss the method and purpose of the interviews.

4.1.1 Interview method

There are mainly three ways to perform a qualitative interview: Structured, semi-structured and unstructured interview (Graham Crow, 2013, p.12).

As the names implies: A structured interview means having a set of questions, often on a form, for the interviewee to answer, leaving the questioner little room for modifying them.
The purpose of a structured interview is to gain the same type of information from a large number of interviewees to compare the data together (Graham Crow, 2013, p.13).
A semi-structured interview has a set of questions very much like a structured interview but offers flexibility when asking the questions. The interviewer can ask the questions in any order they'd like and follow up with their own questions to pursue a particular topic of interest. This allows the subject to answer in their own words, but the answers still have some semblance for comparison (Graham Crow, 2013, p.13).

The last one, an unstructured interview, doesn't necessarily have a set of questions, but rather a topic to cover. This allows the interviewees to answer in a way that is comfortable to them, allowing them to reflect on their own ideas while not being influenced by the questions you ask. This is arguably the hardest interview to hold as you as the interviewer need to keep the conversation going, while trying not to influence the interviewee when asking about a particular topic (Graham Crow, 2013, p.13).

For our purpose, we chose a semi-structured interview with a set of questions. A semi structured interview was more open to discussion than a structured interview and we didn't need to worry about derailing the interview. Having a set of questions also allowed us to categorize the data such that we could draw comparisons with the results from other teachers and studies.

4.1.2 Purpose

The end-user interviews were conducted with a purpose of discovering the users need. To find out specific functionalities that users might desire but may not have been considered when the specification was written by ØOA.

The interviews were also utilized to support the decisions made when designing the specification.

4.2 Interview Analysis

4.2.1 Analysis method

A qualitative analysis method was applied to analyze the interviews with the end-users.

"Qualitative data analysis is a process that seeks to reduce and make sense of vast amounts of information, often from different sources, so that impressions that shed light on a research question can emerge." (6 methods of data collection, page 13).

Since qualitative data is the opinions and interpretations of the interviewees, we needed a way to systematically group it together in a way that allow us to draw comparisons. One approach is called a "*thematic network analysis*" and is the process of "coding" all your data. Coding simply means you condense and categorize the data to sort the answers such that they can be compared against each other later (6 methods of data collection, page 14). For our interviews, we chose to go with the second approach.

4.2.2 Coding and describing data

To better structure the research data, we indexed it prior to coding it. Indexing is the process of applying "*meaning to raw data by assigning key words or phrases*" (Bloor & Wood, 2008). In other words, picking the important parts of an interview. We then had a basis of comparison between the interviews and could settle on categories that would describe certain aspects shared between the interviews. The following are the categories we decided on:

1. "How is the activity performed today?"
2. "What is required to perform the activity?"
3. "What works badly with the current implementation?"
4. "How can our product help the pupil?"

With category 1 we wanted to figure out the current practices of teaching orienteering to pupils. Not necessarily to conclude that it was a poor approach, but rather to draw inspiration from what works well, learn from the mistakes of what didn't, and make sure the changes we made weren't too drastic. Ultimately it was not a team of engineers that were going to spend time teaching this activity, which meant our application had to be easy to use and intuitive, and not alienate teachers who had a lack of experience with such technology.

With category 2 we wanted to know if there were any essentials we did not already know of that was part of the activity and could solve using the application. For example, since our application was purely a drawing, sharing and information tool, it was still required

that the teachers put physical markers at the spots marked on the maps. However, were there any other requirements that would also need to be handled in the application?

With category 3 we wanted to make sure we did not repeat problems that existed with the current solutions. Were there any ways our application could reduce or completely eliminate said problems?

With category 4 we wanted to know how our solution would contribute to ØOA's goal, that is teaching the pupils and raising awareness about orienteering. For example, giving the teachers an easier time to teach orienteering and collaborate between themselves would give the pupils more knowledge about the activity.

4.2.3 Results

With the data categorized, hereunder are the results of the interviews. These are the indexed versions of the answers written in Norwegian because the interviews were held in Norwegian and translating them would subtract from the authenticity of the answers. The full transcript is available in the attachments:

How is the activity performed today?
Interview 1:

- Har en mappe med gamle kart de må bruke, med mindre dem lager sine egne
- Orienteringsklubben/Skiklubben har vært en stor del av opplæringen pga. utstyr og kunnskaper rundt emne
- Er vant til å tegne kart selv for hånd fordi gamle kart er dårlige, ofte med hjelp av google maps eller andre karttjenester på nett. Veldig tidkrevende.

Interview 2:

- Lærerne har gamle og utdaterte kart og noen poster. Utstyr er ikke det største problemet.
- Noen kart er ferdig markert.

What is required to perform the activity?
Interview 1:

- En bank med alle kart tilgjengelig hadde løst mange problemer med orienterings opplæring
- Trenger at undervisningen er litt "proff"
- Vil ha tilgang til alle områder i nærheten, vil ikke bli begrenset til kun skoleområdet, slik at man kan ta med seg klassen vekk fra skolen
- Lærerne har mye å gjøre, artikler og resources til å gå med kartene hadde lettet byrden dems veldig og gjort mye av arbeidet for dem

Interview 2:

- Enklere at lærerne koordinerer på skolen og at de har orientering samme uken. Blir mer naturlig for dem og bruke mer tid på det.
- Sikkerhet.

- Mer utfordring med tanke på passe vriene løyper og oppsetninger samtidig at det ikke blir for vanskelig for elever som ikke er fysisk sterke eller har funksjonsnedsettelser.

What works badly with the current implementation?
Interview 1:

- Lærere har lite utstyr tilgjengelig
- Lite kontroll på hva andre lærere gjør
- Ikke alle barn er interessert i fotball, derfor må gymtimene være varierte for å få frem talenter
- Har lite penger å bruke på utstyr

Interview 2:

- Tidkrevende pga mangel på oppdatering og forenkling.
- Lærerne eier 40 år gamle papirkart
- Noen lærere er flinkere enn andre når det kommer til orientering
- Løper samme løype i 5 klasse som i 7 pga. lite variasjon i kart

How can our product help the pupil?
Interview 1:

- Elevene synes det er gøy fordi det er noe dem ikke gjør veldig ofte
- Synes det er lurt av klubbene og satse, veldig mange barn er interesserte men det er dårlig lagt opp for øyeblikket
- Lærere blir mer motiverte av å få opplæring. Blir de flinkere selv er det morsommere og de gir bedre opplæring for barna

Interview 2:

- Alle elever som har orientering på skolen trives med det
- Å gjøre det lettere for lærerne gjør at de kan ha orientering oftere

4.2.4 Interpreting the results

By conducting this interview, we hoped to answer certain questions about current orienteering practices in the schools. The intent was to gain some insight into their current practices. Was there additional functionality desired that we did not already provide, how often were they performing the activity, what challenges were associated with it and could any of them be solved using technology? While most of these questions were already answered by our contractor, we realized the teachers might have different answers to the same questions. Ultimately the teachers are the main users of the application, which we considered important especially when designing the user interface and prioritizing functionality.

Challenges around implementation of orienteering does not seem to be exclusive to one school in particular, though certain issues are more prevalent in some than others. Both interviewees stated that the maps they had available were usually very old with points and directions already marked on them. This brought some difficulties because the teachers wanted to change the courses between the orienteering classes. It wasn't fun

for the teachers nor the pupils to keep running the same course every time, and since the pupils had at least one course each year, they ended up doing the same course in 5th grade as they did in 7th grade. This was one of the issues that contributed to less orienteering courses being held each year.

One of the interviewees stated that they had the option to find and create their own maps by selecting a location on Google Maps, downloading and drawing on that map manually. However, they also said that the other teachers were not experienced enough with computers to do this themselves. After the interview they stated, "most teachers here don't even want to touch a computer".

There was a huge disparity on knowledge about orienteering among the teachers. Both interviewees stated that while a few of the teachers knew a lot about orienteering, most had almost no knowledge at all. As a result, there was a great desire for external resources to create these courses, especially when considering pupils with disabilities. They would like external materials that explain best practices, rules and guidelines to follow when setting up a course such that it's suitable for the pupils' needs. This was another issue that contributed to less orienteering courses each year.

Orienteering was an activity the pupil loved in general because it's not something they got to do very often. In most PE (Physical Exercise) classes the pupils play football and other general gym activities. This was mostly due to teachers not having the time to prepare everything an orienteering course required. For this reason, the teachers wished the tools were more readily available such that they could be better at planning, learning and teaching orienteering more often.

There are certainly issues the teachers must deal with when teaching orienteering and while the application won't solve all the problems, it should at least help reducing them. Having the maps all available in a digital format and being able to draw new control points on them, solves the issue of old paper-based maps. Having resources tied to the different courses and readily available, solves the issue of teachers lacking experience when dealing with those issues. Being able to distribute the maps amongst themselves, allows the teachers to collaborate on the courses. Solving these issues should help raise awareness around orienteering, recruit more children and provide a greater variety in PE classes for the pupils.

4.2.5 Limitations of the interview data

The group felt lucky to even have a single teacher agree to an interview, though two interviews was not enough and as a result there was a lack of data to analyze to come up with anything conclusive. The data retrieved was still useful to affirm our decisions designing a user interface, and to verify the functional requirements. Further research was required to cover all aspects of user interaction, any desired functionality not currently implemented and additional resources for the users.

4.3 Current solution

Based on the interviews with the teachers we understood that the teachers faced several challenges when trying to apply the orienteering sport into their PE classes. Among those, a limitation of access to orienteering maps and well-established activity plans are mentioned. While we couldn't generalize all schools in Østfold based on the results from a few, there were indicators that other schools struggled with the same issues. This was further reinforced when talking with our contractor, project leader and other entities affiliated with ØOA, as well as from schools that declined our interviews because they did not teach orienteering.

As mentioned in the task description, ØOA already had digital maps available used for orienteering competitions. These were maps created by professionals using OCAD, which is a 3rd party application used for creating detailed orienteering maps (OCAD, n.d.). They were further edited in a second program called Purple Pen to place control points and control point information. While this was a solution that worked for them, it wasn't very efficient. Our team wish to reduce the use of multiple third-party applications. Based on that, the group suggested to replicate at least Purple Pen's drawing functionality in the application developed.

4.4 Technology criteria

There are several factors that affected the selection of technologies used in the project, we used different technologies depending on what part of the development we were in. Preliminary research, group/project meetings, the task description and early/final requirements specification just to name a few.

4.4.1 Preliminary research

As mentioned in the overall process in 3.1 in chapter 3 "project phases", the group researched what tools and technologies would best fit the goals. The team discussed the pros and cons of different solutions.

4.4.2 Drawing format

In one of our early meetings with our thesis advisor we discussed some good ways of drawing in a web application. Our thesis advisor suggested we could use SVG to draw because it's easy to use and learn. It is also natively supported in html5, making it a great choice for drawing elements. Since we needed some additional functionality for certain parts of the drawing (assigning attributes to an SVG element, storing it as an object etc.) we decided to use a library to expand on the basic SVG functionality, instead of writing everything from scratch, svg.js was chosen because it provides this additional functionality and then some.

4.4.3 Selecting a platform

WordPress was suggested by a representant of WOC however, research performed indicated that WordPress doesn't work very well with client-side JavaScript. The best brute force solution was to include scripts in the footer tag, but accessing certain tags, sending JavaScript objects to the database etc. seemed problematic. We researched further and decided that NodeJS would be a good substitute to create the system in. ØOA agreed that NodeJS was an acceptable platform at which point we started educating ourselves on the technology.

4.4.3.1 Setting up the framework

The group chose the MEAN-stack as the NodeJS framework because it interacts well with client-side JavaScript, and we wanted a basis for delegating different tasks amongst the group. Because AngularJS, (the 'A' of MEAN), is old and no longer actively developed (Darwin, 2018) the group replaced it with the new AngularCLI which always runs the latest angular version (v5 as of writing this).

Using Node Package Manager(NPM) developers can benefit from one of the world's largest collection of libraries and resources that can be implemented in Node.js applications. There are over 1600 developers contributing to NodeJS on GitHub and millions of applications are built using NodeJS (About NodeJS, n.d.).

4.4.3.2 Future implementations

The application built in this project can simply be upgraded by adding several functionalities. Node.js have an efficient structure and is fast when working with JSON files and non-CPU consuming applications.
The application is made by taking into consideration new functionalities that can be added in the future like messaging and can be further scaled if there is a need to serve different regions outside of Østfold. The use of MongoDB integrated with NodeJS simplifies this process because there is no need for altering the database for storing a different form of data.

4.4.3.3 The community

The continuous growth of the NodeJS and JavaScript communities, guaranties that developers working with the technology benefit from its resources and support. There are several articles, blogs, active developers on stack overflow and similar services in addition to the official documentation that can be accessed on the web.

After the completion of this project, ØOA should not struggle to find developers that can be hired to further develop the application when there is a need for update, maintenance or new functionalities.

4.4.4 Selecting the database

Because MongoDB is an integral part of the MEAN-stack, and because of its shared asynchronous nature with NodeJS, the team decided there was no reason to switch to a different database.

Chapter 5 - Design

This section elaborates the UI design, use cases and design decisions that was taken. Each feature of the design went through an iterative process and was reviewed by the team and ØOA.

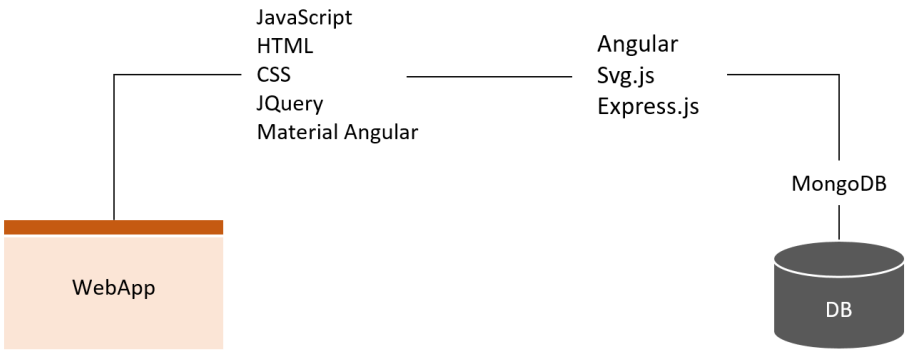


Figure 09: The general structure of the system.

5.1 Domain description

The application serves as platform for teachers, ski clubs and ØOA for sharing and exchange orienteering maps, activities and other related information. ØOA is responsible for administering the application. As opposed to normal users, the admin has access to all functionalities of the application (Add/delete/edit users).

Details stated in the table below illustrates different user groups' access to features of the application. The user groups are divided in three, these are ØOA, Ski clubs, and the schools.

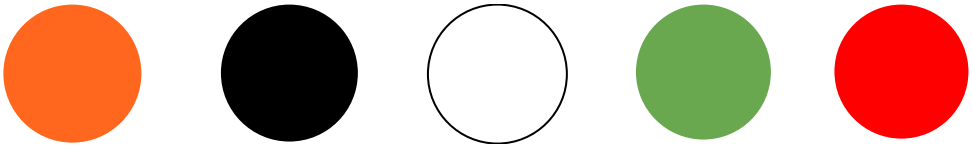
Access	Admin	User	Guest
Upload maps	*	*	
Download maps	*	*	*
Print maps	*	*	*
Delete users	*		
Add new users	*		
Register		*	*
Draw points	*	*	*
Write and publish articles	*		
Save map courses	*	*	*

Table 2: Access permissions by group.

5.2 Interface

5.2.1 Design Requirements

User Interface color palette



Background and decorations	Text and Background	Text and Backgrounds	Positive feedback	Warning
Orange #FF671F	Black #000000	White #FFFFFF	Green #6AA84f	Red #F00

Table 3: Color palette

Icons

Used as	Icon	Description
Login		Appears when users are not signed in
Logout		User logout access from drawer
Map		A page were users access orienteering maps
Home		Main page of the application
News		Hyperlink that sends users to news page

Table 04: Required icons

5.2.2 Use case

Use Case 1 - Admin upload a map:

Goal in Context: Admin/sport clubs upload a map

Scope: Upload Page

Level: Sub function

Preconditions: admin has a map that are created and downloaded from third party applications.

Success End Condition: The map will be stored in database and it will be available to users.

Failed End Condition: The map failed to be stored in database and the user will be prompted with error message.

Trigger: Admin wish to upload new map.

MAIN SUCCESS SCENARIO

1. User clicked on upload button from the hamburger menu or header
2. User will be able to select the file from windows explorer or finder in MAC.
3. User click UPLOAD FILE

4. Admin/clubs will be notified if file is uploaded
5. The map will be available for users

RELATED INFORMATION

Priority: High

Performance Target: -

Frequency: -

Channel to primary actor: Web application

Secondary Actors: windows explorer or finder in IOS operating system

Channel to Secondary Actors: default file explorer of the operating system

EXTENSIONS

2a. Failure of operating systems

3a. Uploading may fail because of network failure.

3b. Uploading may fail if admin try to upload unsupported formats.

OPEN ISSUES

User should be able to get a confirmation of the uploaded file. The system should be able to check if the file is uploaded and confirm the event. Network problem/ error can occur between while uploading.

SCHEDULE

Due Date: Unspecified

Use Case 2 - Edit Map:

Goal in Context: Edit or add an orienteering course on the map

Scope: Edit map page

Level: Main function

Preconditions: A map uploaded in the system available in the user's location

Success End Condition: Save/Print the map

Failed End Condition: Map or orienteering course unable to be saved in the database, or unable to be printed.

Trigger: User wants an orienteering track of a location.

MAIN SUCCESS SCENARIO

1. The user wants to draw an orienteering track on a map.
2. User selects the desired map

3. User draws the orienteering track on the map using the functionality present
4. User is finished and wants to print the map
5. User selects “Print Map”
6. Map is saved the system, and printed for the user
7. Map is available for other users to edit

EXTENSIONS

2a. The map is unavailable to the user

3a. User edits an already existing orienteering track

4.a User didn’t finish drawing the orienteering track

6a. The map failed to save or print

RELATED INFORMATION

Priority: High

Performance Target: -

Frequency: -

Channel to primary actor: Web application

Secondary Actors: Web browser’s print functionality

Channel to Secondary Actors: The default window.print() function in JavaScript

OPEN ISSUES

The map a user wants might not be uploaded in the system.

The map a user wants might not be available in their location.

SCHEDULE

Due Date: Unset date

Use Case 3 - Admin Post article:

Goal in Context: Admin post edited articles

Scope: CMS page for admin

Level: Sub function

Preconditions: Articles should be edited using the rich text editor.

Success End Condition: The article will be stored in database and it will be available to all users.

Failed End Condition: The article failed to be stored in database and the user will be prompted with error message.

Trigger: Admin publish edited rich text -article.

MAIN SUCCESS SCENARIO

1. User click on publish article.
2. System gather all texts with their style features (Bold, italic, underline, new line etc..).
3. The article text will be stored into database.
4. User will be notified that the article is published successfully.
5. The article will be available to all users.

EXTENSIONS

1a. Storing text may fail because of network failure.

3b. Uploading may fail if admin try to insert symbols that are not acceptable by MongoDB.

RELATED INFORMATION

Priority: High

Performance Target: -

Frequency: -

Channel to primary actor: -

Secondary Actors: Server-side functions for saving article text.

Channel to Secondary Actors: Functions exporting data from UI to backend.

OPEN ISSUES

Functionality of rich text editor will only be few and for that reason formatting of text, mathematical symbols and accessibility shortcuts using keyboard will be affected.

SCHEDULE

Due Date: Unspecified

Use Case 4 - Admin Edit article:

Goal in Context: Edit an article

Scope: Edit article page

Level: Main function

Preconditions: An article posted and saved in the system. The actor is an administrator.

Success End Condition: Article is edited and updated in the system

Failed End Condition: Article fails to save properly

Trigger: User wants to edit an article

MAIN SUCCESS SCENARIO

1. The user wants to edit an article
2. User clicks “Edit” on the desired article
3. User edits the contents of the article
4. User saves the article
5. Article is updated in the database

EXTENSIONS

4a. User didn’t finish editing the article

5a Article fails to save in the database

RELATED INFORMATION

Priority: High

Performance Target: -

Frequency: -

Channel to primary actor: Web application

OPEN ISSUES

SCHEDULE

Due Date: Unset date

Use Case 5 - User find map

Goal in Context: User search for map

Scope: Internal map search

Level: Sub function

Preconditions: Maps from database should be available to be indexed

Success End Condition: User will be able to view all maps based on the search key user inserted.

Failed End Condition: No search results available or user gets irrelevant results

Trigger: User search maps

MAIN SUCCESS SCENARIO

1. User search map using keyword
2. System checks maps on keyboard key down
3. System display result

4. User will be able to view maps that contains the search key

EXTENSIONS

1a. No result with the inserted keyword

RELATED INFORMATION

Priority: High

Performance Target: -

Frequency: For each search

Channel to primary actor:

OPEN ISSUES

SCHEDULE

Due Date: Unset date

5.2.3 UI conceptual design

The conceptual design table explains the attributes required for important features of the application. The conceptual design was used before designing the UI as a guideline to identify required functionalities and their attributes.

Activity	Attributes	Operation
Login	user name, email, login button, register button, auth	Allows users to login
Profile	username and email, uploaded maps	Display user profile
Edit map	draw control points, select, print, download	Allow users to draw control points
Upload	Upload button, form to map title, description and location	Allows users to upload map
Content Editor	Rich text editor, save, select type as activity or news	Allows users to post content
Register	User name, email, password, validation, login button	Allows new users to register
Dashboard	Grid view, title, description, edit redirect, search	Display available maps
Articles	activity, news, grid view, read more	Display articles grouped as activity suggestion or news from ØOA

Table 05: Conceptual design.

5.2.4 Information Architecture

The application we build is a single page application, the pages that are linked to each other and their structure is illustrated in figure 10

Navigation routes

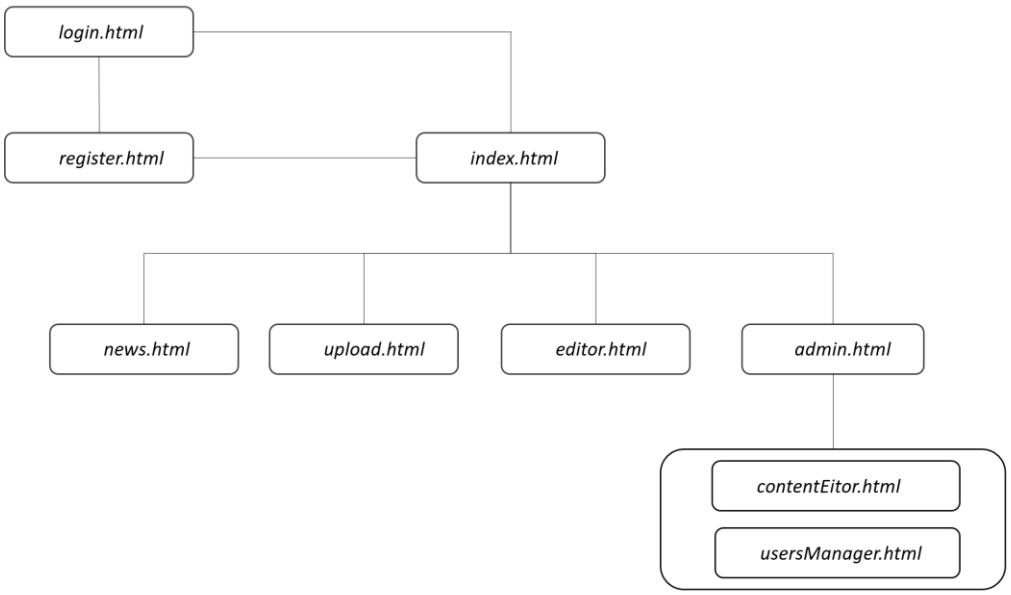


Figure 10: Structure of the navigational routes.

5.2.5 Design iterations

5.2.5.1 Iteration

When comparing the initial design and features with the final design, there is a noticeable difference between them.

The UI designs were a guide as to how the final results should look like and the minimum functional requirements. The design and features were redesigned by iterating and improving the functionality.

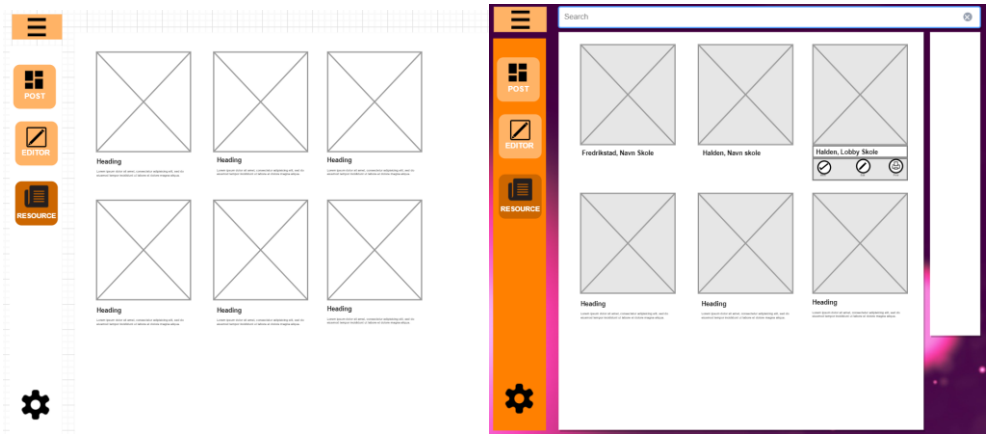


Figure 11: First iteration of the home page.

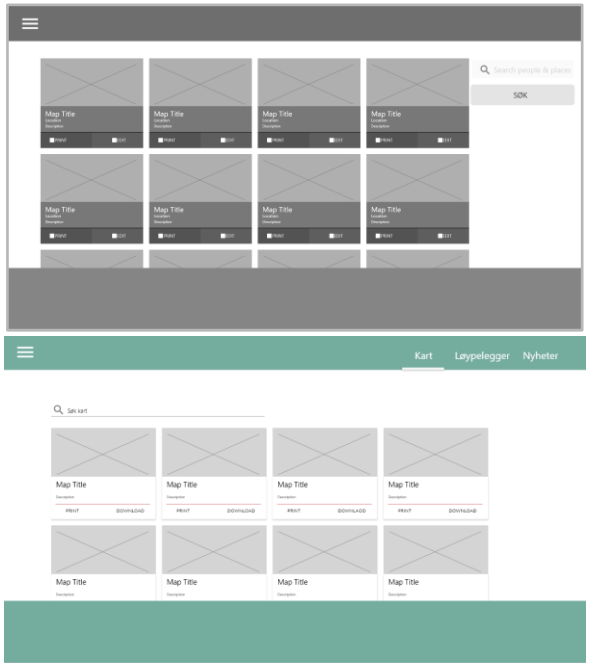


Figure 12: Home page that display maps 2 and 3

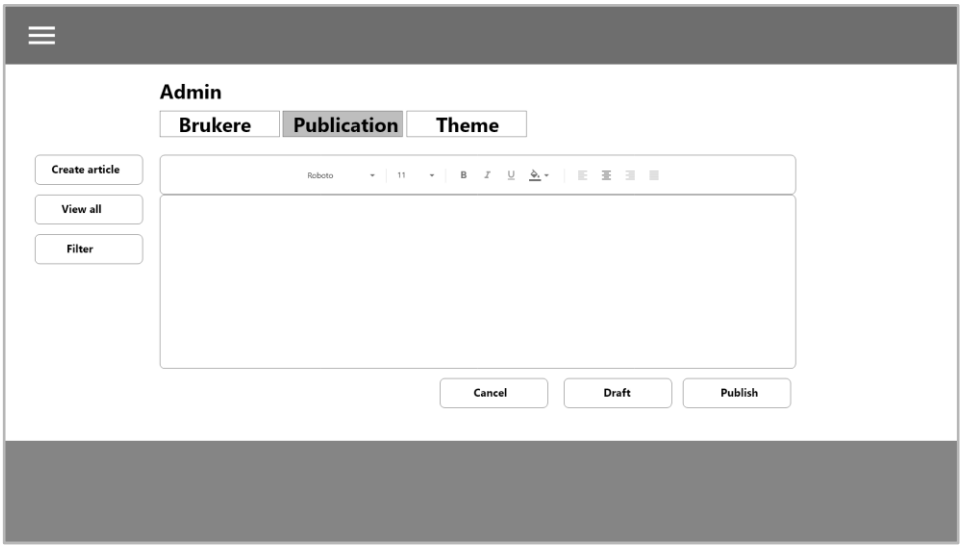


Figure 13: Text-editor/content-editor.

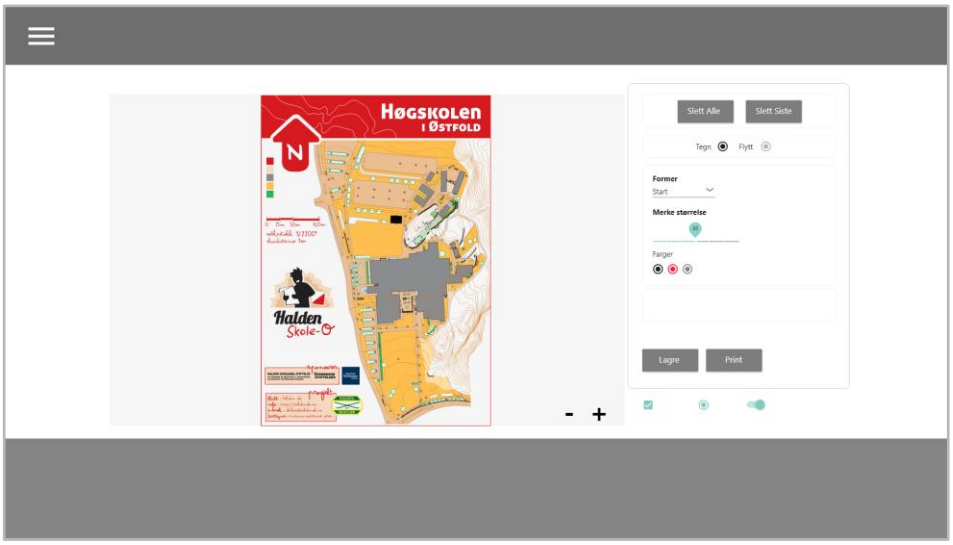


Figure 14: Edit-map page.

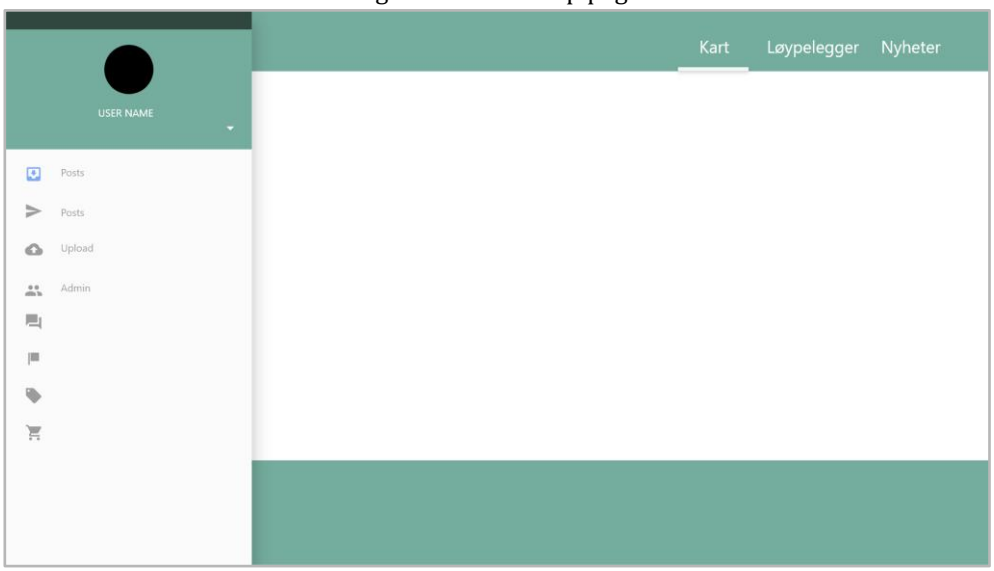


Figure 15: app component with drawer

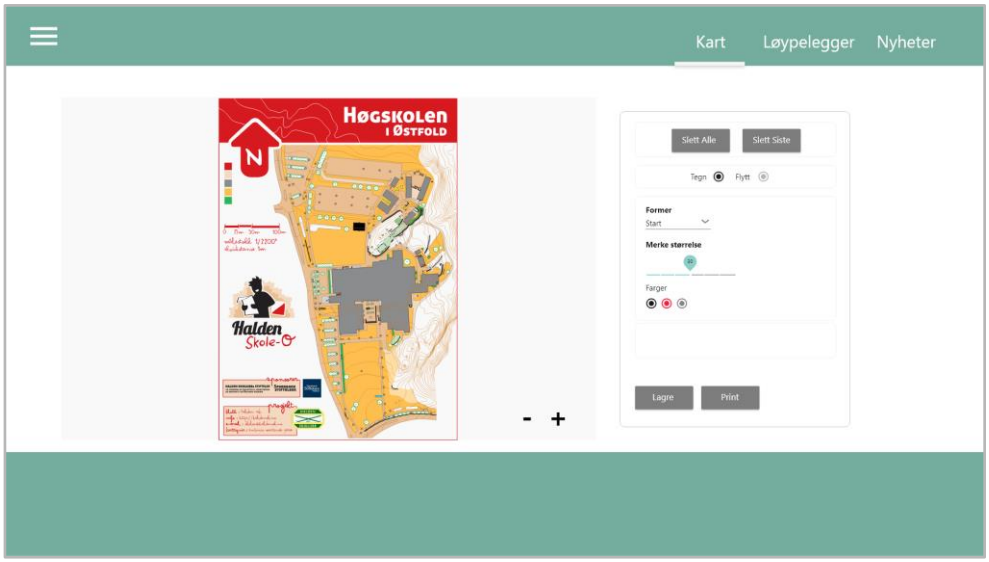


Figure 16: Edit-map page 2

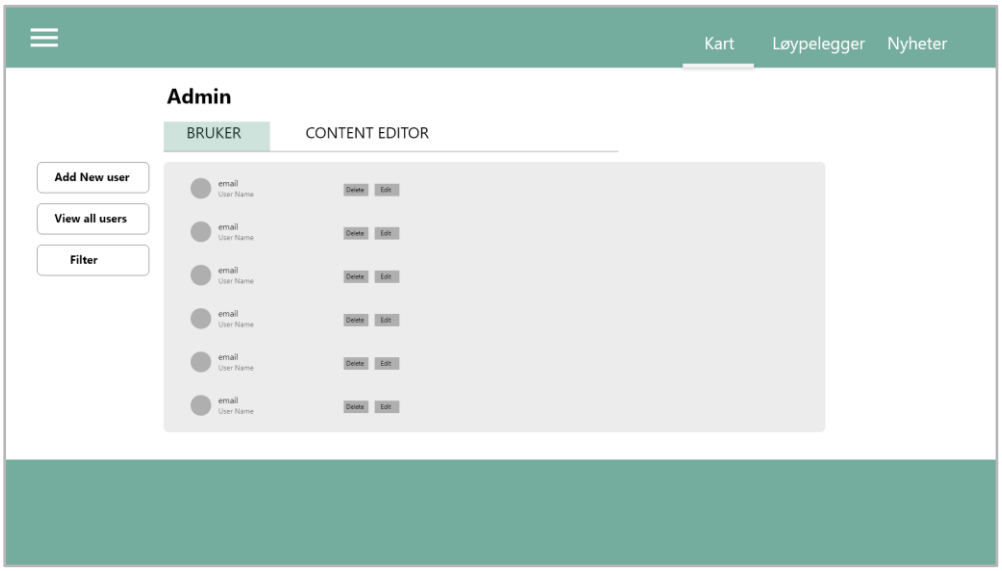


Figure 17: Users list with edit and delete.

Mobile UI Designs

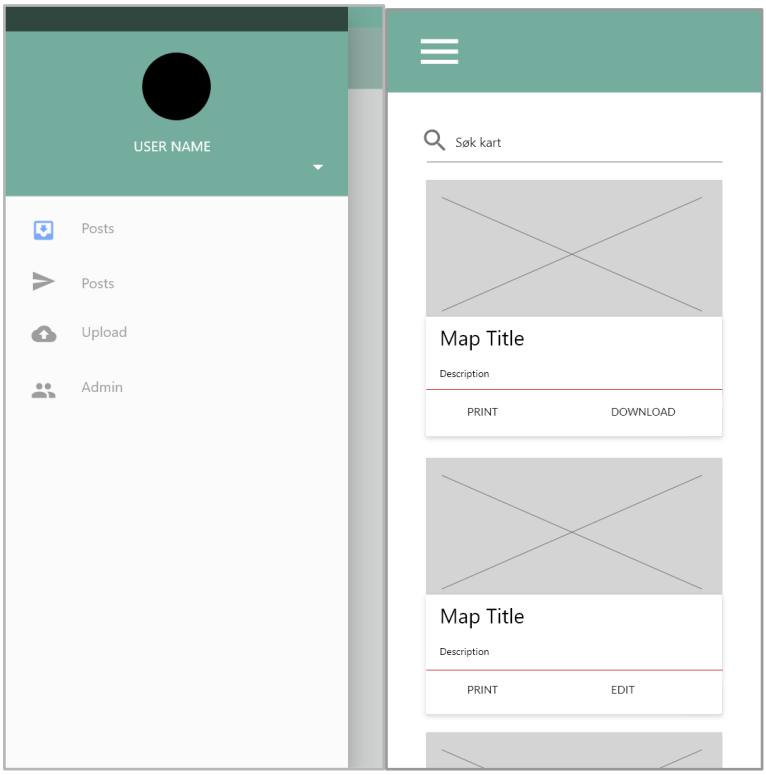


Figure 18: Mobile friendly design illustration.

5.3 Functional Requirements

The functional requirement is essential criterion and guideline for the technical part of the development, it describe expected features and behaviors of the system that are essential to deliver a complete product. The application is expected to comply to the specification, a product that didn't meet the requirement should be documented accordingly.

Application

Requirement	Requirement Statement	Must/Want
001	The application shall contain a list of maps that users can choose from.	MUST
002	The application shall have a search bar that the users can use to search the maps.	MUST
003	The application shall include a map-editor where the users can edit, save and print the maps.	MUST
004	The application shall contain a news publishing editor where the users can publish news and paragraphs.	WANT

005	The application shall have a register and a login function where the users can login as admin and teacher.	MUST
006	The application shall contain a hyperlink to another website with additional resources about orienteering.	WANT

Table 06: Features priority

001: The application system shall;
001.1: The application shall have list of maps
001.2: The application shall give the opportunity to the users to pick a map from the list.
001.3: The application shall make the user enter the map when the user clicks on the desired map.

002: The application system shall;
002.1: The application shall have search bar.
002.2: The application shall give the user the opportunity to enter an input in the search bar.
002.3: The application shall search for the users desired input and output the desired input.

003: The application system shall;
003.1: The application shall have a map-editor.
003.2: The application shall give the user the opportunity to edit the map.
003,3: The application shall have the opportunity to display all the control points on the map.
003.4: The application shall give the user the opportunity to save the map.
003.5: The application shall give the user the opportunity to print the map.

004: The application system shall;
004.1: The application shall have a text-editor.
004.2: The Text-editor shall have the opportunity to create html text.
004.3: The text-editor shall provide users with text-editing tools.
004.4: The text-editor shall give the users to publish/save paragraph or news in a database system.

005: The application system shall;
005.1: The application shall have a register function with forms.
005.2: The application shall have a Login function with forms.
005.3: The application shall give the opportunity to the users to register and login as a admin.
005.4: The application shall give the opportunity to the users to register and login as a teacher.

Chapter 6 - Documentation

This chapter contains relevant details of the code, modules, libraries, functionalities and the database implementation.

Relevant links related to the application:

GitHub:
https://github.com/Vice93/app

UI Mockup
https://xd.adobe.com/view/99305f77-1eaa-4a29-577e-09f7ef1cb464-ef01/
Password: Gruppe34

The application:
http://158.39.162.173:8080/

Login credentials for admin rights:
Email: sensor@mail.no
Password: SensorG34

Login credentials for normal user:
Email: user@mail.no
Password: SensorG34

6.1 Application Structure

Figure 19 is a visual representation of the application structure. The application is divided in two parts, client side and back end.

The client side is the what the user interacts with and contains user interface, services, helper class dependencies and test configuration.

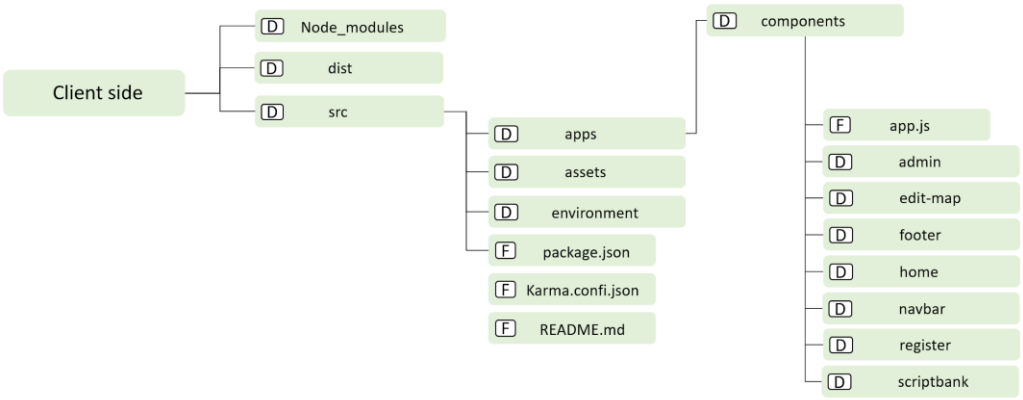


Figure 19: The client-side structure.

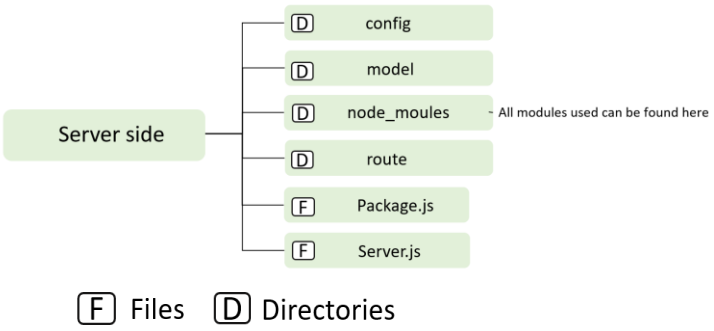


Figure 20: The server-side structure.

6.2 Server-Side Documentation

The application runs on Express server version 4.16.3. Express.JS makes the implementation of the core framework fast and seamless.

Server.js contains the server-side code responsible for running the application. The server-side dependencies including Express.js are called here. Dependencies are external modules that are utilized for utilizing helper classes and services available in npm.js. All server-side dependencies we used can be found in package.js file.

```
"dependencies": {
 "bcrypt-nodejs": "0.0.3",
 "body-parser": "^1.18.2",
 "cors": "^2.8.4",
 "express": "^4.16.3",
 "jsonwebtoken": "^8.2.1",
 "mongoose": "^5.0.17",
 "multer": "^1.3.0"
}
```

Figure 21: Dependencies on the server side

6.2.1 Modules (Appendix A1: Code block 2)

All the modules included in the server.js file can be found in "Appendix A1: Code block 2". These are the set of functions we have required in the application.

6.2.2 Server port and HTTP request (Appendix A1: Code block 3)

The app variable is initialized with express. We have a route to the homepage which sends the index.html from the dist folder and a listen function which takes the port 8080 and starts up the server.

6.2.3 Middleware (Appendix A1: Code block 10)

The middleware is specified in server.js file. In addition, there is a route middleware inside authentication.js this function takes the token and authenticate users before redirecting to selected routes. This middleware is causing problem when users redirect page to page. The bug is not fixed yet.

6.3 Database documentation

6.3.1 Encryption (Appendix A1: Code block 4)

The data sent from the application to the database are encrypted using Node.js Crypto module. RandomBytes (256) is the algorithm that generates random cryptographic data and the argument 256 indicates the number of bytes to be generated. This algorithm accepts a call back (crypto.randomBytes(size[, callback])) and it can be applied to return an error.

6.3.2 Hash (Appendix A1: Code block 5)

The createHmac() function generates Hmac object from the given string and the hash simplifies or reduces the returned value from randomBytes function and creates a set of numbers from it.

6.3.3 Database (Appendix A1: Code block 6)

The database is generated from the config.js file and the name of the database is specified as "mean-stack-test" which should be changed before deploying. A suggestion for appropriate naming can be for instance the application name. The database schema can be seen in the appendix described above.

6.3.1 Setting up and running MongoDB locally

- Download and install MongoDB.
- Unless specified, a MongoDB file is by default stored in C://Program Files/mongodb directory.
- MongoDb runs on port 27017 by default.
- Navigate to ../bin in the MongoDB directory file, and run the command "mongod" to start the database.

6.3.2 Accessing the database

There are a variety of applications available to access a MongoDB database. We selected two of the many options available, installing MongoDB compass or manually in Command-prompt. We tried both approaches and the easiest alternative was to use MongoDB compass *"MongoDB compass is a graphical user interface that can help you explore data, create databases, create collections and run queries."* (mongodb.com, n.d)

Collection Name ▲	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size
articles	34	2.8 KB	95.9 KB	1	36.0 KB
maps	3	166.3 B	499.0 B	1	36.0 KB
users	28	155.1 B	4.2 KB	3	108.0 KB

Figure 22: Database collections for article, maps and users.

6.3.3 Running the application on an external server

Running the application on a server is straightforward as long as you have root access to the server. Otherwise consult your server administrator to follow the following steps.
These steps assume you are using a Linux based server with access to the Linux Package Manager (apt). First of all, the server needs Node installed to be able to run node applications. This can be done simply by following Node's documentation to run the command `apt-get install nodejs`. However, a better solution in our opinion is to install Node Version Manager (NVM). NVM is a simple solution to the problem of managing multiple NodeJS installations on a server as well as easily updating the current Node version.

Insert these commands into your terminal window:
1 - `apt-get install git`
This installs git on your server and adds it to the environment path

2 - `apt-get install https://github.com/creationix/nvm`
This installs Node version manager on the server allowing you to specify which version you want to run globally, as well as on each specific project.

3 - `nvm use 8.9.4`
This tells NVM to use Node version 8.9.4 globally on the server. You can specify any other version here if you want to.
3.1 - If you want to use a different version on a specific project, create a .nvmrc file in the project root folder (in our case, "app/") and specify which version you want to use:

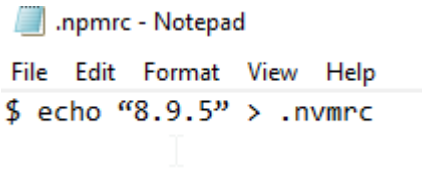


Figure 23: Create .nvmrc file

Your server can now run node applications. Next you need to clone a project from a git repo. Make sure you navigate to your “root” folder (or where you otherwise want your app) before you run this command:

5 - git clone https://github.com/Vice93/app
Here you can specify any git repo you want to clone and install on the server. Next you may want to test the application and see if it runs:

6 - node server.js
If your startup script is called server.js, the above command will start the application. Change this to whatever your startup script is called (e.g. node app.js).
If your git repository doesn’t have node modules uploaded, then you need to install them on the server first:
6.1 - npm install
Make sure to navigate to client and run the same command if you have an AngularCLI app.
Your application should now be working and viewable on the url:port you specified in your server.js file.

Next, we need to install Mongodb on the server.
7 - apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
This retrieves mongo dB’s public key used to ensure package consistency and authenticity.

8 - echo "deb [arch=amd64] https://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.6 multiverse" | tee /etc/apt/sources.list.d/mongodb-org-3.6.list
This creates the Mongodb list file specifying various information used for the Mongodb installation.

9 - apt-get update
This will reload the local package database.

10 - apt-get install -y mongodb-org
This will install the latest stable version of Mongodb on the server. If this command doesn’t work, run the following instead:
10.1 - apt-get install -y mongodb

Now you're ready to start Mongodb on the server, you can do that with the following command:
11 - `service mongod start`
This can also be stopped with:
11.1 - `service mongod stop`
Or restarted with:
11.2 - `service mongod restart`
NB: If `mongod` is not recognized, type `mongodb` instead.

Mongodb is now installed and you can specify what localhost address and port you want it to listen to:
12 - `mongo --host 127.0.0.1:27017`

6.4 Client-Side Documentation

6.4.1 Front-End components

This section explains the front-end components. The code blocks discussed are included in appendix A2 - Client-side code.

6.4.1.1 Edit-map-component

The entire map drawing function is written in JavaScript/jQuery and is loaded client side. We don't take full advantage of an AngularCLI application this way, so the intent was to refactor it into the typescript component of edit-map. However, because of time constraints we kept the original structure. The most relevant parts of this component are explained below.

6.4.1.1.1 editMapFunction.js

Create SVG element event (Appendix A2: Code block 1)
If every parameter has a value, then this event will create an SVG element based on those values. We check for the type the user wants to create and push that object into an array.

The SVG objects (Appendix A2: Code block 2)
To make the process of remembering an SVG element's information easier, we assign the attributes of the element to an object. This way we can pass the objects directly into the database, or vice versa, we can retrieve them from the database and pass them directly into the data array. This idea was further reinforced by our selection of MongoDB as the database.

Mousedown/Touchstart event (Appendix A2: Code block 3)
This function gets invoked when the user clicks the map. Clicking set to true enables the move event. We use elementFromPoint which is a handy JavaScript function that returns a html element from mouse position.

DataInPosition (Appendix A2: Code block 4)
When the user clicks the map (and don't release) we get the currently clicked SVG element from the data array. This function simply returns the object whose id equals the clicked one.

Mousemove/Touchmove event (Appendix A2: Code block 5)
After clicking and holding down, the user can now move the object around. We simply set the object's position to equal the current mouse/touch position. Because we have line objects as well with 4 positions, we need to check how it's currently positioned to retain its angle and length. If not, simply update its position.

CreateCircle and temporary objects (Appendix A2: Code block 6)
When moving an SVG object, we create a temporary SVG element to let the user see where the real object was previously, and where they are currently moving it.
This is the function that creates a circle (or a post) on the map. If "isTemp" is true, the SVG element created is temporary and will be deleted when the user lets go of the mouse. This allows us to use a single function for both types simply by passing a Boolean.

Mouseup/Touchend event (Appendix A2: Code block 7)
When the user lets go of the mouse, we delete the temporary object we created, and update the real object's position. Clicking set to false disables the move event from functioning.

SetTransformScale (Appendix A2: Code block 8)
Because we wanted to enable the application for mobile/tablets as well, we required a way to retain the SVG elements' positions when the screen size changed. For example, if a user initially created their orienteering course on desktop, but later wanted to view it on mobile, the positions would be way off.

This function currently sets transformScale to equal the width differences between the max width and the current width. transformScale is multiplied with an elements position when its created or moved.

GetOffSet & GetPage(Client)XY (Appendix A2: Code block 9-11)
Because we use various CSS rules to style the map container, we need to calculate the offset. And because we use those offset values multiple times, we created a simple function to return them whenever needed.

In addition to offset values, we also required page x & y values multiple times, so we made it into a function, and in a few cases, we need to get client x & y values instead.

DeleteSvgFromPoint (Appendix A2: Code block 12)
The delete method behaves much like the DataInPosition function. This one, however, deletes whatever element's id matches the clicked one by removing it from the data array.

Update (Appendix A2: Code block 13)
Whenever this function is invoked, we remove every SVG element from the group container, and redraw all the objects found in the data array. Data type 1 and 4 is line and text respectively, and when those are drawn we don't want to add automatic numbering to it (if the user has selected this option).

Automatic numbering (Appendix A2: Code block 14)
When an actual SVG element (not temporary) is created, we add a number next to it indicating what order the elements were created in. This is an optional feature for the user. We would like to add the option to slide the number around the axis of the SVG element, instead of setting its position fixed relative to the element.

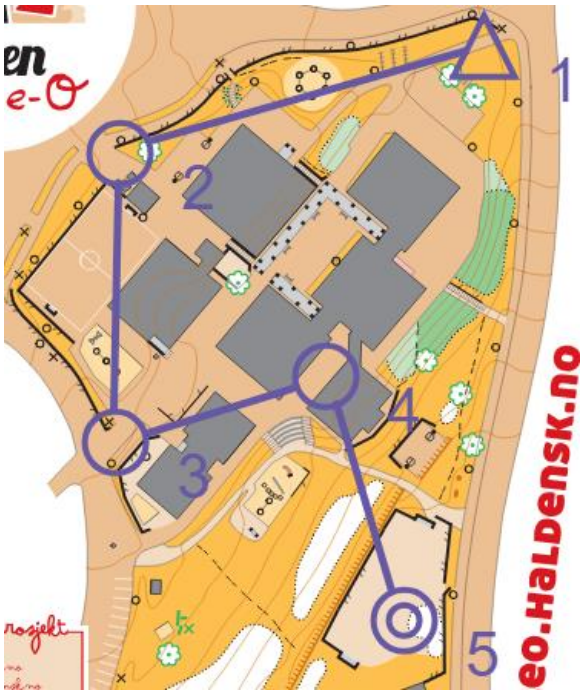


Figure 24: Automatic numbering example.

Undo/Redo/DeleteAll (Appendix A2: Code block 15)
This is a very simple "undo/redo" solution. It does not do everything it "should", but the current implementation remembers the last deleted object if the user deletes it, and lets the user redraw it if they regret deleting it. It does not remember the old location of the last moved object, which is something else we would've liked to include.

6.4.1.1.2 edit-map-component.ts

The angular part of this component mainly has 3 functions: Print the map, save a course and list all available courses from the database.

Print (Appendix A2: Code block 16)
The print function simply creates a new popup window in the browser, and writes a new html page using only the SVG element container, and custom CSS rules to remove margin, padding etc.

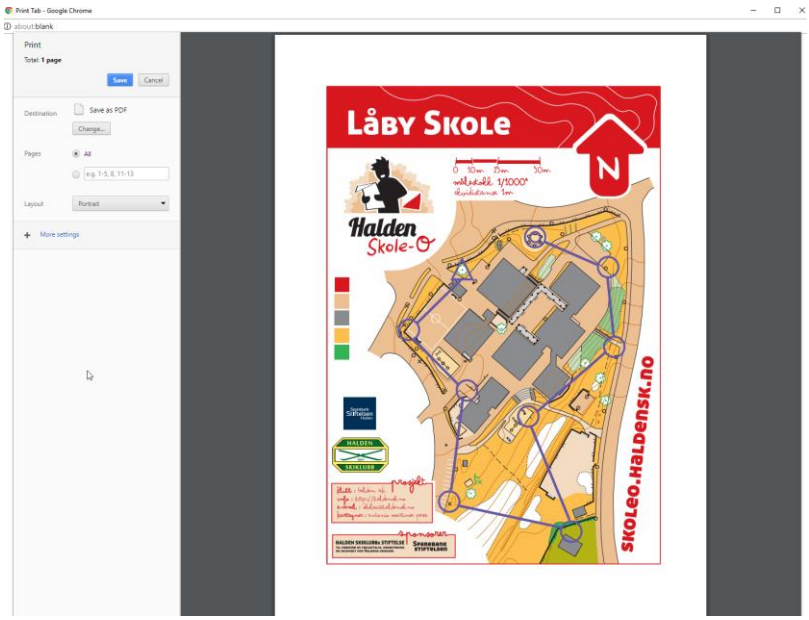


Figure 25: Print of map using chrome.

module.js export functions (Appendix A2: Code block 17)
Because angular can't use un-exported functions from JavaScript, we need to declare the exports. However since this can only be done in ES6 (where script type="module"), we need to create a middleware that understands both plain JavaScript and typescript. We then create a module.js file that exists in the same DOM as the JavaScript file, and thus understands the global variables and functions written. With these functions declared we can access them in the component.ts files to get the data array containing the SVG objects.

Save Course (Appendix A2: Code block 18)
Then in typescript we can simply import the function getData() and voila, we have the currently drawn course in angular, and can save it in the database. The user can then choose to save this course in the database along with a title and description of the course.

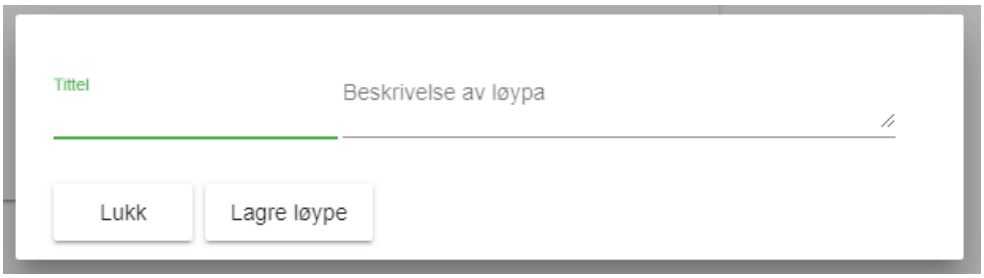


Figure 26: Save a course dialog.

Load course (Appendix A2: Code block 19)
To select a course, we retrieve all courses from the database that exists on the currently shown map. We then push it into two different arrays: One with all available verified course, and one with the users own personal courses. To find a user's personal courses, they must be logged in, or else it will simply be empty.

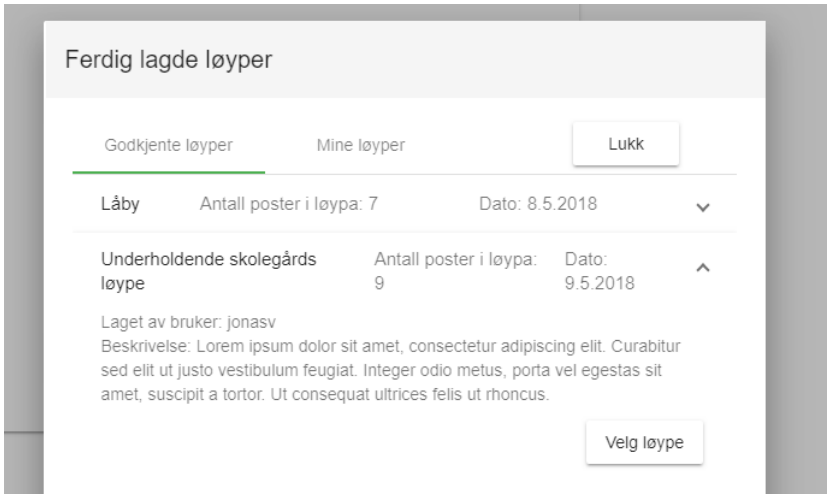


Figure 27: Load a course dialog.

Return the course (Appendix A2: Code block 20)
When the user selects a course, we return that single object back to the component and then send it back to the editMapFunction.js script. Since the array is populated, it is now fully editable using the drawing tools.

Figure 28: Course loaded after selecting it.

6.4.1.2 Home-component

The home page is where you search for orienteering maps by selecting a location. For performance reasons, we are showing the .png version of the SVG maps as thumbnails on this page. Converting the maps to .png is further detailed in the back-end section of this documentation chapter.

Load Maps (Appendix A2: Code block 21)
When the user selects a location, all locations with that name are retrieved from the database. All results are then displayed in a list, and the user can search to filter the results.

Figure 29: Search and location select in home page.

Filter pipe (Appendix A2: Code block 22)
To filter the result, we export a filter pipe, and tell the home component to use this filter in the results. This filter returns all elements that match the search input in the description, or the area. To further improve this, tags should be implemented and passed when saving a map and included in this filter.

Using the filter pipe in html (Appendix A2: Code block 23)
To actually filter the results, it needs to be included in the *ngFor in the html.

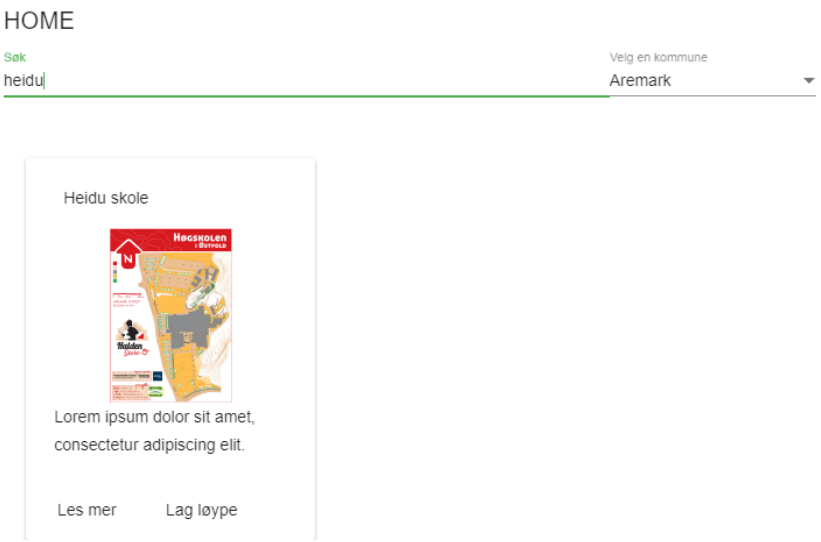


Figure 30: Filtered results based on search input.

Read more about map (Appendix A2: Code block 24)
Further we have two buttons on each result: Read more and Create course. Read more simply opens a mat-dialog with the clicked element and displays more information about the map (e.g. description). First, we open the mat-diag and pass along the clicked element:

Opening a mat-dialog and passing the data (Appendix A2: Code block 25)
When opening the mat-dialog we need to inject the data and display it using data bindings (data bindings are simply a way to reference a typescript variable in html). Since we already have the objects in home components, we simply pass along the object in position 'i' (the clicked elements position) and read the values we choose to from the object.

Navigating to edit-map-component (Appendix A2: Code block 26)
The second button on the results, "Lag løype", sends the user to the edit-map-component along with the selected orienteering map. The map reference is then set directly to the src attribute in the SVG holder.

6.4.1.3 Upload-map-component

This component is used for uploading orienteering maps made in other softwares, like OCAD, from the user's disk. When first navigating to the page, you are prompted to select a file.

Selecting map file for upload (Appendix A2: Code block 27)
When the user opens the select file dialog, we apply a filter to only look for .svg files. However, JavaScript can't further control the file selection tool, so if the user selects "all files" and choses a non-SVG file, we alert the user that this file is unsupported and asks them to select another. If the file is valid, they may proceed to upload it by hitting the "upload" button.

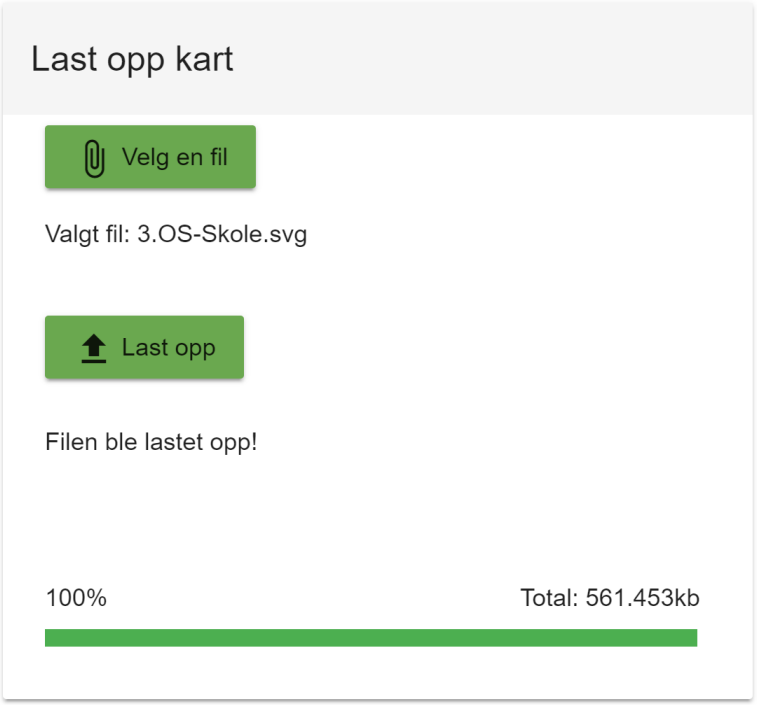


Figure 31: Upload map feature with uploaded map info.

Upload the map file (Appendix A2: Code block 28)
This function does 3 things: Check if file was already uploaded, check if user hasn't selected a file yet, and send the file to the server for storage. We also track the event of this particular request to display a progress bar, and to return the name of the stored file.

Add information to the map (Appendix A2: Code block 29)
After the map is uploaded and we have the name of the file, the user can add information to the map. Since the location must be queryable in the database, we use a select box the user can choose a location from. They may also input additional information if they desire, though it's not required. Once they hit save, the map reference along with information is saved in the database.

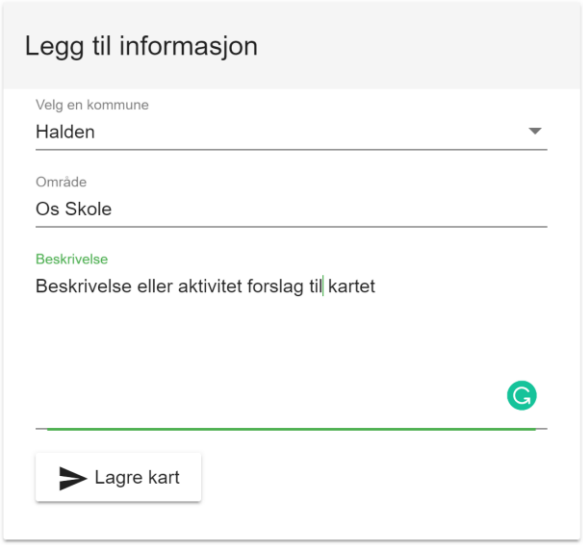


Figure 32: Information about the map and description form.

6.4.1.4 Admin-Component

The admin component is accessed by ØOA or other parties like the ski clubs. This component allows them to publish activity suggestions and contents that can contribute to enhance the schools and teachers' competence about orienteering sport. The published contents are available to users from 'Nyheter' page.

Meta data

Users can insert the title, description and select the type of the content.
The title of the content will be displayed as a title text when the content is accessed in the news page

Figure 33: Content title form from admin page text editor.

The description is a small size text with less than 50 characters that describe shortly the content. This will hint users what the content is about.

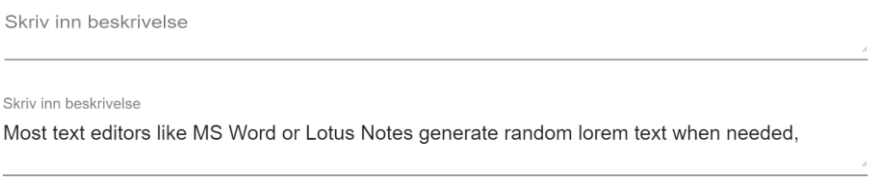


Figure 34: Content description form from admin page text editor.

Users select the content type by clicking on of the two alternatives. The news and activities. This will help readers to sort which type of content they want to read when the content is displayed inside 'Nyheter' page.

Figure 35: Select content type before publishing.

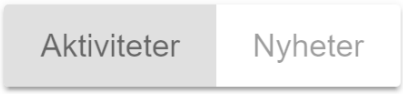


Figure 36: From the news page users can sort by content type.

The published text will be displayed as figure nr. Inside 'nyheter' page. Users can click on 'Les mer' to view details and users that published the text can delete the text if they wish to. The 'slett' button is visible for users that have the sufficient access level.

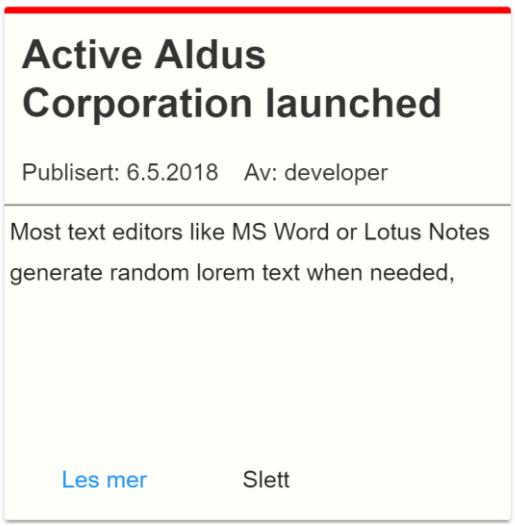


Figure 37: The content display card from news page.

Content Editor

The content editor is an API driven a rich text editor plugged in the component using Quill.js. The text editor allows users to write contents and format as desired. The formatted content will be stored accordingly and retrieved from the database with all its title, description and title.

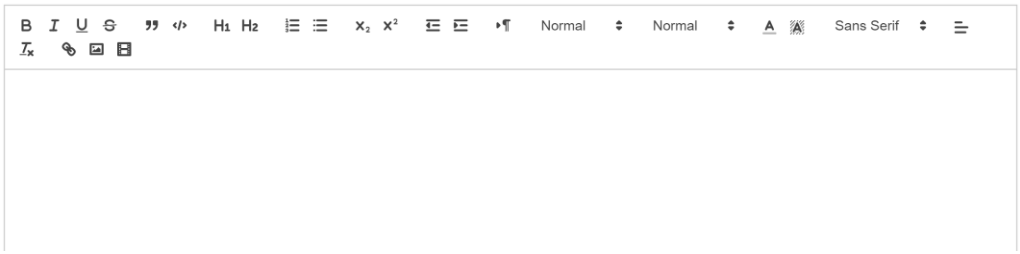


Figure 38: Text editor powered by Quill.js

Further improvement.
The text editor is not completed and there are certain features that should be added.

1. No auto store, users can lose content not stored if browser update, crush or shuts down.
2. No draft save, no options to save drafts before publishing.
3. Image and Video adding videos and image in the text editor can cause unexpected behavior. Slow system and the images and videos may not be retrieved as expected.

Solution:

1. Implement onkeydown or auto save using functions that triggers based on time duration (run function every x minute).
2. Build a function to save drafts or change the database model by adding a "draft" identifier.
3. Short solution is to add image and video ext field and upload the images and video in storage and add reference in db. But a better solution can also be implemented.

Initialization of Quill text-editor (Appendix A2: Code Block 34)
First and foremost, before initializing quill text-editor the quill scripts and the CSS-file needs to be in the html file. After that, Quill was initialized by creating a new object of quill. All the Quill functionality/attributes can be seen in the code block 34.

Publish article/activity function (Appendix A2: Code block 30)
This code takes the value from the editor and the other fields. It creates a http-post request and post the data to Mongodb.

Fetching the text-editor content from Mongodb (Appendix A2: Code block 32)
Getting data and loading it from Mongodb when the page initializes. The code goes through the data in a "let" loop and checks if the content being fetched from Mongodb is of type article or of activity.

Toggle function (Appendix A2: Code block 31)
This code makes an element of a toggle appear with article or activity information. This function executes at the time of saving the article/activity to the database. An example of the element can be seen at figure 36:

6.4.1.5 Register-Component

There are certain criteria to be fulfilled when registering new user, inputs that doesn't meet these criteria are considered invalid and will not be accepted by the system.
These criteria are:

1. User should use a proper email ex. username@email.com.
2. Username should be three or more characters
3. Password should contain 8 or more characters with one or more capital letters and a number. Example “Userpassword1”
4. Username should not be reserved.
5. User should agree to the “user agreement” by clicking the check button before registration. The agreement that users can access by clicking the toggle button (les brukervilkår), it is a temporary text filled for now but ØOA is expected to implement that before publishing the website to the public.

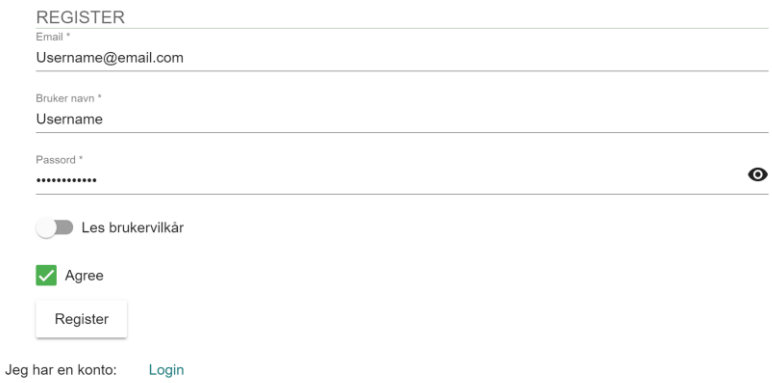


Figure 39: Register form from register component.

6.4.1.6 App-Component

App component is the engine of the application that runs all other components inside it. The app components use the “app routing module” to access the rest of the components that will be plugged into the application.

Usually it’s normal to separate headers and footers defined on as independent component. But the implementation of material design caused some unexpected result. Because of that, the team decided to apply the header and footer inside the app component.

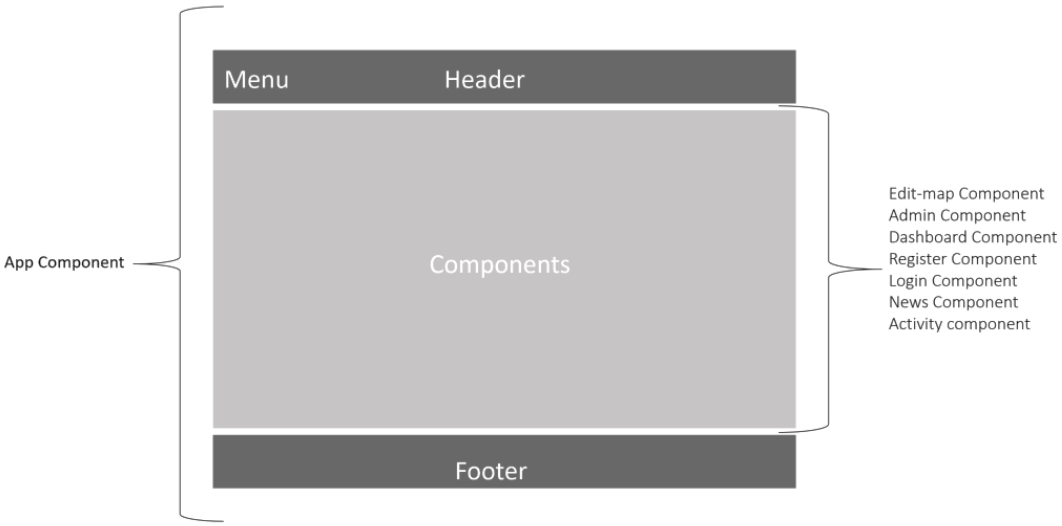


Figure 40: Implementation of components inside the app component.

The app routing module inside the app component declares a constant object array of "Routes" that assigns the paths and components that will be plugged inside the app component. Every components that will be used should be assigned inside "appRoutes" object to be available when accessed from other components and to be navigable.

```
{
  path: 'admin',
  component: AdminComponent,
  canActivate : [AuthGuard]
},
```

Figure 41: Example of component and path declaration

The app module that are declare inside "app.module.ts" are all the required modules and components using the "declarations" object. The "import" object key introduces all dependency modules required inside the application. And the "providers" key accepts the imported services.

6.5 Services

Services are classes that are responsible for managing a communication between the back end and the client-side data connection. They help writing a clean code by reducing the works of the components and separating responsibilities between data receivers (components) and data sender's (database) by sitting between these two.

The service in the application can be found inside "services" folder and they are only triggered by calls from components, taking care of API calls.

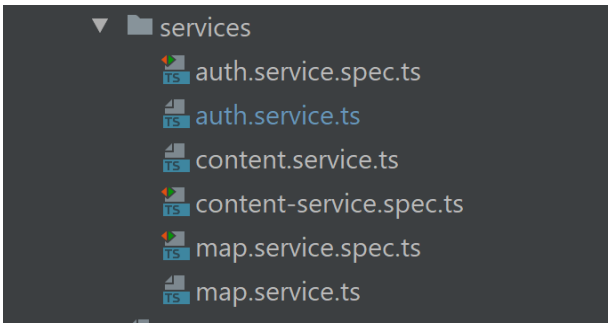


Figure 42: Services for authentication, content and map.

All services in angular imports “Injectable” and uses the “@Injectable” decorator. This will make the class be a participant in the dependency injection system.

“Dependency Injection (DI) is a way to create objects that depend upon other objects. A Dependency Injection system supplies the dependent objects (called the dependencies) when it creates an instance of an object.” (Angular - Services, n.d.).

Authentication Service (Appendix A3: Code block 1)
The authentication service that can be found in “auth.services.ts” file serves as a responsible class for managing the listed functionalities under.

- Registering users.
- Login.
- Logout.
- Generating token.
- Verifying token.
- Confirming token expire.
- Generate header options that will be send to the user browser.

Content Service
The content service was only used to send a get request of articles.
The getContent() function is called from the component that are responsible for displaying contents.

Further Improvement
Storing data direct from the components it’s not a good practice and not advised by angular documentation. Based on that, the direct store functions from the admin component can be moved here to avoid workloads of the components and maintain consistency.

Map Service (Appendix A3: Code block 2)
As well as the content service the map service also used the map.service.ts file to perform a post request. The map information (location, title, etc..) are stored via the service. The improvement suggested to the content service applies also to the map service.

The post request inside the amp service sends the token header together with the request as “httpOptions” variable. refer (Appendix A3 code block 5).

Chapter 7 - Testing

Testing is a very crucial part of any development. It requires time, attention and strategies with responsible people assigned to perform the tests.

7.1 Test plan

To identify problems that could appear in the application, the team planned to conduct observational tests that focused on functionality, usability and accessibility of the application. To cover each of these test aspects, the team has documented the criteria necessary for each one of them. These criteria requirements can be found in 7.2.1 and 7.2.2 in this chapter. Because of time-constraints, the team did not perform the planned user-tests.

7.1.1 Functionality testing

"The system is treated as a black box whose behavior can only be determined by studying its inputs and the related outputs. Another name for this is 'functional testing', so-called because the tester is only concerned with functionality and not the implementation of the software." (Ian & Sommerville, 2011)

A black box test is a type of functional test that is performed with users that do not have a knowledge of how the application is built. Developers should not be performing this type of test themselves because they have intricate knowledge of the application at hand. The black box test should be performed after each sprint release.

"... The objective of release testing is to check that the system
meets its requirements and is good enough for external use (validation testing)."
(Ian & Sommerville, 2011)

7.1.2 Usability testing

"Usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a test, participants will try to complete typical tasks while observers watch, listen and takes notes." (Usability Testing, n.d.).

The test can be performed by teachers, sport clubs and administrators of the application. The different groups will perform different tasks depending on their access level in the application.

7.1.3 Accessibility

The Web Accessibility Guidelines 2.0 (WCAG) publishes criteria and recommendations that can be applied when building a web technology that is accessible to wider range of people regardless their physical ability.

The public and private sectors in Norway are obligated to fulfill certain requirements when designing their web technologies. The regulation for accessibility and information communication technology solutions § 1. stated the purpose of the regulation as "Forskriftens formål er å sikre universell utforming av informasjons- og kommunikasjonsteknologiske løsninger, uten at det medfører en uforholdsmessig byrde for virksomheten. Med universell utforming menes at utforming eller tilrettelegging av hovedløsningen i informasjons- og kommunikasjonsteknologi er slik at virksomhetens alminnelige funksjon kan benyttes av flest mulig." (Forskriftens formål, 2013)

Agency for Public Management and eGovernment (Difi) has a simple guideline that suggest solutions and guidelines that support attaining the requirements stated in the regulation. The agency is responsible for following up the regulations on universal design of ICT solutions, related to the Discrimination and Accessibility Act.

The application we build attempt to meet the requirements for accessibility therefore we need to test the application using the guidelines of Difi. Accessibility can be tested using automated testers, keyboard and the straw test.

7.2 Test criteria

All test criteria that have been categorized and will be ranked from important to not so important and are among all;

7.2 1 Requirement for functionality Testing

In this type of test the group will check if every aspect of functionality of the application works without checking the internal structure of the application.

7.2.1.1 Upload Map

Input:

1. SVG format orienteering map.
2. Orienteering map on different formats (JPG, PDF).

Tests:

1. Test if the user will be able to upload the map.
2. Test if the user get feedback if the upload succeeded.
3. Test if the user get feedback when upload is interrupted.
4. Test if the user notified connection fails.
5. Test if the user can get feedback when uploading unsupported format

Output:
Success or failure feedback when uploading map.

7.2.1.2. Edit the maps

Input:

1. User select drawing select drawing buttons.
2. User select resize drawer select button.
3. User select different shapes.

Tests:

1. Test if the user can draw shapes.
2. Test if the user can change shape size.
3. Test if the user can draw lines between two points.

Output:
The user will be able to draw shapes and make a line between the shapes.

7.2.1.3 Download Map

Input:

1. User attempt to download a map.

Tests:

1. Test if the user download can download map and save the map in a desired location in their local hard disk.
2. Test if the user will get a feedback when download is interrupted.

Output:
User is able to download the map.

7.2.1.4 Register

Input:

1. User enters username.
2. User enters password.
3. User enter unaccepted character.
4. User register with less than characters.
5. User enters invalid characters as a username.

Tests:

1. Test if the user can enter accepted username and password
2. Test if user get feedback when user invalid input.

Output:
User register and redirected to home page

7.2.1.5 login

input:

1. User enters username
2. User enters password
3. User enter unaccepted character.
4. User register with less than character.
5. User enters invalid character as a username.

tests:

1. Test if the user can enter accepted username and password.
2. Test if the user get feedback when user invalid input.

Output:
User is logged inn and redirected to home page

7.2.1.6 search bar

Input:

1. User enters text in the search bar

Tests:

1. Test if the search bar searches for the text that was entered.
2. Test if the search bar searches by letter and gives out a list.

Output

1. The search bar lists the desired input.

7.2.1.7 text-editor

Input:

1. User can enter a String in the editor.
2. User can edit the String in the editor.
3. User can submit the String.

Tests:

1. Test if the editor lets the user enter a String.
2. Test if the editor lets the user use the options and edit the String.
3. Test if a message appears that says that the text was published when button is clicked.

Output:

1. The text is saved in the news page of the application.

7.2.1.9 Navigation

Input:

1. User navigates from menu bar
2. User navigates from the sidebar

Test:

1. Test if the user navigates through pages

Output:

User is able to navigate to a desired page

7.2.2 Accessibility test requirement

The requirements for accessibility is based on the WCAG 2.0 best practices. The team attempt to apply the solutions that fits the requirement for the application.
The tests can be performed manually by member of the group, and automated tests can also be applied when it is applicable.

7.2.2.1 Color contrast

1. The AA color contrast ratio for 4.5:1 will be used for texts and images.
2. Large text 3:1 ratio.
3. The color contrast ratio should not be applied to decorative text and images
4. The group will use webaim.org tools for testing contrast and adjusting contrast.
5. The color ratios specified are the minimum requirement and not the absolute value.

7.2.2.2 Image and graphics

1. Images should contain “alt” attribute to describe the content of the images. Though the alt attribute is not required when the images has decorative purpose and contains no relevant information.

7.2.2.3 Fonts and font size

1. The fonts should have at least 14-18 points because it is described as large-scale-text by WCAG 2.0.
2. The fonts should be resizable.

7.2.2.4 Failure feedbacks

1. The user should be notified when something unexpected happens.
2. The user should be notified what happened.
3. The user should be suggested on how to solve the problem.
4. The feedback should be #F00 (Red) color.
5. The feedback should be placed as near as possible to the area where the problem occurred.

7.2.2.5 Forms

1. Forms should be grouped together in a same area.
2. The form elements should contain tool tips.
3. User should get appropriate feedback when invalid characters are inserted.

7.2.2.6 Links

1. The purpose of the link should be described in the text of the link for the user to distinguish this link from other links in the webpage.
2. Providing link text that describes the purpose of a link in the element.

7.2.2.1 Navigation

1. Use same design patterns to distinguish menus from other
2. Label menus to make them easier to find and understand. Make them short and use Aria-label to provide the label.
3. Use markup for indication.
4. Display the menu where you think the user expects it, for instance in the middle or to the left.
5. Add a icon for indication for users that have difficulty with colors can see what they have selected.
6. Make sure that drop-down menus can be used by keyboard and mouse.

Chapter 8 - Discussion

In this chapter the team will discuss the assignment given, the choice of methods, the application developed and how the team worked during the project period. Additionally, the team will reflect on improvements that could be done on the application, and finally future work required to make it meet the contractor's as well as the team's expectations.

8.1 The assignment

The goals and deliverables listed in chapter 1.4 reflect the contractor's requirements and expectations for the application. These objectives were revisited and modified multiple times through meetings and feedback from the contractor. Initially the assignment given wasn't more than a wish to save orienteering maps in a database. This is briefly covered in chapter 3.1 and will be further discussed in chapter 8.4.

The contractor gave the team the freedom to decide on most requirements and functionality beyond the goals listed in chapter 1.4.1. As a result, the requirements about accessibility and user interface are all requirements the team decided on themselves. Overall the assignment didn't feel challenging enough when it was first drafted, which is why the team suggested additional functionality at various meetings during the period. Ultimately the team wanted to challenge themselves and create something that could be directly implemented and maintained by OØA, as well as use this as an opportunity to learn new frameworks and libraries.

8.2 Choice of methods

As mentioned in chapter 1.4.3 and further detailed in chapter 3.3.2, the team decided to work using elements from agile development, mainly scrum. However, the team did not follow this methodology entirely, but rather took elements that worked well with a small team, such as delivering in increments using a backlog to track the progress. Overall it worked out well for the team and yielded in increased productivity when compared to earlier projects. The use of Trello was a big part of why this worked so well, as managing, distributing and tracking different tasks can be quite the work in of itself. Trello resolved these issues, such that the team could focus on more important work.

However, while the use of the aforementioned methodology yielded great results, more research in others could have been a better fit for the team and resulted in an even better product. Unfortunately, the team did not have the time required to perform this research which meant sticking to what worked well was the best course of action.

8.3 The product

The team is happy with what they've accomplished during this period. The basic functionality requested by the contractor has all been developed into a functioning web application. There are still a few elements missing to make the application usable, which will be discussed further in chapter 8.6.

The landing page
The first page you are confronted with when you visit the website, is the home component. It was made to be as simple as possible with as few elements showing as possible. This was to make it clear what the user needs to do here to find the orienteering maps. Once the user selects a location, all available maps for that location will show up in a list, allowing the user to do the following: Filter the results by searching, read more information about a map or select that map for drawing a course and printing.

Drawing a course page
When the user selects a map, they are automatically navigated to the edit-map page where they can draw a course on the desired map. This page will automatically adjust to the image type (horizontal/vertical) and display the contents in a way that fits the screen. The user can either use the tools to draw their own course, or they can select one from a list of verified courses. The verified courses can then be edited as the user sees fit. If a normal user tries to save the map, it is stored in their own personal list, and they can then reuse this later. However, personal courses are only available to the currently logged in user and are bound to a specific map. When the user is happy about a course, they can save the map and its courses as a pdf or print it on a printer.

News and activities page
When navigating to the news page all news and activities stored in the database will load into the page. The results are separated by their type, allowing the user to select which type they are interested in reading. This page is not yet feature complete and will be further discussed in chapter 8.6.

8.3.1 Administrative tools

The administrative tools of the application are all supposed to be gated by the user's access level. The access levels range from 0-3, from Super-admin to User. This basis of this functionality is implemented but requires more work. It will be further discussed in chapter 8.6.

Writing news/articles page
One of the administrative tools is the content editor page, locked behind access level 2 (Content-manager) or higher. This page lets the user write news and activities they want to publish on the site. The intent was to use an editor that saved in plain html code such that it could be directly inserted into a page to display its contents. This page is almost complete but are lacking a couple of functions which will be discussed in chapter 8.6.

Upload map page
This is the page where administrators of the system can upload an orienteering map to make it usable in the rest of the system. It features two parts: Uploading a map and writing information about the uploaded map. The user is first prompted to select a file and if its supported (.jpg or .jpeg) they can upload it. Once uploaded the user can write the desired information and store it in the database.

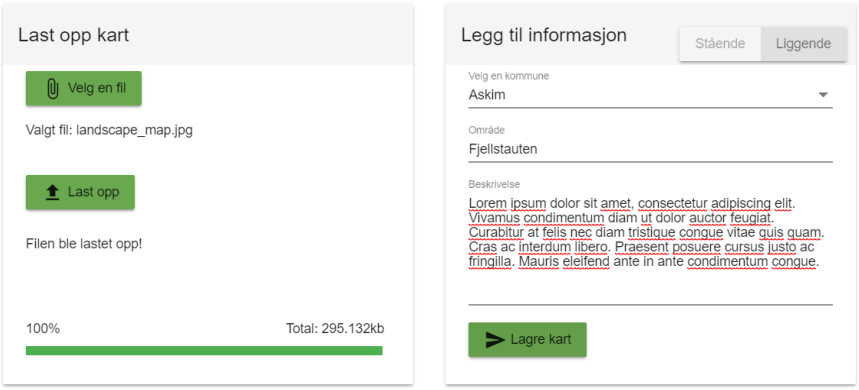


Figure 43. Upload map and write information

If an event was successful, the button will turn green letting the user know they can safely proceed.

User control page

This page should only be accessible to users with access level Super-admin or Admin (0,1). The page shows all users registered in the system and allows the administrators to change a user's access level or delete a user from the system. It is not yet feature complete and the team plans to add search functionality to filter users, and confirmations that the admin wants to change something.

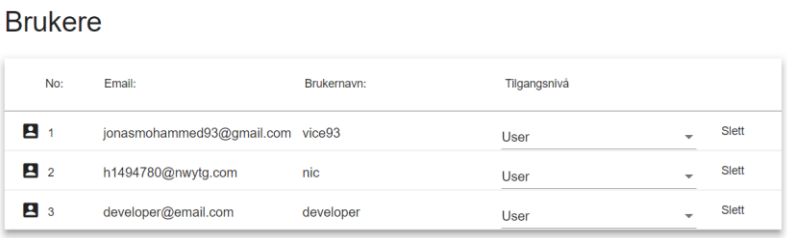


Figure 44: Admin user control.

8.4 The process

As mentioned in chapter 3.1 the initial project description wasn't what the contractor intended with the application, rather it was a brainstorm of ideas to pitch to the team. They only wanted a database to save maps in and a platform to view them in. However, this wasn't explained until a meeting held in the middle of February, which meant all the research the team did prior was no longer relevant to the project. This gave the team a rough start as they were given the task of figuring out what's best for the application themselves. Because of this, the team held multiple meetings amongst themselves and with the thesis advisor to sort out core functionality the application required. A few weeks later the team pitched these new requirements to the contractor, at which point they agreed to the terms and drafted a requirements specification for the team.

For instance, the idea of using SVG to draw a course on the map was pitched by the team as a way to make it simpler for teachers to create the courses. Instead of printing the maps out and drawing by hand, it was more efficient to provide this functionality in the application. This also meant courses could be saved for later in case teachers wanted to reuse an old course.

However, because of the late start everything else was also pushed further ahead. It took longer to finish researching, to holding the interviews, and as a result the system design and interface of the application wasn't completed until early April. Luckily the team did some extensive testing with SVG and the drawing functionality parallel with the research, which meant it was almost completed by the time it was required in the application.

The use of agile methods such as scrum, and design thinking helped the group throughout the development period. Drafting the wireframes and deciding how everything should look was much easier done when there was a set of rules and guidelines to follow. The team made sketches of functions and iterated on these multiple times, further improving the elements at each iteration, as are detailed in chapter 5.2.5.

Overall the team adhered to the design wireframes throughout development, though a few changes were made along the way. Firstly, the team split up the content editor and user management pages because it wasn't very intuitive to have to navigate to the user's page to publish articles. Secondly the team removed certain elements from the edit map tools, such as selecting a color as well as added a few such as saving and retrieving a course. And finally, the print and download buttons were removed from the map search page because the team realized users should at least view a map before printing or downloading it.

The use of scrum as an agile method wasn't incorporated until the team started actually developing. Before that, the team relied on mutual trust between its members to complete the assignments given. This worked out well, though most of the time the team was working in the same room anyway. However, once development of the actual application began, the team was a bit slow to incorporate scrum properly. It wasn't very efficient in the beginning, as the team spent a couple of weeks just setting up the core framework (the mean-stack). Though once the team properly set up a backlog on Trello and assigned different tasks, the productivity skyrocketed. Most of the applications functionality was then completed in less than a week, which meant the rest of the time could be used to extensively test, fix bugs and add additional functionality. While the team did not have time to perform user tests, at least the application saw rigorous testing amongst the team members, and fixes were pushed accordingly.

The team used GitHub as the version control software to make sure no progress was ever lost, to track changes made and to make sure everyone could work in parallel without issue.

8.5 Future work

The team is mostly happy with the current state of the application, however as mentioned earlier there are a few things the team did not have time to complete.

The news and activities page currently display both types on the same page, however in a recent meeting, the contractor specified they wanted these pages separated. This is a rather simple task and the team will resolve this in the next update of the application.

Each of the news/article cards contains two buttons (as seen in figure 45), both of which does not do anything as of writing this. The button “Les mer” is supposed to navigate to a new page to display the contents of this element. The content is simply retrieved from the current array in the clicked position making it easy to implement this function. The other button “Slett” deletes the post from the database, though the API has not been written to accept this request yet. This button will be hidden behind the user’s access level, meaning if the user is not of access level 2 (Content-manager) or higher, they will not be able to see it.

Figure 45: Activity card view.

The final component that requires improvement is the user control page. Right now, it only retrieves all users saved in the database and displays them. There are multiple functions the team intends to implement here.

Firstly, the option to search for users and filter them, which is solved by implementing a filter pipe much like the one made on the home page component.

Secondly a way to add a new user. This can be achieved by copying most of the functionality present in the register component.

Thirdly a way to change a specific users access level. The idea is to instantly perform the change when the user changes the selection box, however it's probably a good idea to include a confirmation here. And finally implement the delete user functionality when

clicking the “Slett” button. Again, it’s probably a good idea to include a confirmation here as well.

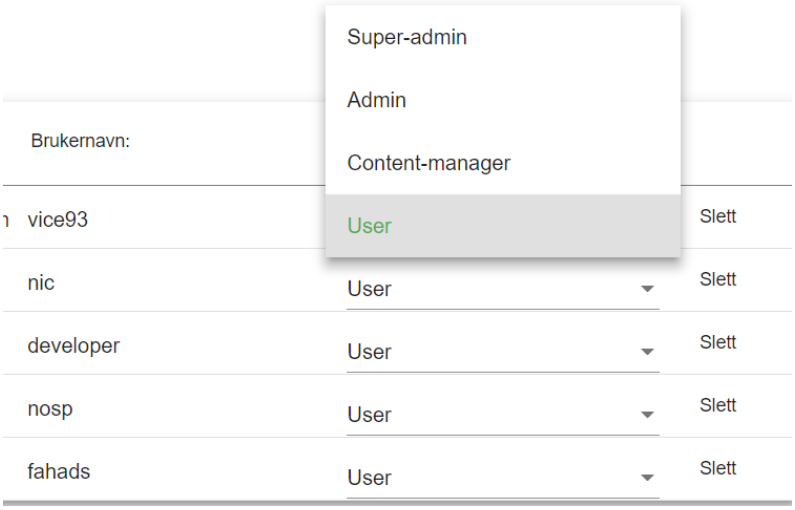


Figure 46: User control & Access levels.

The aforementioned issues are the ones that require immediate attention and improvement, however there are certainly more changes and fixes that needs to be done. Though most of those are bug fixes, code logic changes and general application restructure which should be done once everything else is completed.

Chapter 9 - Conclusion

The primary objective of this project was to digitalize the current orienteering practices in schools and clubs in Østfold. The intent of which was to simplify the process of storing, drawing and printing orienteering maps as well as help raise awareness around the sport leading up to the world orienteering championship in 2019.

To achieve this the team developed a web application incorporating the functionality requested by the contractor, that could be managed by ØOA and used by everyone interested in orienteering. Other than the basic functionality of uploading and saving maps in a database, and providing a platform for sharing, printing and downloading said maps, the team was free to decide on any functionality themselves.

To support the team's decisions, interviews with end-users were held as a way to map some of the ways the application would be used, and whether additional functionality was necessary. The team also had regular meetings with the stakeholders to confirm the current iteration of the application and design choices, and many features were redesigned as a result of this.

The end result is an application that with some additional work can immediately be transferred and maintained by Østfold Orienteering association. The application covers all the requirements imposed by the contractor as well as the ones set by the team themselves. The team believes that if the application launched, orienteering will not only be made easier for teachers and enthusiasts alike, but considerably more efficient and practical.

References

1. Angular (n.d.). Retrieved from https://docs.angularjs.org/guide/introduction
2. Express - Node.js web application framework. (n.d.). Retrieved from https://expressjs.com/
3. Svg.js (n.d.). Retrieved from http://svgjs.com/
4. What Is MongoDB? (n.d.). Retrieved from https://www.mongodb.com/what-is-mongodb
5. Chrome V8 (n.d.). Retrieved from https://developers.google.com/v8/
6. W3C. (2008). *Web Content Accessibility Guidelines (WCAG)*(Standard No. 2.0) Retrieved from https://www.w3.org/TR/WCAG20/#guidelines
7. Timo Foegen. (2010). What is Scrum? *Wibas*, s.1. Retrieved from https://www.wibas.com/media/filer_public/2013/09/30/wibas_whatisscrum_en.pdf
8. Orienteering. (n.d.). About orienteering. Retrieved from http://www.orienteering.ca/about/orienteering/
9. Norsk Orientering. (n.d.). Østfold. Retrieved from http://www.orientering.no/kretser/ostfold/
10. Jens Erik Mjølnerød. (2016). *Østfold O-Krets Årsberetning*. Retrieved from http://www.orientering.no/media/filer_public/6c/cd/6ccd8bdd-cbd2-4aa5-bcdf-7e78adb40f55/arsberetning_2016_oppdatert.pdf
11. Mike Cohn (2005). *Agile estimating and planning*,USA : Pearson education, inc.
12. Graham Crow. (2013). What is Qualitative Interviewing? *ISSN.* 2048–6812. Retrieved from http://eprints.ncrm.ac.uk/3276/1/complete_proofs.pdf
13. Individual interview. (n.d.). Retrieved from https://www.usability.gov/how-to-and-tools/methods/individual-interviews.html
14. 6 methods of data collection (n.d.). Retrieved from http://www.open.edu/openlearncreate/mod/resource/view.php?id=52658
15. OCAD (n.d.). Retrieved from https://www.ocad.com/en/functionality-chart/
16. Purple Pen (n.d.). Retrieved from http://purplepen.golde.org/#about
17. About NodeJS (n.d.) Retrieved from https://foundation.nodejs.org/about
18. Bloor, M. & Wood, F. (2006). Indexing. doi: http://dx.doi.org/10.4135/9781849209403.n30
19. Pete Bacon Darwin. (2018, 26.january). Stable AngularJS and Long Term Support[Blog post]. Retrieved from https://blog.angular.io/stable-angularjs-and-long-term-support-7e077635ee9c
20. Ian Sommerville. (2011). Software Testing, *Software engineering* (9. edition, p. 210-241)

21. Forskriftens formål. (2013. *Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)-løsninger*. Norge: Kommunal og moderniseringsdepartementet
22. Jerry Cao. (n.d.) The practical guide to empathy maps: 10 minutes user personas[Blog post]. Retrieved from https://www.uxpin.com/studio/blog/the-practical-guide-to-empathy-maps-creating-a-10-minute-persona/
23. Technopedia (n.d). Web-based application Retrieved from https://www.techopedia.com/definition/26002/web-based-application
24. Rikke Dam & Teo Siang (2018). *What is design thinking and why is it so important?, Design thinking's phases,* Retrieved from https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular
25. Angular - Services (n.d.) Retrieved from https://angular.io/tutorial/toh-pt4
26. Definition of Methodology (n.d.) Retrieved from https://www.merriam-webster.com/dictionary/methodology
27. Usability Testing (n.d.) Retrieved from https://www.usability.gov/how-to-and-tools/methods/usability-testing.html

Appendix

A.I - Back-end code

Code block 1 **[Dependencies]**

```
{
  "name": "meanstacktest",
  "version": "1.0.0",
  "description": "test mean stack with angular cli",
  "main": "server.js",
  "scripts": {
    "run": "nodemon server.js"
  },
  "author": "mike",
  "license": "MIT",
  "dependencies": {
    "bcrypt-nodejs": "0.0.3",
    "body-parser": "^1.18.2",
    "cors": "^2.8.4",
    "express": "^4.16.3",
    "jsonwebtoken": "^8.2.1",
    "mongoose": "^5.0.17",
    "multer": "^1.3.0"
  }
}
```

Code block 2 **[Modules]**

```
const express = require('express');
const router = express.Router();
const app = express();
const mongoose = require('mongoose');
const dbConfig = require('./config/database');
const path = require('path');
const authentication = require('./route/authentication')(router);
const map = require('./route/upload')(router);
const article = require('./route/articles')(router);
const bodyParser = require('body-parser');
const cors = require('cors');
const fs = require('fs');
const multer = require('multer');
const upload_dir = "./client/src/assets/uploads/";
```

```
const storage = multer.diskStorage({
  destination: function(req,file,cb){
    cb(null,upload_dir)
  },
  filename: function(req, file, cb) {
    cb(null, Date.now() + '.svg')
  }
})
const upload = multer({
  storage: storage
});
```

Code block 3 **[Defining routes]**

```
// authenticate
app.use('/authentication', authentication);
// upload maps
app.use('/upload', map);
// add article
app.use('/articles', article);
```

Code block 4 **[Server connection]**

```
app.use(cors({
            origin: [
              'http://localhost:4200',
              'http://158.39.162.173:4200',
              '<andre lokale iper du vil slippe til>',
    ],
    optionsSuccessStatus:200
}));
const port = 8080;
app.listen(port, () => {
    console.log('Server listening on port: '+ port);
});
```

Code block 5 **[Encryption]**

```
try {
    secret = crypto.randomBytes(256).toString('hex');
} catch (err) {
    console.log('ERROR' + err.message);
}
```

Code block 6 **[Hash]**

```
const hashedSecret = crypto.createHmac('sha256', secret)
   .update('DB')
   .digest('hex');
```

Code block 7 **[Database connection]**

```
module.exports = {
   uri: 'mongodb://localhost:27017/mean-stack-test',
   secret: hashedSecret,
   tokenSecret: secret,
   db: 'mean-stack-test'
};
```

Code block 8 **[Map schema]**

```
const mapSchema = new Schema({
   location: { type: String, required: true, validate: locationValidators },
   area: { type: String, validate: areaValidator },
   description: { type: String, validate: descValidator },
   image: { type: String, required: true, validate:imageValidators},
   postedby: { type: String },
   createdAt: { type: Date, default: Date.now },
});
```

Code block 9 **[Article schema]**

```
const ArticleSchema = mongoose.Schema({
  content: String,
  title: String,
  description : String,
  author: String,
  contentType: String,
  date: String
});
```

Code block 10 **[User schema]**

```
const userSchema = new Schema({
   email:  {type:String, required:true, unique:false, lowercase:true, validate: emailValidator},
   username: {type:String, required:true, unique:false, lowercase:true},
   password: {type: String, required: true },
});
```

A.II - Client-side code

Code block 1 **[Create svg element event]**

```
if(document.getElementById("createElement-input").checked &&
document.getElementById("selectSize").innerHTML &&
document.getElementById("selectType").innerHTML){
    $this = $(this);
    var x = GetPagXY(e).x - GetOffset().x;
    var y = GetPagXY(e).y - GetOffset().y;
    var rand = Math.floor(Math.random() * 100000000);
```

Code block 2 **[The svg objects]**

```
var type = parseInt(document.getElementById("selectType").innerHTML);
      if(type !== 1){
       newSvgElement.size =
parseInt(document.getElementById("selectSize").innerHTML);
       newSvgElement.type =
parseInt(document.getElementById("selectType").innerHTML);
       newSvgElement.id = 'svgElement' + rand;
       newSvgElement.position = [x * transformScale,y * transformScale];
       newSvgElement.scale = transformScale;

       if(input !== ""){
        var input = document.getElementById('writeText').value;
        newSvgElement.text = input;
       }
       data.push( Object.assign({}, newSvgElement) );
```

Code block 3 **[Mousedown/Touchstart event]**

```
function MouseAndTouchDown(x,y){
 if(document.getElementById("moveElement-input").checked) {
  var elementMouseIsOver = document.elementFromPoint(x, y).id;
     clicking = true;
     elem = DataInPosition(elementMouseIsOver);
 }
}
```

Code block 4 **[DataInPosition]**

```
function DataInPosition(id) {
 if(data.length){
  var pos = data.map(function (element) {
   return element.id;
```

```
  }).indexOf(id);
  return data[pos];
 }
}
```

Code block 5 **[Mousemove/Touchmove event]**

```
$('#svgCanvas').bind('mousemove touchmove', function(e){
  if(clicking && document.getElementById("moveElement-input").checked){
   e.preventDefault();
   var x = GetPagXY(e).x - GetOffset().x;
   var y = GetPagXY(e).y - GetOffset().y;
   if(elem){
    if(elem.position.length === 4){
     var x1 = elem.position[2],
       x2 = elem.position[0],
       y1 = elem.position[3],
       y2 = elem.position[1],
       xDif = Math.abs(x2-x1) / 2,
       yDif = Math.abs(y2-y1) / 2;

     if(x1 > x2){
      elem.position[0] = ((x * transformScale) - xDif);
      elem.position[2] = ((x * transformScale) + xDif);
     } else {
      elem.position[0] = ((x * transformScale) + xDif);
      elem.position[2] = ((x * transformScale) - xDif);
     }
     if(y1 > y2){
      elem.position[1] = ((y * transformScale) - yDif);
      elem.position[3] = ((y * transformScale) + yDif);
     } else {
      elem.position[1] = ((y * transformScale) + yDif);
      elem.position[3] = ((y * transformScale) - yDif);
     }
    } else {
     elem.position = [x * transformScale,y * transformScale];
    }
   }
```

Code block 6 **[CreateCircle and temporary objects]**

```
function CreateCircle(object,isTemp){
```

```
  var size       = object.size,
      x          = object.position[0] / object.scale,
      y          = object.position[1] / object.scale,
      id         = isTemp ? 'tempElement' : object.id,
      col        = isTemp ? '#000' : color,
      drawObject  = group.circle(size).id(id);
  drawObject.attr({
    fill: col,
    'fill-opacity': "0.0",
    stroke: col,
    'stroke-width': size/10,
    cx: x,
    cy: y
  });
}
```

Code block 7 **[Mouseup/Touchend event]**

```
$("#svgCanvas").bind('mouseup touchend',function(){
    clicking = false;
    if(SVG.get('tempElement') !== null){
      SVG.get('tempElement').remove();
    } else if(SVG.get('tempTriangle') !== null){
      SVG.get('tempTriangle').remove();
    }
    Update();
  });
```

Code block 8 **[SetTransformScale]**

```
function SetTransformScale(){
  var mapWidths = [500,400,300];
  if(Modernizr.mq('(min-width: 768px)')){
    transformScale = mapWidths[2] / mapWidths[2];
  } else if(Modernizr.mq('(min-width: 600px)')) {
    transformScale = mapWidths[0] / mapWidths[1];
  } else {
    transformScale = mapWidths[0] / mapWidths[2];
  }
}
```

Code block 9 **[GetOffset]**

```
function GetOffset(){
 var offset = {x: $this.offset().left, y: $this.offset().top};
 return offset;
}
```

Code block 10 **[GetPageXY]**

```
function GetPagXY(e){
 var arr = {x: e.pageX? e.pageX : e.touches[0].pageX, y: e.pageY? e.pageY :
e.touches[0].pageY};
 return arr;
}
```

Code block 11 **[GetClientXY]**

```
function GetClientXY(e){
 var arr = {x: e.clientX? e.clientX : e.touches[0].clientX, y: e.clientY? e.clientY :
e.touches[0].clientY};
 return arr;
}
```

Code block 12 **[DeleteSvgFromPoint]**

```
function DeleteSvgFromPoint(id){
 if (data.length) {
  var elementIdx = data.map(function (element) {
   return element.id;
  }).indexOf(id);

  if (elementIdx > -1) {
   redoData.push(data[elementIdx]);
   data.splice(elementIdx, 1);
  }
 }
}
```

Code block 13 **[Update]**

```
function Update(){
 $("#svgGroup").empty();
 var skipIndex = 0;

 for(var i = 0; i < data.length; i++) {
  if(data[i].scale !== transformScale){
```

```
    data[i].scale = transformScale;
   }
   if(data[i].type === 1 || data[i].type === 4){
    DrawPoint(data[i],i);
    skipIndex++;
   } else {
    DrawPoint(data[i],i - skipIndex);
   }
  }
}
```

Code block 14 **[Automatic numbering]**

```
var autoIndex = document.getElementById('autoIndex-input').checked;
 if(svgObject.type !== 1 && svgObject.type !== 4 && autoIndex){
  var x = svgObject.position[0] / svgObject.scale;
  var y = svgObject.position[1] / svgObject.scale;
  var objectNumber = group.text(function(add){
   add.tspan(index + 1).id(svgObject.id).fill(color);
  });
  objectNumber.font({
   size: svgObject.size
  });
  objectNumber.move(x + (svgObject.size), y );
 }
```

Code block 15 **[Undo/Redo/DeleteAll]**

```
 $('#undo').click(function(){
  var length = data.length;
  if(length !== 0){
   redoData.push(data[length-1]);
   data.splice(length - 1, 1);
   Update();
  }
 });

$('#redo').click(function(){
  var length = redoData.length;
  if(length !== 0){
   data.push(redoData[length - 1]);
   redoData.splice(length - 1, 1);
   Update();
  }
```

```
 });
});

$('#deleteAll').click(function(){
  redoData.push.apply(redoData,data);
  data = [];
  Update();
 });
```

Code block 16 **[Print Map]**

```
printMap(): void {
  let printContents, popupWin;
  printContents = document.getElementById('map-holder').innerHTML;
  popupWin = window.open('','_blank','top=0,left=0,height=auto,width=auto');
  popupWin.document.open();
  popupWin.document.write(`
   <html>
    <head>
     <title>Print Tab</title>
     <style>
      #image {
       padding=0px;
       margin=0px;
       height=750px;
       width=500px;
      }
      #svgCanvas {
       padding=0px;
       margin=0px;
       height=800px;
       width=750px;
      }
      #map-holder{
       padding=0px;
       margin=0px;
       height=800px;
       width=750px;
      }
     </style>
    </head>
    <body onload="window.print();window.close()">${printContents}
    </body>
   </html>`
  );
  popupWin.document.close();
```

```
}
```

Code block 17 **[module.js export functions]**

```
export function getData(){
 return data;
}

export function setData(dataArray){
 data = dataArray;
 Update();
}

export function update(){
 Update();
}
```

Code block 18 **[Save Course]**

```
saveCourse(){
   let user = this.authService.getUser();
   if(user === null){
     this.openSnackBar("Du må logge deg inn for å lagre en løype.","Lukk");
     this.close();
     return;
   }
   this.courseObject = getData();
   if(Object.keys(this.courseObject).length === 0){
     this.openSnackBar("Du må tegne en løype først.","Lukk");
     this.close();
     return;
   }
   else if(this.description != "" && this.title != ""){
     for(var i=0; i<Object.keys(this.courseObject).length; i++){
       if(this.courseObject[i].type !== 1 && this.courseObject[i].type !== 4){
         this.numberOfPoints ++;
       }
     }
     let course = {
       user: user,
       verified: true, //TODO: set this to true only if the user is from ØOA
       image: this.imagePath,
       course: this.courseObject,
```

```
      description: this.description,
      numberOfPoints: this.numberOfPoints,
      title: this.title
    }
    this.http.post(URL + '/saveCourse', course).subscribe(res => {
      let json = JSON.stringify(res);
      let responseObject = JSON.parse(json);
      let success = responseObject.success;
      let message = responseObject.message;
      if(success === true){
        this.openSnackBar("Løypa ble lagret!","Lukk");
        this.close();
        return;
      } else {
        this.openSnackBar("Kunne ikke lagre løypa. Prøv igjen senere.","Lukk");
        return;
      }
    });
  }
  this.openSnackBar("Vennligst fyll ut alle feltene","Lukk");
 }
```

Code block 19 **[Load courses]**

```
ngOnInit() {
  let user = this.authService.getUser();
  this.http.get<any[]>(URL + '/getVerifiedCourse?image=' +
this.imagePath).subscribe(res => {
    if(res.length > 0){
      this.noResults = "";
      for(var i in res) {
        if(res[i].verified === true){
          this.data.push(res[i]);
        }
        if(res[i].user === user){
          this.personalCourses.push(res[i]);
        }
      }

      return;
    }
```

```
    this.noResults = "Fant ingen løyper.";
  });
 }
```

Code block 20 **[Return the course]**

```
dialogRef.afterClosed().subscribe(result => {
    if(result){
      setData(result);
    }
  });
```

Code block 21 **[Load Maps]**

```
loadData(){
  this.mapData = [];
  let imagePath = '/assets/uploads/';
  this.http.get<any[]>(URL + "/getMap?location=" + this.selectedLocation).subscribe(res => {
    if(res.length > 0){
      this.noResults = "";
      for(var i in res) {
        let imagename = res[i].image.slice(0,-4);
        res[i].image = imagePath + imagename + '.png';
        this.mapData.push(res[i]);
      }
      return;
    }
    this.noResults = "Fant ingen kart i " + this.selectedLocation;
  });
 }
```

Code block 22 **[Filter pipe]**

```
import { Pipe, PipeTransform } from '@angular/core';
@Pipe({
  name: 'filter'
})
export class FilterPipe implements PipeTransform {
  transform(items: any[], searchText: string): any[] {
    if(!items) return [];
    if(!searchText) return items;
      searchText = searchText.toLowerCase();
      return items.filter( it => {
        return it.description.toString().toLowerCase().includes(searchText) ||
it.area.toString().toLowerCase().includes(searchText);
```

```
    });
  }
}
```

Code block 23 **[Using the filter pipe in html]**

```
<mat-card *ngFor="let map of mapData | filter: searchText; let i = index"
class="example-card map-card-holder">
```

Code block 24 **[Read more about map]**

```
readMore(i) {
  let dialogRef = this.dialog.open(MatDialogComponent, {
    data: this.mapData[i],
    width: '600px'
  });
 }
```

Code block 25 **[Opening a mat-dialog and passing the data]**

```
export class MatDialogComponent implements OnInit {

 area:string;
 description:string;
 image:string;

 constructor(
     private dialogRef: MatDialogRef<MatDialogComponent>,
     @Inject(MAT_DIALOG_DATA) {
       area,
       description,
       image
     }: any) {
       this.area = area;
       this.description = description;
       this.image = image;
     }
 }
```

Code block 26 **[Navigating to edit-map-component]**

```
createCourse(i){
  //Change the file exention back to .svg before going to the course drawing tool
  this.mapData[i].image = this.mapData[i].image.slice(0,-4) + '.svg';
```

```
  this.router.navigate(['/edit-map'], { queryParams: { path: this.mapData[i].image}});
 }
```

Code block 27 **[Selecting map file for upload]**

```
onFileSelected(event) {
  if(this.uploadButtonClicked === true){
   this.openSnackBar("Fyll ut informasjon før du laster opp ett nytt kart.","Lukk.");
   return;
  }
  this.selectedFile = <File>event.target.files[0];
  if(this.selectedFile != null){
   let ext = this.selectedFile.name.match(/\.([^\.]+)$/)[1];
   switch(ext)
   {
      //Add filetypes you want to accept here
      case 'jpg':
      case 'jpeg':
         this.fileSelectMessage = "Valgt fil: " + this.selectedFile.name;
         this.findButtonClicked = true;
         break;
      default:
         this.openSnackBar("Filtypen er ikke godkjent.","Lukk")
         this.fileSelectMessage = "";
   }
   return;
  }
  this.reset();
 }
```

Code block 28 **[Upload the map file]**

```
onUpload(){
  //If the map is already uploaded, prevent uploading it again.
  if(this.uploadButtonClicked === true){
   this.openSnackBar("Dette kartet er allerede lastet opp.","Lukk");
   return;
  }
  //If the user hasn't selected a file yet, ask them to
  if(this.selectedFile === null){
   this.openSnackBar("Velg en fil først.","Lukk");
   this.findButtonClicked = false;
   return;
```

```
  }
  //Create a new form and send the image file to the server
  const fd = new FormData();
  fd.append('map',this.selectedFile);
  this.http.post(URL, fd, {
    reportProgress: true,
    observe: 'events'
  }).subscribe(event => {
    //Subscribe to the response
    if(event.type === HttpEventType.UploadProgress){
      this.progress = Math.round(event.loaded / event.total * 100);
      this.progressMessage = this.progress + "%";
      this.totalMessage = "Total: " + (event.total/1000).toString() + "kb";

      //Display upload progress
      if(this.progress === 100){
        //console.log(event.statusCode);
        this.successMessage = "Filen ble lastet opp!";
        this.uploadButtonClicked = true;
        this.displayCard = true;
      }
      //Get the stored map's name
    } else if(event.type === HttpEventType.Response){
      Object.getOwnPropertyNames(event.body).forEach(key => {
        let value = event.body[key];
        this.res.push(value);
        this.responseMessage = this.res[1];
      });
      console.log(this.responseMessage);
    }
  });
}
```

Code block 29 **[Add information to the map]**

```
onSaveInformation(event){
  let map = {
    location: this.selectedLocation,
    area: event.path[0][0].value,
    description: event.path[0][1].value,
    image: this.responseMessage,
    postedby: "test2" // add currently logged in user here & add validation for them
```

```
  };
  try {
    // Post the map information and save it in the database
   this.mapService.mapPost(map).subscribe(res => {
     // Initialize the response object as a new object so we can read the values
     let json = JSON.stringify(res);
     let responseObject = JSON.parse(json);
     // read the response values
     let success = responseObject.success;
     let message = responseObject.message;
     // If false, map did not get saved
     if (!success) {
       console.log("Bad response: " + message);
       this.openSnackBar("Vi kunne ikke lagre dette kartet. Prøv igjen senere.","Lukk");
     } else {
       this.postButtonClicked = true;
       this.openSnackBar("Kartet er lastet opp!","Lukk");
       console.log("Success: " + message);

       setTimeout(() =>
       {
          this.reset();
       },
       5000);
     }
  }
```

Code block 30 **[Publish article function]**

```
publishArticle(){
  let content = {
    data: document.querySelector(".ql-editor").innerHTML,
    title: this.titleValue,
    description: this.descValue,
    contentType: this.contentType,
    author: this.authService.getUser()
  };
  this.http.post(this.domain.url + '/articles/add', content).subscribe(res => {
  let json = JSON.stringify(res);
  let responseObject = JSON.parse(json);
  let success = responseObject.success;
  let message = responseObject.message;
```

```
  if(!success){
    console.log("Bad response: " + message);
    this.openSnackBar("Noe gikk galt.","Lukk");
  } else {
    this.openSnackBar("Nyhets artikkelen er publisert!", "Lukk");
    console.log("Success: " + message);
  }
 });
}
```

Code block 31 **[Toggle type of article shown in list]**

```
toggleGroup(){
  this.displayContents = [];
  this.value = document.getElementById('toggle_value_hidden').innerHTML;
  if(this.value === "activity"){
    for(let i in this.activities) {
      this.displayContents.push(this.activities[i]);
    }
  } else {
    for(let i in this.news) {
      this.displayContents.push(this.news[i]);
    }
  }
 }
```

Code block 32 **[Read more about article]**

```
readMore(i){
  if(this.value === "activity"){
    //Navigate to activities page with clicked element
    console.log(this.activities[i]);
  } else {
    //Navigate to news page with clicked element
    console.log(this.news[i]);
  }
 }
```

Code block 33 **[Load data when page initialises]**

```
ngOnInit() {
  this.contentService.getContents().subscribe( content => {
    for(let i in content) {
```

```
      if (content[i].contentType === "activity") {
        this.activities.push(content[i]);
        this.displayContents.push(content[i]);
      } else {
        this.news.push(content[i]);
      }
    }
  });
}
```

Codeblock 34: **[Initialize quill, js function]**

```
var toolbarOptions = [
  ['bold', 'italic', 'underline', 'strike'],
  ['blockquote', 'code-block'],
  [{ 'header': 1 }, { 'header': 2 }],
  [{ 'list': 'ordered'}, { 'list': 'bullet' }],
  [{ 'script': 'sub'}, { 'script': 'super' }],
  [{ 'indent': '-1'}, { 'indent': '+1' }],
  [{ 'direction': 'rtl' }],
  [{ 'size': ['small', false, 'large', 'huge'] }],
  [{ 'header': [1, 2, 3, 4, 5, 6, false] }],
  [{ 'color': [] }, { 'background': [] }],
  [{ 'font': [] }],
  [{ 'align': [] }],
  ['clean'],
  ['link', 'image', 'video']
];

var quill = new Quill('.editor', {
    modules:{
      toolbar:toolbarOptions
    },
    theme: 'snow'
  });
```

A.III - Services and page guards

Code block 1 **[Create auth header]**

```
createAuthenticationHeaders() {
 this.loadToken();
 this.httpOptions = {
  headers: new HttpHeaders({
   'Content-Type': 'application/json',
   'authorization': this.authToken
  })
 };
```

Code block 2 **[Store token]**

```
storeUserData(token, user) {
 localStorage.setItem('token', JSON.stringify(token));
 localStorage.setItem('user', JSON.stringify(user));
 this.authToken = token;
 this.user = user;
}
```

Code block 3 **[Load token]**

```
loadToken() {
 this.authToken = localStorage.getItem('token');
}
```

Code block 4: **[Check user logged in]**

```
loggedIn() {
 this.isExpired = this.helper.isTokenExpired(localStorage.getItem('token'));
 return this.isExpired;
}
```

Code block 5: **[Post map information]**

```
mapPost(map) {
 this.createAuthenticationHeaders();
 let user = this.authService.getUser();
 map.postedby = user;
 return this.http.post(this.domain.url + '/upload/newMap', map, this.httpOptions).map(res
=> res);
}
```

B - Meeting minutes

Møtereferat

Dato	09.01.2018
Tilstede	Børre Stenseth, Jonas Vestgarden, Michael Simon, Adis Jasarevic
Referent	Jonas Vestgarden

Agenda

1. Bli kjent
2. Hva skal gjøres
3. Hvordan bør vi jobbe

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Oppdragsgiver må kontaktes. Kravene som er nevnt i arbeidsbeskrivelsen er bredt og sammen med arbeidsgiver oppgaven må avgrenses.	15. jan. 2018	Alle
2	Ordne trec for å håndtere prosjektet. IT ansvarlig (Ted) skal kontaktes.	15.jan.2018	Jonas
3	Vi må et navnt til prosjektet (Applikasjonsnavn)	15.jan.2018	Alle

Note: Adis var til stedet etterhvert og har oversikt på hva som skal utføres fremover.

Møtereferat

Dato	Fredag 12. Jan. 2018
Tilstede	Michael Mebrhatu Simon Jonas Vestgarden Adis Jasarevic Per Bergerud Elsie Brenne
Referent	Michael M. Simon

Agenda

1. Diskutere detaljert prosjekt kriterier og krav
2. Introduksjon

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Vi har diskutert om oppdraget og oppdragsgiver har gitt et inntrykk av at prosjektbeskrivelsen som er lagt ut på bacheloroppgaver siden er noe de ønsker seg generelt men ikke et absolutt krav. Vi diskuterte videre hva målet med prosjektet er og de nevnte hovedløsningen de ser etter er et system som håndterer orienteringskart. Med dette skal systemet gjør deling av kart enkelt og orienterings sport spennende for lærere.		
2	I møtet var det avtalt andre person skal være ansvarlig og vi skal ha kontakt med han hvis vi eventuelt har spørsmål. Svend Sondre Frøshaug tel: 994 75 032 Michal m. simon skal ta kontakt med via mail		

Møtereferat

Dato	17/01/2018
Tilstede	Adis Jasarevic Michael Mebrhatu Simon Jonas Vestgarden Børre Stenseth
Referent	Adis Jasarevic

Agenda

1. Diskutere litt om hva vi snakket om med oppdragsgiver
2. Finne ut av hva vi burde gjøre
3.

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	ta konsekvenser av samtalen vi hadde og komme opp med en ide.	17.jan	Alle
2	Designe en løsning	17.jan	Alle
3	Jobbe iterativt (Metode)	17.jan	Alle
4	Fullføre og linke til forprosjektrapport	17.jan	Alle
5	Finne SVG fil	17 jan	Alle

Note:

Møtereferat

Dato	Onsdag 24. Jan. 2018
Tilstede	Michael Mebrhatu Simon Jonas Vestgarden Adis Jasarevic Børre Stenseth
Referent	Adis Jasarevic

Agenda

1. Diskutere prosjektbeskrivelse
2. Snakke om SVG løsningen så langt

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Fixe videre på dokumentet	24.jan	Alle
2	Få til samtalen med Sven Sondre.	24.jan	Alle
3	Teste ulike verktøy og kilder	24.jan	Alle

Note:

Møtereferat

Dato	Tirsdag 30. Jan. 2018
Tilstede	Michael Mebrhatu Simon Jonas Vestgarden Jens Erik
Referent	Michael M. Simon

Agenda

1. Diskutere detaljert prosjekt kriterier og krav
2. Introduksjon til Østfold orienteringskret
3. Funksjonaliteter til web applikasjon

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Applikasjonen skal gjøre lettere for lærere å hente dokumenter som er laget for kompetanseheving og undervising metoder som er nyttige. Jens Erik skal sende oss linker til de nødvendige dokumentene og dette skal legges ut på websiden som en ekstern link som en ressurs.	Før utvikling av applikasjon begynner	Jens Erik
2	Lisens til OCAD skal ordnes, om dette ikke er mulig skal Jen Erik Sende oss SVG filer som skal brukes til å teste mens systemet skal utvikles.	30.feb.2018	Jens Erik
3	Det ble nevnt om det er mulig å delen roller av brukere. Det skal være admin og vanlige bruker(lærere) de to brukere skal ha forskjellige roller og tilgang til funksjonaliteter. Dette skal beskrives i første utkast av dokumentet og system design dokumentet.		
4	Brukere skal ikke være avhengig av å bruke OCAD det skal også være mulig å legge ut vanlig bilder eller pdf tegne på den om de vil. Eller dem skal printe den og tegne posten med pen.		
5	Jens Erik nevnt sikkerhet og hvordan en bruke skal registeres, login osv… Angående third party login systemer kan føre til en del andre utfordringer. Bruk av API eller kjøp av slike tjenester er alternativet. Om dette skal ikke utføres i dette prosjektet skal det være mulig å implementere slike autentifikasjon funksjonalitet eventuelt hvis produktet skal utvikles		

	videre.		

Møtereferat

Dato	Onsdag 31. Jan. 2018
Tilstede	Michael M. Simon Adis Jasarevic Børre Stenseth
Referent	Adis Jasarevic

Agenda

1. Diskutere møtet som var med oppdragsgiver

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Teknisk Avklaring		Alle
2	Conceptual design, UI		Michael
3	Vi venter kravspec fra Jens		

Note:Jonas hadde et intervju derfor møte han ikke opp, han har gitt beskjed tidlig.

Møtereferat

Dato	07.02.2018
Tilstede	Michael M Simon, Adis Jasarevic, Jonas Vestgarden
Referent	Adis Jasarevic

Agenda

1. Snakker om selve systemet og planene.
2. Snakker om hva vi bør bruke tiden vår på mest i dette prosjektet.
3. Snakker om hva skal gjøre videre. Design på overordna nivå.

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Få beskjed slik at vi får kravspesifikasjon	7. Februar	Alle
2	Ocad lisens(Ennå ikke i boks, venter fortsatt)	7. Februar	Alle
3	Kvalitativ undersøkelse(skal utføres på en av skolene)	7. Februar	Alle
4	Fortsette å skrive på dokumentet. Samle inn kilder. 9.3.2018 er frist. Gjennomgang og metodikk bør holdes på riktig vei.	7.Februar	Alle
5	System:(Få lærere til å dele erfaring)	7.februar	Alle
6	Vi bør starte design neste uke	7.februar	Alle

Note: Vi glemte å kjøpe kaffe.

Møtereferat

Dato	28.02.2018
Tilstede	Michael M Simon, Adis Jasarevic, Jonas Vestgarden
Referent	Adis Jasarevic

Agenda

1. Snakker om selve kravspesifikasjonen og planene vi må gjøre.
2. Snakker om første versjon av dokumentet.

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Finne ut om vi skal bruke cms som open source plattform, eller lage et system selv fra scratch slik at vi ikke har noen begrensninger.	28. Februar	Alle
2	Snakke med svend sondre før vi skal beslutte oss til en avgjørelse.	28. Februar	Alle
3	Endre på dokumentets oppbygging og legge inn mer informasjon.	28. Februar	Alle
4		28.Februar	Alle
5		28.februar	Alle
6		28.februar	Alle

Note: Vi glemte å kjøpe kaffe.

Møtereferat

Dato	07.03.2018
Tilstede	Michael M Simon, Jonas Vestgarden
Referent	Jonas Vestgarden

Agenda

1. Gå igjennom emner diskutert med oppdragsgiver 06.03.2018
2. Se videre på hoveddokument første versjon

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Levere hoveddokument første versjon til Børre	07. Mars	Alle
2	Snakke med Ted om server for NodeJS	07. Mars	Alle
3		07. Mars	Alle
4		07. Mars	Alle
5		07. Mars	Alle
6		07. Mars	Alle

Note: Vi dro på Finn.no sitt seminar i Oslo etter dette møtet

Møtereferat

Dato	Onsdag 14. mars. 2018
Tilstede	Michael M. Simon Adis Jasarevic Børre Jonas
Referent	Adis Jasarevic

Agenda

1. Diskutere den innleverte versjon 1 av dokumentet.

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Endre på dokumentet		Alle
2	Få i gang utviklingen av produktet		Alle
3	Bli ferdig med plan om designelementer etc		Alle

Note: To rom ble booket. Men alle kom til det riktige rommet.

Møtereferat

Dato	Onsdag 04. apr. 2018
Tilstede	Michael M. Simon Jonas Vestgarden Børre Stenseth
Referent	Jonas Vestgarden

Agenda

1. Analyse - Hva skal vi analysere?

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	For analyse kan vi også snakke om valg vi har tatt i forhold til verktøy osv. i tillegg til intervju med lærere.		Alle
2	Antall intervjuer til nå (2 stk) er for lite til å kalle det en undersøkelse, prøv å ha minst 4 om vi skal basere valg på resultatene.		Alle
3	Få satt opp en server for å teste deployment av løsningen. Snakk med Ted.		Alle
4	Få skrevet mer i dokumentet, spesielt Analyse kapittelet.		Alle

Note:

Møtereferat

Dato	11/04/2018
Tilstede	Adis Jasarevic Michael Mebrhatu Simon Jonas Vestgarden Børre Stenseth
Referent	Adis Jasarevic

Agenda

1. Snakke om selve strukturen og koden

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	vi skal snakke med ted angående server bruk	11/06/2018	Alle
2	vi skal fokusere på dokumentet og dens innhold før mandag	11/06/2018	Alle
3			Alle
4			Alle
5			Alle

Note: vi kjøpte kaffe

Møtereferat

Dato	Onsdag 04. apr. 2018
Tilstede	Michael M. Simon Adis Jasarevic Jonas Vestgarden Børre Stenseth
Referent	Jonas Vestgarden

Agenda

1. Gå igjennom hoveddokumentet så langt

Saker

Sak	Beskrivelse	Forfall	Ansvarlig
1	Skrive en setning om at vi har valgt engelsk (vi har intervjuene på norsk).		Alle
2	Pupils over children kanskje		Alle
3	Få definert ØO tidlig, så kan vi bruke forkortelsen senere. Finne en Engelsk definisjon på det siden vi skriver på engelsk		Alle
4	Bytte "runs" til "competitions" kanskje		Alle
5	Ikke nevne "world cup 2019" så mye, en gang i starten holder.		Alle
6	Skille mellom skolenes papir kart og ØO digitale kart tidlig i oppgava.		Alle
7	Tydeliggjøre at lærere har tilgang til kart for deres område		Alle
8	Referere til vår definisjon av MVC (i terminologi)		Alle
9	Unit testing er et stort begrep vi ikke kan bruke, det er ikke noe vi har tid til uansett		Alle
10	Orienteering termer må defineres bedre gjennom oppgava		Alle
11	Agile development i Overall Process mangler (Bytte navn til Scrum)		Alle
12	Mer kortfattet definisjoner i terminologi, og beskrive mer i overall process på Design Thinking, Scrum osv.		Alle

13	I research kapitellet: Hold research rundt libraries osv. kortfattet (hvordan vi gikk frem osv).		Alle
14	Snakket med oppdragsgiver osv. og fikk definert oppgava, kanskje snakke om det i starten av Kap. 4		Alle
15	Ta vekk research kapitellet og heller inkludere det i kapitell 5 som egne titler. Se på dette og kom til en løsning.		Alle
16			

Note:

C - Interview transcripts

Intervju med Lærer 1 - Låby skole

Format: Semi-structured
Dato: torsdag 8. mars 2018

1. Hvilken nytte har elevene av orientering?

"Elevene har stor nytte av orientering fordi det å lære seg kompass og kart er kjempenyttig videre i livet. Det er også veldig morsomt for dem fordi det er ikke noe vi bruker spesielt mye tid på. Spesielt når vi har gjort det så ser jeg at dem har det kjempegøy. "

2. Kan du nevne noen utfordringer med orienteringssporten?

"Det som er utfordringen vår er at vi ikke har utstyr til det. Vi har ikke et klassesett, så når vi er f.eks. 27 må vi organisere de kompassene vi har og litt av det gamle utstyret. Men vi er så heldige at vi har hatt orienteringsklubben her som tilbyr seg å komme og undervise. Jeg som lærer kan da slappe litt av med det og vet at de dukker opp med et 3 timers opplegg i tillegg til at de har proft utstyr."

2.1 Off topic spørsmål: Lærer dere opp alle barna med kart og kompass?

"Storskolen og oppover, altså 5 klasse til 7 klasse får tilbudet om å lære kart og kompass. Jeg er veldig sjeldent lærer for småskolen så jeg vet ikke helt hva lærerne der gjør og hvordan opplæringa er. Det er altså ingen god, helhetlig plan på kart opplæring, det er heller litt sånn tilfeldig. Men jeg har sett flere sånne håndtegnet kart i gymsalen. "

2.2 Off topic spørsmål: Når orienteringsklubben kommer hit, kommer de med eget utstyr og eget kart for området?

"Ja. Sist dem var her, i fjor, så var det vel halden skiklubb som kom, og da hadde dem laget et kart over området, altså over skolen som vi fikk en kopi av. Så akkurat skolen og det nærmeste området rundt som var på det kartet. Men jeg vet ikke hvor jeg kan finne det kartet, jeg tror ikke det ligger i noe system, jeg tror bare de hadde laget det for det opplegget. Og da har de med seg sånn håndkompass og ordentlig maskin for å pipe inn på tid, sånn tøft utstyr, så ungene synes dette er kjempemorsomt. Det er en happening da. Men det å gjøre litt mer før eller etter, det er opp til oss, og det skal jeg gjøre i neste runde for å si det sånn."

3. Tror du noen av disse kan løses med hjelp av teknologi?

"Ja hvis det er en type bank et sted som jeg kan hente kart fra, det hadde løst veldig mange problemer. Før har jeg tegnet kart for hånd, for eksempel ved å finne området over på google maps og det tar veldig mye tid. Ellers så har vi en sånn dokumentmappe med 30 år gamle kart som vi kan kopiere og tegne på."

4. Får dere tilbakemelding av elever på hvordan man kan forbedre aktivitetene og om de trivdes med sporten.

"De synes det er veldig morsomt fordi det er gøy for dem. Altså jeg ser at det fenger veldig, fordi det er alt du gjør i gymtimen som ikke har med ball å gjøre som er ganske gøy. Man har et rart forhold til det, enten så liker man fotball eller så liker du det ikke, og det er så veldig viktig å få opp alle de andre talentene da. Plutselig har du barn som ikke er så glad i gym som synes det er utrolig gøy og løper rundt når de har noe å gjøre. Så det å orientere, det er veldig gøy for mange som i utgangspunktet ikke synes gym er gøy. Og når det kommer besøk så er det jo helt fantastisk, da er dem jo himmelrike med en gang.

Og hadde vi hatt det litt proft, hadde jeg fått kart så hadde jo det hjulpet på min undervisning også og det blir kult med en gang. ”

5. Hvordan kan man gjøre sporten spennende for studenter og lærere?

“Lære meg mer kunne gjort det spennende? ”

5.1 Clarification: Den ene planen til Østfold orienteringskrets er å motivere læreren til å engasjere seg med aktiviteten.

“Vi elsker jo kurs da, med en gang vi blir lært opp til noe, altså alt man kan er jo morsommere, så det hadde jo vært fantastisk å fått en god innføring sjøl fra noen som er proffe. Og bare det og ha litt mer utstyr tilgjengelig, det motiverer jo det også. Vi har jo ikke flust av penger og utstyr så så fort det kommer noe ekstra så er jo det motiverende i seg selv.”

6. Hvor ofte har dere orientering på skolen.

“Det er en periode hvert år i storskolen som dem skal igjennom. Hvor lang den perioden er, det er jeg ikke helt sikker på for jeg har ikke helt oversikt på de andre lærerne og hva de gjør. Så det ligger jo i fagplanen at de skal lære seg orientering i løpet av storskolen, 5 klasse ut 7. Så det er litt forskjellig når det kommer i opplegget, men nå har vi jo som sagt hatt besøk de siste 3 årene så de som går i 7 nå har jo liksom hatt litt orientering hvert år. Totalt en sånn 2-3 uker tror jeg, hvert år.”

6.1 Off topic: Så hvis du hadde hatt mer utstyr tilgjengelig, tror du det hadde blitt lagt opp til mer orientering totalt?

“Jaja, det er jo helt klart. Nå er det jo veldig opp til oss, vår interesser. Meg som gymlærer kan jo bestemme veldig mye innanfor læreplanen. Er du interessert i fotball, blir det kanskje mye fotball i gymtimene. Vi skal jo nå kompetansemåla, men det er vi som lager planen. Det er lettere å undervise i ting du er motivert for og ting du kan selv. Er du plutselig plassert som gymlærer og ikke har lyst så er det kjipt. Vi får jo beskjed om hva vi skal, men vi kan jo ønske oss det vi vil”

7. Hvor henter dere kartet fra?

Dette fikk vi svar på tidligere, hoppa over.

8. Hvordan markerer dere orienteringskarta?

“De siste karta jeg så, der var det markerte poster, så de kan jeg ikke endre noe på. Men som sagt, de selvtegnede kan jeg fikse på helt selv. Poster må jeg legge selv”.

9. Er det noen utfordringer for å tilgang til kart?

“Lite tilgjengelighet, må lage det selv. Vi har en gammel sånn til å sette papir i vet jeg, med kart i, så vi må grave litt her og finne det som er best og tegne på det.“

10. Litt forklaring på hva vi lager, funksjonalitet, kart tilgjengelighet osv.

“Og da ser dere for dere flere steder i halden ikke sant? Jeg får ikke tilgang bare til skoleområdet, da kan jeg finne området rundt i remmen, altså andre steder?”

10.1 Clarification: Ideen er jo at man skal bygge opp denne banken med kart for nærområdet, skole og skog slik at lærer kan gå inn og hente kart for stedet dem selv ønsker.

“Ikke sant, og det er jo det som er kjempeinteressant for meg når jeg kan ta klassen med meg ut og bort herfra også. En ting er jo her, det er jo helt gull, men å kunne rød herregård for eksempel, eller remmen eller silledalen, altså et eller annet område hvor vi også kan finne kart over, ikke sant”

10.2 Off topic: Vi pratet litt med Hjortsberg skole og dem la frem at de hadde kun samme område å holde orientering på og det ble kjempe kjedelig for barna i lengden.

“Her er det heller kanskje litt motsatt, her er det ikke så mye av det samme i det hele tatt nesten, litt mer tilfeldig mer spredt kanskje.”

10.3 Off topic: Det skal også på samme nettside bli laget litt artikler/resources som dere kan bruke som orienteringskretsen skal publisere. Det blir kanskje ikke direkte kurs, men dere får kanskje tilgang til litt forskjellig informasjon som trengs, bl.a. hvordan man utfører forskjellige aktiviteter.

“Så bra, for det var det jeg skulle spørre om nå, fordi det som letter vår jobb veldig og gjør oss, altså at vi kommer til å bruke det mer, er jo hvis det kommer gode planer. Det er ikke det at vi er late, men det er innmari mye å gjøre. Så det at det da ligger litt klart f.eks. litt ideer til opplegg, øvelser, sånne ting som passer for barn, det hadde jo vært helt fantastisk. For da vil man jo bruke det enda mer. At man kan gå og finne litt ideer, ikke at det skal være ferdig servert, men litt gode ideer som er utarbeidet av de som kan det. Er det halden skiklubb?”

10.4 Off topic: Det er Halden Orienteringskrets, men de jobber en del parallelt med halden skiklubb.”

“Veldig klokt av klubber og satse, fordi ungene er jo der, og de blir jo interessert i det som blir lagt frem for dem, og det hjelper for rekruttering. Jeg heier på det, det er sunt å være ute så det er noe med det at idretter er lurt å få opplæring i.”

10.5 Off topic: Neste gang, om du har muligheten til det, så kommer vi med en liten prototype av funksjonalitet vi kan gå igjennom med dere. Så tester vi sammen og ser om det er noe funksjonalitet som trengs. Vi vil at applikasjonen skal være brukerorientert, og de som skal bruke den skal være fornøyd og ha de verktøyene de trenger. Vi vil ikke sitte og bestemme at det skal være den, den og den funksjonaliteten.

“Men da skal jeg jobbe litt hardere for å få med meg fler så vi er i litt ulike kategorier her, det er noen som er redd for data og sånn her vet du, av disse gamle lærerne som ikke tør å ta på en datamaskin, vi har fortsatt de. Så om jeg får med meg litt fler kan vi teste et bredt spekter, hahahaha, ikke at jeg er veldig god på data, men jeg er ikke redd for nye ting da.”

Intervju med Lærer 2 - Hjortsberg skole

Format: Semi-structured
Dato: onsdag 14. mars 2018

Hvilken nytte mener du elevene har om orientering?

Den er stor, fordi for det første så er det viktig for dem å lære/å bruke et kart og det er et sikkerhetsperspektiv også. Når dem er på tur og behersker kart og kompass har dem muligheten til å orientere seg og komme seg hjem hvis det skulle skje noe blant annet. Så den biten er ganske stor! Også er det at de må bruke kroppen samtidig som de bruker hodet, det er bevist at det er veldig lurt og orientering er helt max der, så det er vel det beste.

Er det noen utfordringer med orientering?

Utfordringene er det at det er tidkrevende, feks sette ut poster hvis det over et stort område. Og karta hvis dem ikke er oppdaterte så skal det mye ting til at alt glir.

Postene må jo settes ut uansett?

Ja det må dem, men hvis alt er i orden så er det enklere at vi koordinerer på skolen og at vi har orientering samme uken og da blir det et annet fokus rundt det. Nå er det veldig enkelt å la det glippe. Så det jeg tror er at det er det viktigste. For hvis dere lager et bra opplegg som vi kan bruke, så tvinger det oss fram og vi har veldig lyst til det, men da blir det sånn at det er naturlig for oss og sette oss ned å bruke mer tid på det.

Mange mener at det er utfordringer til ustyr, hvordan er det?

Vi har noen gamle kart som er utdaterte, også har vi vel noen poster. Åssen det er med kompass det vet jeg ikke. Også vet jeg ikke helt om man må ha kompass for å orientere på mellomtrinnet, også er det jo gjerne ledelinjer , stier og sånne ting. Så jeg trukke utstyret er det største problemet sånn sett.

Tror det er andre problemer enn det du sa som kan løses med en sånn teknologi som vi lager?

En utfordring som jeg ikke nevnte som kan dukke opp innimellom, det er jo den der sikkerhets greia om at unger forsvinner. Orientering er jo en av de få tinga der unger kan faktisk ende på å bli borte. Så den sjansen må jo bare ta. Det er jo det uansett.

Får dere noen tilbakemeldinger fra elever på f eks. om de trivdes med orientering?

Alle som dem mener har orientering så trives alle med det. Jeg tror alle liker det. I klassen min her så er det faktisk 9/12 gutter som driver med orientering. Det er veldig lavterskel, men dem er på orientering hver uke. Så for oss ville det ha vært helt magisk!

Hvordan kan vi gjøre sporten spennende for både dere og elever? Hvordan kan man gjøre det tenker du?
Jeg vet ikke, kanskje passe vriene løyper eller oppsetninger, det er jo kanskje det viktigste. Sånn at det er utfordringer, men at det ikke er for vanskelig. Det er mulig at det er å legge opp til, hvis dere kunne kanskje legge opp til noe slags differensiering og at det er noe fornuftig. Men jeg vet ikke helt om det er det dere skal eller om dere skal bare lage tekniske varianter eller om dere skal ha noen pedagogiske å?

Vi skal også ha muligheten til å linke til forskjellige opplegg og forskjellige aktivitetsforslag.

Ja det er bra, fordi det er stor forskjell på de som kommer til å mestre det her best og de som mestrer det dårligst. Får dem samme oppgave så kommer det ikke til å funke så godt. Så det å ha litt ulike vanskelighetsgrad da og utfordre de. Plutselig så har vi jo da noen som kan litt orientering og som er veldig fysisk sterke som lett kan løpe 60 minutter uten å slite med det, mens de som er svakest fysisk orker kanskje ikke løpe mer en 10 minutter og kanskje kan gå i 30 minutter ikke sant. Så det blir ihvertfall viktig for oss.

Ja eventuelt folk som har funksjonsnedsettelser

Ja det er jo noen som har innimellom og det er jo et veldig godt poeng.

Hvor ofte er det dere har orientering?

Vi har jo det alt for sjeldent, så hvis vi har det en gang i løpet av tre år, altså at en klasse får ha det en gang på mellomtrinnet så er jo det nærme. Også går vi jo en del på tur da, men da har vi jo ikke orientering, men da snakker vi jo litt om stier, kart og sånne ting og vi har jo om kart og målestokk og sånne ting.

Varierer dere løpyene fra år til år?
Vi har kun de samme kartene tilgjengelig, så ungene vil ofte løpe den samme løypen i 5 klasse som dem gjør i 7ende.

Hvor henter dere kart fra?

Vi har papirkart som står i en gammel kassett og de karta er vel 40 år gamle.

Og det er bare det dere bruker nå?

Ja. Noen av lærerne har vært flinkere enn andre på å bruke kart og bruke dem til orientering. Så de bruker det innimellom.

Hvordan markerer dere da? er det bare ferdig markerte?

Ja det er litt begge deler. Noen er ferdig markert, men der kommer problemet alt er jo annerledes. Oppe i skauen her er det en del nye stier, som er ganske store, så da mister dem(karta) jo helt verdien.

Ja de er ikke oppdaterte

Ja

Utfordringer med å få tilgang? Ja det er jo bare de dere bruker som dere har her

mhm

Har dere lyst til å bli med når vi tester og når vi er ferdige (prototype)?

Ja vi er med på det :) Det er bare å ta kontakt. Det er bare gøy.

D - Project description

January 18, 2018

Map Management System for

Orientation Sport

Group Number:
BO18-G34

Group Members:
Michael Mebrhatu Simon
Adis Jasaravic
Jonas Vestgaden

Project title: FinnFram i Østfold – Elevorientering med klasse- og skolekonkurranser.	**Date:** 18.01.2018 **Version nr.** 0.1 **Pages:** 12 **Appendix:** n/a
Document type: Project Description	
Author(s): Michael Mebrhatu Simon Jonas Vestgarden Adis Jasarevic	**Approved By:** Michael M. Simon. Project Leader
Department: HIØF, Faculty of computer science **Program:** Information Systems **Course**: Bachelor thesis	
Mentor: Børre Stenset	

Extract:
This document contains the project description of the FinnFram i Østfold project.
The document is prepared with the purpose of introducing the project goal, plan and implementation methods. The contents of the document may alter in the future.

Project proposed by: Østfold orienteringskrets	

Table of contents

Introduction

Background

This document is coordinated based on the project proposed by Østfold orienteringskrets. Østfold orienteringskrets wishes to have a system that can be used to share orientation maps among teachers and students.

The applications requirements are to implement functionality that enables teachers to import/upload a premade map using third party applications, mark the spots were posts are located, save the map in the system, and share the maps with others.

Teachers and students will also download or print the map to use it while participating on orientation activities.

Orientation Sport

Orienteering is the sport of navigation with map and compass. It's easy to learn, but always challenging. The object is to run, walk, ski, or mountain bike to a series of points shown on the map, choosing routes, both on and off trail, that will help you find all the required points and get back to the finish in the shortest amount of time.

The points on a course are marked with orange and white flags and pouches or electronic devices, so you can prove you've been there. Each "control" marker is located on a distinct feature, such as a stream junction or the top of a knoll (Orienteering, 2018).

The sport demands the students ability to read maps and compass, teamwork and a coordination of physical and mental activity when performing.

In schools the sport is used an instrument for learning use of compass and map reading.The activity helps increasing social activities and enhance interaction between students.

Team

Name: Jonas Vestgarden | Birth: 1993 | Course: Informasjonssystemer | Location: Halden

---

Is a third year information systems student. Interests are programming, modelling and consultation. After a completed bachelor's degree, he will pursue being a consultant or a game developer, if not at the very least try both fields

Name: Adis Jasarevic | Birth: 1995 | Course: Informasjonssystemer | Location: Sarpsborg

---

Third year information system student at Halden University College. His interests are programming languages, database design, 3d modeling programs (maya and inventor) and as well as some marketing and music production. After a finished bachelor, he wants to specialize himself in machine learning and therefore in the future after the master's degree collaborate with his brother.

Name: Michael M. Simon | Birth: 1994 | Course: Informasjonssystemer | Location: Halden

---

Michael is an Information systems student. He's interested in user experience, Human computer interaction (HCI), Front-End developing.
He's currently managing his own private business which mainly deliver service in graphic design for print and digital. He wishes to further develop his business with a variety of services in web technology and continue to study on a master's degree at the University of Oslo and specialize in Interaction design.

Project Owner

ØO(Østfolds orienteringskrets) is an active association hosting multiple official runs and training runs each year. They are a branch of the Norwegian orientation association and were founded in 1940 (Jens Erik Mjølnerød, 2016, s.7). They have an

active and engaging community, though they wish for more young people to join them, hence this project was created (Norsk Orientering, 2018).

Project Scope

Deliverables

The finished product will be a website for teachers on all schools in Østfold. This website will allow the teachers to have a common place to view and share orientation maps for areas in Østfold. The maps will be saved on the same database as the system should be available for every teacher.

They can upload, share, draw (set annotations), print and retrieve saved maps.
They can automatically generate annotations on the maps.
They can edit individual annotations on maps already saved in the system by other teachers.
Additionally the following functionality will be present in the webapp:

- Register user
- Login and Logout
- Save the result
- Share the result in the application
- Import maps that are made using other map drawing application, for example OCAD.

There might also be a light-weight mobile app version to show saved orientation maps so teachers can check their map while outside in the environment.

Goal

The main goal of the project is to make a system that can be used to upload, draw and share Orienteering sport maps among teachers and schools.

ns.Subgoal 1

Create a website application that can be used to store maps, download and print them. The application shall also store the maps in groups based on locatio

Subgoal 2

Add additional functionality allowing for editing of existing maps, placing markers on a fresh map, and share the maps with other Teachers.

Subgoal 3

Implement a light-weight version of the webapp for mobile,The application can

Will be used by teacher during orientation with the students. .

Method

During this project we are fortunate to have proper guidance by people with great knowledge in their respective fields. As this project has a lot to do with orientation, we brought in an orienteering expert to consult us on what this really entails. For the technical parts we will be using our assigned mentor, lecturer with the required knowledge and integrate it with the research the group perform. We plan to hold

regular meetings with our project owner to ascertain that our idea of the project contributes to them realising their vision.

The project given by ØO(Østfold Orienteringskrets) is open for creative solutions. Besides the underlined requirements, which is a platform for sharing orienteering maps, ØO is open for additional functionalities that makes it interesting for the teaching process.

The openness of the solution makes the project special. Continuous confirmations of deliveries and involvement of the ØO is vital for the project to deliver effective results. Based on the mentioned facts, the student group decided that a scrum methodology and design thinking process to be the core of the the general developing process.

Scrum
Scrum is a framework of a development process that can be used to deliver a software product.
Sprints are results that the project team produces in scrum, they are small fraction of the overall deliveries that are shipped in a one to two weeks of an interval and iterative development process. Scrum developing suits a team of three to nine persons, and the team members are delegated certain task focusing on three categories. These categories are scrum master, product owner and development team. Scrum increases the effectivity of project processes and development activities, but it may cause overlapping of roles in a team with a few members as ours. The team decided to implement the suitable elements of scrum and incorporate with projects tasks and processes.

Design Thinking
Design thinking is a user oriented design and development methodology that can be applied when developing technology and solving problems. The design thinking process contains five-stages; empathise, define, ideate, prototype and test, the team iterate through these stages when discovering and denoting the users pain points.

This project is open for creative ideas that are expected to contribute making the orientation activity interesting for teachers and students. Using the design thinking methodology the team will attempt to discover, analyse, ideate, prototype and test the suggested solutions and try to resolve the problem.

We have seen that a powerful methodology for innovation has emerged. It integrates human, business, and technological factors in problem forming, -solving, and -design: "Design Thinking." Its human-centric methodology integrates expertise design, social science, engineering, and business. It blends an end-user focus with multidisciplinary collaboration and iterative improvement to produce innovative products, systems, and services.
(Plattner, Meinel, & Leifer., 2010, s.14)

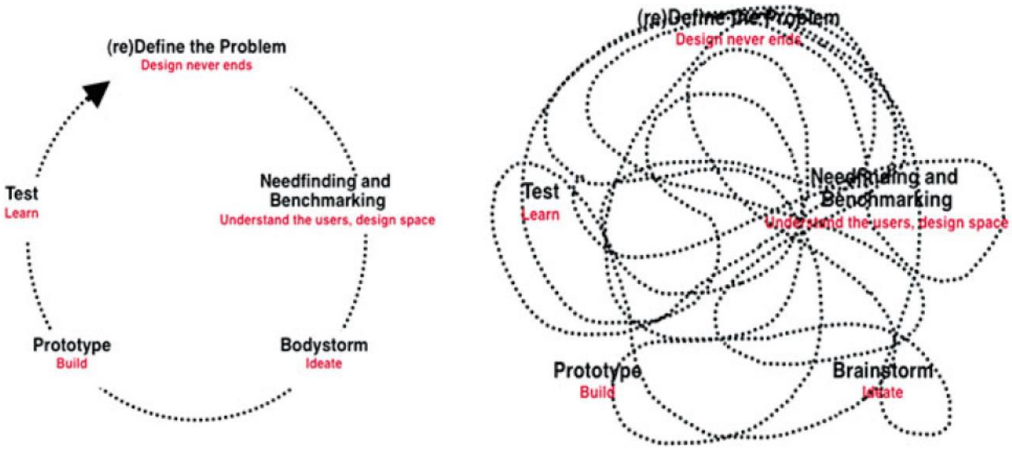


Figure 1. Illustration of standard and approximate look of the process of design thinking in real world. (*Design thinking: understand–improve–apply.*, 2010)

By Integrating the design thinking process with agile methodology the team will be equipped with the tools that helps managing the project and deliver user-centric solutions. The figure above (figure 1.) illustrates the iterative process of the design thinking approach and the picture on the left illustrates the standard form of design thinking.

Project Development Phases

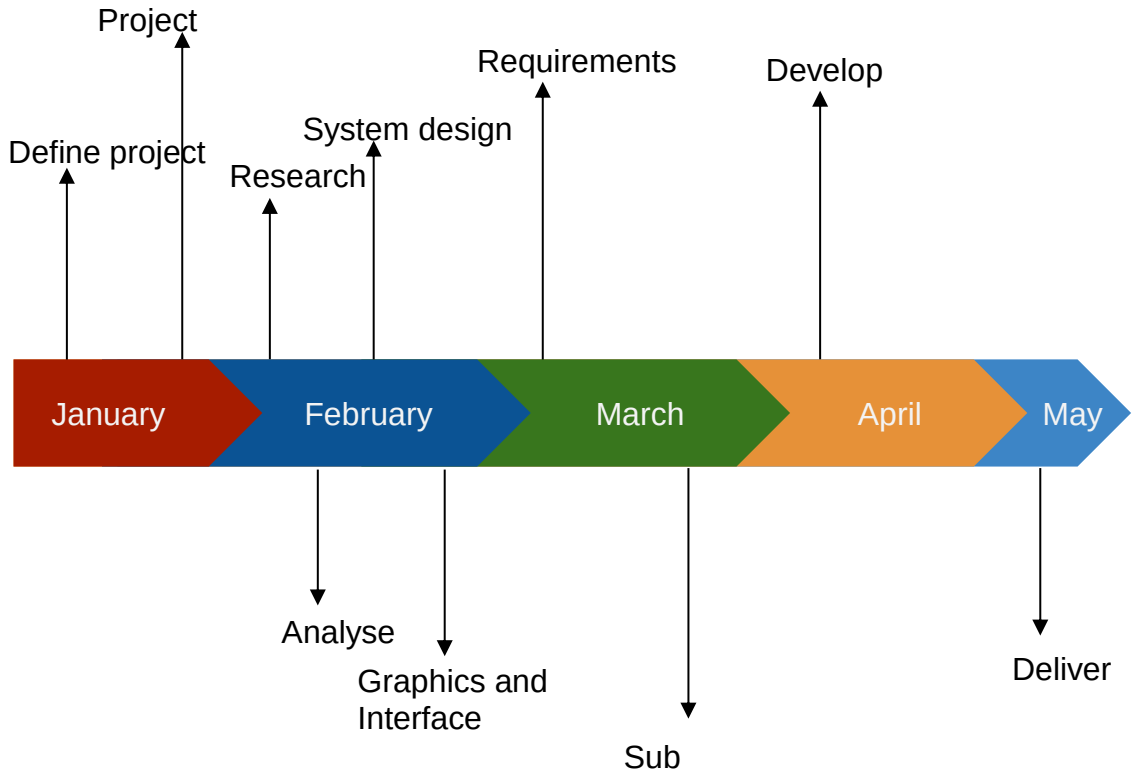


Figure 2. A framework of the overall project phases.

Activity 1: Gathering information (research)

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: Processable information.
- Description: In this activity there should be focus on obtaining information for the project. We will find important sources that will help us to initiate the idea and help us move further to the development phase.

Activity 2: Defining the project

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: The grounds upon which we build the project
- Description: It's important that we define exactly what we are going to be developing for the project owner. Once defined we know the core of what we are creating, with room for innovation and modification if agreed upon.

Activity 3: Planning

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: The different planning phases before the development
- Description: Plan the phases in the project and

Activity 4: Analysis of the research material

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: Processed data of the research.
- Description: We will be analysing the research data we retrieve to work out the pros and cons of different solutions, what they entail to implement and figure out the best course of action / solution for the project.

Activity 5: Drafting a requirements specification

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: The requirements specification
- Description: The requirements specification of the solution.

Activity 6: Graphical interface wireframes

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: Wireframes of the graphical interface of the solution
- Description: This will be a wireframe of how we plan the graphical interface of the webapp solution is going to look like.

Activity 7: Functionality wireframes

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: Wireframes of the webapp's functionality
- Description: We are creating a wireframe of how we intend the functionality of the
- webapp to behave when interacted with.

Activity 8: Evaluation of the wireframes.

- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon

- Delivery: The result of the evaluation
- Description: Evaluate the wireframes and see if it meets all requirements in the requirement specification.

Activity 9: Wireframe Adjustment
- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: Updated wireframe
- Description: Change the wireframe to match the requirement specification.

Activity 10: Development
- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: A prototype of the webapp solution
- Description: The web application itself. This will be the result of iteratively developing the solution throughout the project period.

Activity 11: Testing
- Staffing: Jonas Vestgarden, Adis Jasarevic, Michael Mebrathu Simon
- Delivery: Testing data of the webapp to process and analyse
- Description: The result of the testing will determine if any changes needs to be made.The solution might change multiple times depending on the analysis made here.

Project Implementation

The process behind the project shall include weekly meetings with the representative of "Østfold Orienteringskrets" and the members.

TEAM MEETINGS

The group will have 4 weekly meetings each week, where discussions will be held about the plans for the project and the solutions. All members should meet up at the meetings.

If one of the members does not meet up, then that person must report it to the project manager in the group. People who do not meet up and do not report the mediation may risk being thrown out. Sick leave should also be reported to the group.

In the meetings, the members can discuss and reflect on questions together with the supervisor. The meetings are here to improve and help the members get on the right path and as well get answers to relevant questions regarding the project.

GROUP

The group have a project manager and that position will be rolled every month. Right now, the project manager is Michael Mebrhatu Simon. The group consist of 3 members where everyone shares the same education, Information Systems (IS).

METHOD

The project's development will have an iterative development process using repetition. Version control will be handled through github.

Sources

Plattner, H., Meinel, C., & Leifer, L. (Eds.). (2010). *Design thinking: understand–improve–apply*. Springer Science & Business Media.

Jens Erik Mjølnerød. (2016). *Østfold O-krets Årsberetning.* Hentet fra http://www.orientering.no/media/filer_public/6c/cd/6ccd8bdd-cbd2-4aa5-bcdf-7e78adb40f55/arsberetning_2016_oppdatert.pdf.

Orienteering. (2018). *About orienteering*. Hentet fra http://www.orienteering.ca/about/orienteering/.

Norsk Orientering. (2018). *Østfold*. Hentet fra http://www.orientering.no/kretser/ostfold/.

E - Team timetables

Timetable for Jonas Vestgarden

Timeframe	01/01/2018 - 05/14/2018		
Total duration	394:11:04		
Date	**Team member**	**Duration**	**Note**
01/09/2018	jonas	6:35:00	Arbeidsmøte - Avklare hva vi lurer på angående oppgaven og danne spørsmål for møtet med oppdragsgiver i morgen.
01/10/2018	jonas	7:12:00	Møte med Børre
01/12/2018	jonas	3:47:47	Møte med oppdragsgiver
01/17/2018	jonas	4:49:00	Veiledningsmøte med Børre og arbeid etterpå
01/18/2018	jonas	5:38:24	Fullføring av prosjektbeskrivelsen
01/21/2018	jonas	3:52:00	Research av teknologi og testing av svg som tegneformat
01/23/2018	jonas	3:32:00	Jobba videre med funksjonalitetstesting av svg hjemme
01/24/2018	jonas	2:33:38	Møte m børre
01/25/2018	jonas	7:46:00	Arbeidsmøte. Testet med diverse løsninger vi kan benytte
01/26/2018	jonas	3:42:00	Jobba videre med funksjonalitet på tegnefunksjonen
01/30/2018	jonas	8:24:00	Jobbet med hoveddokument første versjon og litt mer funksjonalitetstest av svg
01/31/2018	jonas	5:53:00	Jobbet med hoveddokument og funksjonalitet av tegnefunksjon
02/06/2018	jonas	4:15:00	Design thinking
02/06/2018	jonas	0:00:01	Design thinking
02/07/2018	jonas	6:32:15	Møte med Børre, intervju med lærere
02/13/2018	jonas	6:16:00	Dratt til 3 skoler i Halden og avtalt møte med 1 skole og fått kontaktinformasjon fra de andre
02/21/2018	jonas	6:58:00	Worked on titles in the main document v1
02/28/2018	jonas	7:14:00	Went through the requirements specification with Børre, the group and Svend Sondre
03/01/2018	jonas	6:27:00	Researched MEAN (MongoDB, Express, AngularJS, NodeJS) and different nodeJS CRM solutions
03/05/2018	jonas	4:24:00	Set up and tested a NodeJS web app, and did some more research on the topic
03/06/2018	jonas	7:02:00	Had a meeting with the projectowner. Discussed the way forward, wordpress vs nodejs and testing for the future. Continued working on the document.
03/08/2018	jonas	5:16:32	Planlegging, jobbing og intervju med lærer på Låby skole
03/09/2018	jonas	7:35:00	Fullført første versjon av hoveddokumentet
03/13/2018	jonas	5:23:00	Planlegging for div funksjonalitet for database, fieldtypes etc.
03/14/2018	jonas	5:12:00	Møte med børre, intervju med hjortsberg lærer, skrevet use case
03/20/2018	jonas	8:08:38	Jobbet med å sette opp nodejs rammeverket I mean stack med mike
03/21/2018	jonas	9:29:00	Jobbet videre med tegnefunksjonen
03/22/2018	jonas	7:58:00	Jobbet mer med angular rammeverket
03/27/2018	jonas	7:42:00	Fullføring av tegnefunksjon i javascript. Nå må den overføres til angular.
03/30/2018	jonas	6:08:00	Mesteparten av funksjonalitet fungerer nå på angular

04/03/2018	jonas	5:36:00	Jobbet videre med rammeverket og back-end kode. Fikset og satte opp en felles repo på github som funker for alle på gruppa.
04/04/2018	jonas	8:54:00	Møte med Børre, Oppdatering for Svend Sondre og videre arbeid med dokumentet og analyse kapitellet.
04/05/2018	jonas	4:52:00	Satt opp mongodb for testing i appen. Et enkelt api post interface er lagt til for users
04/06/2018	jonas	8:10:00	Optimalisert kartfunksjon kode betraktlig (endret fra å update i realtime til slutten av en event) Dette fikset også mobil touchmove event slik at appen kan brukes på touchdevices
04/09/2018	jonas	4:49:00	Optimalisert videre på kartfunksjonalitetskoden
04/11/2018	jonas	1:37:00	Oppdatert trello for å håndtere workflow litt bedre
04/12/2018	jonas	7:21:00	Jobbet med hoveddokumentet
04/17/2018	jonas	6:29:00	Skrevet på hoveddokumentet
04/18/2018	jonas	6:53:00	Skrevet på hoveddokumentet
04/19/2018	jonas	2:25:00	Mer arbeid på dokumentet
04/19/2018	jonas	3:54:00	Rettskriving og div
04/20/2018	jonas	6:18:00	Fikset på punkter, lagt til informasjon, placeholder tekst osv i tegneverktøyet
04/24/2018	jonas	3:14:00	Hoveddokument
04/25/2018	jonas	5:32:00	Hoveddokument arbeid
04/26/2018	jonas	8:09:00	Hoveddokument
04/27/2018	jonas	7:19:00	Hoveddokument
04/30/2018	jonas	4:19:00	Documentation of edit map in main document. New functionality in edit map
05/01/2018	jonas	1:30:00	Jobbet med dokumentet.
05/02/2018	jonas	9:22:00	Kodet i appen. Laget hele map upload komponentet
05/03/2018	jonas	12:12:00	Laget løypelagring funksjonalitet. Man kan lagre, og hente løyper fra databaser basert på kartet brukeren er på.
05/04/2018	jonas	12:20:00	Home page, edit-map, upload map, save/edit/print er nå ferdig. Optimalisering og bugfixes gjenstår.
05/05/2018	jonas	13:38:00	Fixed a lot of bugs and issues with the app overall
05/07/2018	jonas	4:32:00	Fikset mer issues med appen
05/08/2018	jonas	9:06:00	Set up the node app on a linux server. It can be visited if you have a vpn connection
05/09/2018	jonas	8:36:00	Rettskrevet dokumentet så langt
05/10/2018	jonas	7:45:00	Rettet opp i dokumentet og skrevet server tutorial
05/11/2018	jonas	5:37:00	Møte med Børre og Svend Sondre, jobbet med dokumentet og appen
05/12/2018	jonas	11:15:00	Fikset flere bugs i appen. Kan leses mer om på github pushes
05/12/2018	jonas	3:03:00	Writing discussion
05/13/2018	jonas	14:05:49	Finished writing discussion and conclusion, formatted text, added appendixes. This is my last entry, I will miss you little handy application. Bye.
05/14/2018	Jonas	9.33	Formatering og rettskriving I dokumentet. Test printet og sjekket at sideoppsett er riktig.

Timetable for Adis Jasarevic

Adis Jasarevic			Sum timer	216,7175
Timelise Bacehlor				
Dato	Kl(Fra og til)	Beskrivelse	Timer	
1/9/2018		Arbeidsmøte med gutta	6,35	
		Møte med oppdragsgiver	3,5	
1/17/2018	09.00 - 14.00	Møte med børre og arbeid med prosjektbeskrivelse.	5	
1/18/2018	11.43 - 14.50	Projektbeskrivelse	2,57	
1/24/2018	09.00 - 11.00	Møte med børre	2	
1/30/2018	21.28 - 23.00	Arbeid med hoveddokument.	1,32	
1/31/2018	09.45 - 14.00	Arbeid med hoveddokument og prøving av bibliotekverktøy som react.	4,15	
2/1/2018	11.00 - 14.00	Arbeid med hoveddokument, diskutere om hvilken type metode vi skal bruke for å hente inn data, blant annet intervju og kvalitativ metode.	3	
2/7/2018	09.15 - 14.00	Møte med børre og intervju med lærere	4,45	
2/13/2018	10.00 - 14.00	Skole besøk. 3 skoler ble besøkt.	4	
2/18/2018	17.00 - 02.26	Funnet informasjon om på hvordan man klarer å bruke microsoft sql server med php og fikk testet ut sql managment verktøy. Lært rundt hvordan man benytter det. Det virker som jeg må teste ting videre hvis vi skal bruke microsoft server istede for wamp og mysql. For å få dette til å fungere må vi lage server hjemme	9,26	
2/20/2018	10.00 - 14.00	Snakket om design-elementer og database	4	
2/28/2018	09.15 - 14.47	Hatt møte med børre med resten av gruppen, samt jobbet med prosjektet og funnet ut andre type løsninger som kan være gode for opensource platform. Vi har også hatt en samtale med dagligleder gjennom telefon pga informasjon vedlagt i kravspesifikasjon. Snakket om prioriteringer og punkter vi kanskje ikke rekker før mai.	5,32	
3/2/2018	11.00 - 12.00	Lett på nettet angående opensource og database. Hva slags database teknologi har opensource managmentsystem og hvilke passer best for vår løsning.	1	
3/6/2018	10.00 - 14.34	Skype møte med Svend Sondre og jobbet videre med dokumentet	4,34	
3/8/2018	11.45 - 16.45	Jobbet med dokumentet og hatt undersøkelse på barneskolen og tatt optakk av samtalen vi hadde med læreren.	5	
3/9/2018	11.45 - 14.45	Jobbet med førsteversjon av dokumentet. Fyllt inn det som trengtes for innlevering.	3	
3/13/2018	11.25 - 15.00	Jobbet med design og wireframes. Vi opp hvordan ting skal fungere f.eks. knapper, log in og log ut og andre funksjonalitet.	3,35	
3/14/2018	09.15 - 11.55	Hatt møte med børre og snakket om første versjon av dokumentet. Vi har også hatt møte med læreren "Jonas" fra Hjortsberg barneskole i halden. Sammens med barneskole læreren så har vi stilt spørsmål og hatt en undersøkelse og samt tatt optakk av samtalen.	2,4	
3/15/2018	18.35 - 19.35	Skrev ned intervju/undersøkelse 2 til tekst dokument. Laget møte referat vi glemte å gjøre på onsdag	1	
3/16/2018	11.00 - 16.00	Jobbet med selve utviklingsplanen. Funnet ut hva vi trenger av utviklingsverktøy. Som .f.eks. handlebars.js eller pug.js	5	
3/18/2018	11.22 - 03.25	Jobbet med koden individuelt. Fått lastet ned mongodb og mongoose. Tuklet litt med funkjsoner i cmd. Lagt inn data til en database som jeg lagde selv. Sett på diverse videor koblet til mongodb og hvordan det skal brukes.	4,03	
3/20/2018	11.00 - 13.25	Jobbet med koden og sjekket hvilken alternativ var bedre pug.js eller handlebar.js	2,25	
4/3/2018	11.00 - 16.30	Jobbet med selve koden	5,3	
04.04/2018	24.00 - 03.00 og 11.00 - 16.30	Lett etter forskjellige test-editors og lest litt rundt på hvem som er bra og hvem som ikke er bra.	8,3	

4/11/2018	09.15 - 12.00	Møte med børre og bygget videre på dokumentet. Analyse delen.	2,45	
13.04.2018	21.28 - 03.24	Jobbet videre med dokumentet overallprocess og etc	5,56	
4/17/2018	11.00 - 16.00	Jobbet med dokumentet og text-editor	5	
4/18/2018	09.00 - 16.00	Hatt møte med børre og jobbet med dokumentet	7	
4/19/2018	12.00 - 15.00	Jobbet med dokumentet. Requirement specification	3	
4/21/2018	18.35 - 03.34	Jobbet med koden (Database)	8,59	
4/25/2018	09.00 - 13.02	Møte med børre og diskutert rapporten(implementasjonsdelen og server delen), samt jobbet med dokumentet.	4,02	
4/26/2018	12.00 - 20.10	Jobbet med dokumentet med gruppen. Skrevet test kapittel delen og justert rundt omkring i rapporten. Jobbet med kilder.	8,1	
4/27/2018	10.53 - 17.16 og 19.44 - 03.00	Hovedokument og koden til text-editor.	13,39	
4/30/2018	15.50 - 22.00	Jobbet med funksjonaliteten til applikasjonen.	6,1	
5/3/2018	19.30 - 06.00	Laget kode for backend når det gjelder lagring av informasjon fra admin editor til databasen. Initialiserte Quill gjennom js-kode for testing og lagring av delta objekter. Så samt på forskjellige videor crud operation videos.	10:30	
5/4/2018	21.30 - 09.15	Jobbet med koden sammens med gruppen. Fått koden til å lagre delta objekter til databasen gjennom angular komponent og fått den til å hente informasjonen fra databasen.	11,45	
5/6/2018	18.30 - 03.45	Jobbet med koden og design elementer.	9,55	
5/7/2018	20.04 - 22.19	Skrev litt på dokumentasjon html og css	2,15	
5/8/2018	13.33 - 14.36 og 18.15 - 20.08	Skrev på dokumentet (database dokumentasjon og backend dokumentasjon)	3,23	
5/9/2018	09.15 - 17.24	Møte med børre og rettet dokumentet med gruppen	8,09	
5/10/2018	18.03 - 21.12	Jobbet med dokumentet	3,09	
5/11/2018	13.01 - 14.07 og 19.38 - 20.16	Jobbet med dokumentet og lagd forside design.	1,44	
5/12/2018	01.00 - 03.06 og 03.45 - 04.22	Rettet på dokumentet og lagt til kommentarer her og der	2,43	
5/13/2018	12.41 - 22.26	Jobbet med dokumentet og rettet.	9,45	
5/14/2018	11.00 - 13.30	Jobbet med dokumentet og byttet forside-design.	2,3	

Timetable for Michael M. Simon

Dato	Timer	Kommentar
04.01.2018	2	laget hjemmeside til gruppen,koble til server
05.01.2018	1	Ordnet mail og kontakt personer
09.01.2018	3	Møte med Børre og arbeid etter det
10.01.2018	4	Forprosjekt rapport template
11.01.2018	3	møte og arbeid med forprosjektreapporten
12.01.2018	3.5	Møte med oppdragsgiver
15.01.2018	2	Skrivekurs på skolen
16.01.2018	1	Orienterings delen, research og Jobbet med pb sammen med adis og jonas. Jeg tok metode delen og undersøkt flere metoder som finnes og hviken metode vi bruker
17.01.2018	4	Møte med Børre og jobbet
18.01.2018	5	Gruppe arbeid
23.01.2018	6	Ukentlig møte med veilder og har jobbet med å finne teknlogier, valg av programming språk
24.01.2018	6	
30.01.2018	5	Har et møte med Jens Erik, snakket om fremdrift av prosjektet og jobbet med doukumentet etter det
31.01.2018	7	Ukentlig møte med børre, Arbeid med hjemmeside etter det
01.02.2018	5	rsearch hvordan man utfører forskjellige type unndersøkelser, laget målet, metode og targetgroup av undersøkelsen
06.02.2018	2	Ukentlig møte
07.02.2018	1	Interface og design elements
08.02.2018	7	Interface plan and design elements
13.02.2018	6	Skole Besøk
17.02.2018	5	UI design research, learn and find out about Material design
21.02.2018	3	Document template
22.02.2018	4	Gruppe arbeid
24.02.2018	3	Gruppe arbeid
27.02.2018	4	Conceptual desing and ui
28.02.2018	3	UI design
01.03.2018	4	Dev enviroment research and testing it
06.03.2018	4	
07.03.2018	3	
08.03.2018	3	Interview with the schools
14.03.2018	5	Møte med skole
15.03.2018	3	Nav bar build
20.03.2018	4	form/register/db register

21.03.2018	6	Token design and development
22.03.2018	7	confirmation bug and document
23.03.2018	5	User data retrieve
27.03.2018	4	Developmentn
28.03.2018	6	Code documentation and development
29.03.2018	3	Documentation
30.03.2018	4	Coding
04.04.2018	4	Coding
05.04.2018	8	Coding
07.04.2018	6	Coding
10.04.2018	4	Coding
11.04.2018	7	Document
12.04.2018	5	Document
13.04.2018	5	Document
10.04.2018	5	Coding
18.04.2018	8	Document writting
19.04.2018	7	rxjs fix and doc
20.04.2018	4	Coding
23.04.2018	9	Coding
01.05.2018	12	Coding
02.05.2018	12	Coding
03.05.2018	13.5	Coding and document
04.05.2018	3	Coding
08.05.2018	5	Document writitng
09.05.2018	5	Documentation
10.05.2018	3	Coding
11.05.2018	7	Document Writitng
12.05.2018	8	Document Writting
13.05.2018	8	Document writting
14.05.2018	8	Document final fix
Total	**313**	

My Contribution

I have been leading the group for a while, taking care of contacts with stakeholders and setting meetings.

For that reason, I haven't limited myself from contributing in many other aspects of the application both in the document and developing.
To be precise these are the main parts I have contributed with
1. Setting up the initial mean stack environment
2. Setting up database
3. Building schemas for user, article and map
4. Database configuration
5. UI mockup design
6. navigators and routes
7. login, logout, validation
8. token
9. Initial server-side configuration with express and NodeJS
10. Authentications and validations
11. The user interface partially
12. Guard for controlling users access

Contribution with the team
1. Content retrieve data
2. news and activity display page

Document
Very hard to say a specific part because everyone has been free to fix, suggest overall
Generally, 30 up to 35 % contribution

Researches and administrative
1. Understanding mean stack
2. Learn Typescript
3. Learn MongoDB
4. Researched alternative solutions PHP and other JS CMS libraries
5. Attended all meetings with stakeholders
6. Attended meetings with thesis advisor (only 2 absence)
7. Attended interviews

F - Group contract

Prosjektkontrakt

Bacheloroppgaven Avdeling for Informasjonsteknologi

Høgskolen i Østfold

Gruppe BO18-G34:
Jonas Kristoffer Vestgarden
Michael Mebrhatu Simon
Adis Jasarevic

Oppdragsgiver:
Østfold Orienteringskrets

Kontakpersoner:
Jens Erik Mjølnerød
Per Bergerud
Elsie Brenne

Veileder:
Børre Stenseth

19. desember 2017

Prosjektbeskrivelse
Formålet med prosjekt FinnFram i Østfold er å mobilisere for fysisk aktivitet og bruk av kart blant barn og unge, samt å synliggjøre VM i orientering i 2019.
Resultatet av prosjektet er et system design og prototype for elektroniske skolekart og skolegårdskart for barne- og ungdomsskoler i Østfold.

Prosjektet skal utføres av studentene i samarbeid med oppdragsgiver og veileder. Studentene skal jobbe med system design og utvikling av løsningen i løpet av semesteret.
Dette innebærer å lage detaljert prosjektbeskrivelse som skal bekreftes av oppdragsgiver, system design dokument samt en prototype.

Studentgruppen
Studentgruppen har selv det fulle ansvar når det gjelder organisering og gjennomføring av prosjektet. Ikke minst innebærer dette å følge opp tidsfrister. Kontakt med oppdragsgiver og veileder er også studentenes ansvar. De må også holde seg oppdatert på beskjeder og meldinger fra fagansvarlig (som regel pr. epost). Studentene skal selv fordele arbeidet i gruppa, og eventuelt velge prosjektleder (kan byttes pa underveis hvis ønskelig).

Gruppen har rett til veiledning fra en eller flere av HiØ/IT sine fagansatte tilsvarende ca. en halv time pr. uke gjennom hele semesteret. Studentene skal også ha fri tilgang til program- og maskinvare og annet nødvendig utstyr. Er det behov for reiser og lignende, skal dette dekkes etter avtale med oppdragsgiver og/eller HiØ/IT.

Oppdragsgiver
Det er viktig at oppdragsgiver er innforstått med rammene rundt bachelor-oppgaven. Oppdragsgiver må ikke definere en oppgave som ellers ville bli utført av eksterne konsulenter o.l. Det vil alltid være en risiko for at studentene ikke kommer helt i mål, eller leverer noe som ikke nødvendigvis er ferdige, operative løsninger.

Oppdragsgiver skal gi rammene for prosjektet, dvs. hjelpe til med å definere formål, leveranser, og evt. metode. Det er svært viktig å gi studentene nok informasjon, og evt. tilgang til utstyr, programvare o.l.

Oppdragsgiver må være forberedt på å svare på spørsmål på kort varsel, og sørge for å være oppdatert på prosjektets framdrift. Det kan også være aktuelt å hjelpe til med å revidere og endre prosjektplanen underveis.

Det er vanlig å ha regelmessige møter med gruppa, med en frekvens som passer prosjektets egenart.

Veileder
Veileder skal hjelpe til med: 1) Gjennomføring og dokumentasjon, og 2) Faglige spørsmål (i de tilfellene der det faglige blir for smalt og spesielt, må studentene enten skaffe faglig hjelp fra oppdragsgiver, eller fra annet hold).

Rettigheter
Deltagere med rettigheter i prosjektarbeidet er ideforfatter, utøvende medarbeider-
e og oppdragsgiver.
Ideforfatter kan være veileder, en eller flere studenter, oppdragsgiver eller en kombinasjon av disse.
Utøvende medarbeidere er studentene og i noen grad veileder. Oppdragsgiver kan enten være en ekstern institusjon/bedrift eller en avdeling ved Høgskolen i Østfold.

Følgende rettigheter er knyttet til prosjektdeltagelse
Opphavsrett Rett til benevnelse ved publikasjon og eiendomsrett til originalt åndsverk (som sikres ved publikasjon). Opphavsrett innehas av ideforfatter og ut vende medarbeidere.

Disposisjonsrett Rett til videreutvikling av et avsluttet arbeid. Denne innehas av oppdragsgiver.
Eiendomsrett Rett til oppbevaring og omsetning av et avsluttet arbeid og innehas av oppdragsgiver.

Avvik fra disse prinsippene kan avtales før godkjenning av prosjekter etter ønsker fra oppdragsgiver.
Det skal da lages en avtale som inngås av alle deltagerne i prosjektet. Tvister om rettigheter avgjøres etter deltakernes samtykke av avdelingsstyret, subsidiært ved forfølgelse i det alminnelige rettsvesen.
Hvis oppdragsgiver ikke ønsker at resultatene eller deler av disse skal offentliggjør-
es, må dette avklares ved oppstart av prosjektet.

Underskrifter

Jens Erik Mjølnerød Per Bergerud Elsie Brenne

Jonas Kristoffer Vestgarden Michael Mebrhatu Simon Adis Jasarevic

Børre Stenseth

E - Requirements specification from ØOA

Kravspesifikasjon

Dette dokumentet inneholder en kravspesifikasjon for en kartplattform som utvikles i forbindelse med prosjektet «FinnFram i Østfold» Med denne kravspesifikasjonen ønsker Østfold orienteringskrets å spesifisere hva det er ønskelig at en slik kartplattform burde inneholde.

1. Bakgrunn og formål

Formålet med prosjekt «FinnFram i Østfold» er å mobilisere for fysisk aktivitet med kart blant barn og unge, samt å synliggjøre orienteringssporten og markedsføre WOC2019 (VM i orientering) for Østfolds befolkning. Økt fysisk aktivitet blant barn og unge er også et viktig mål med prosjektet som arbeider med utgangspunkt i Norges idrettsforbund sin visjon «Idrettsglede for alle».

Prosjektet vil arbeide for å nå en rekke resultatmål. Dette innebærer blant annet å tegne skolegårdskart for alle skoler i Østfold, bidra til å øket kompetanse i skolen omhandlende bruken av kart og kompass, utvikle god og bærekraftig praksis for samarbeid mellom orienteringsklubber og skoler, samt å utvikle skolearrangement/konkurranser basert på orientering. Med dette håper vi på en økt rekruttering til orienteringssporten i Østfold.

En kartplattform utviklet i samarbeid med studenter fra Høgskolen i Østfold. Denne vil få en sentral betydning i prosjektet. Ønsket er at plattformen som utvikles er såpass sikker og enkel å bruke at aktivitetslærere kan finne alt av informasjon de skulle behøve for å gjennomføre god orienteringsundervisning i skolen på ulike nivåer. Dette innebærer blant annet funksjoner som: en kartbase med orienteringskart av ulik karakter, en enkel løypeleggerfunksjon (tegnefunksjon), aktivitetsforslag til undervisningen, enkle kartredigeringsmuligheter.

2. Prosjektets mål

Prosjektet skal anskaffe og implementere en ny portalløsning med følgende kjennetegn:

- **Enkelt og effektivt**

  Det skal være enkelt å finne frem, både via søk og ved navigasjon. Klart språk skal gi god forståelse.

- **Brukervennlig og intuitivt**

  Bruken skal oppleve nettstedet som intuitivt og få en god opplevelse. Nettstedet skal være i tråd med kravene til universell utforming.

- **Tilpasset nettbrett og mobil**

Innholdet på nettsiden skal være godt tilgjengelig uansett hvilken plattform brukeren velger.

- **Grunnlag for digitale tjenester**

  Å kunne velge digitale tjenester er lettvint og effektivt. Nettstedet skal utformes som en god plattform med sikte på dagens og fremtidens digitale løsninger.

3 Rammebetingelser

Tjenestene i portalen skal forholde seg til og oppfylle krav i gjeldende lov- og regelverk, bl.a.

- Lov om personopplysninger

- Personopplysningsforskriften
- DIFI's kvalitetskrav

3 Anskaffelsesprosessen

Vi ønsker å gjennomføre utviklingen av nettportalen som en dialogprosess mellom utvikler og oppdragsgiver.
I denne fasen avtales jevnlige møter mellom utvikler og oppdragsgiver. Det er også ønskelig at løsningen gjennomgår en testfase før den settes i produksjon.

4 Krav til plattform og verktøy

Det er ønskelig at utvikleren bruker en CMS plattform basert på «open source», eks. Wordpress e.l. Leverandør av ISP og domenenavn avtales nærmere med oppdragsgiver. Eier av domenet er ØOK(Østfold orienteringskrets).

5 Behov og funksjonskrav

- **Tilgangsmodul**
  Portalen skal ha en tilgangsmodul(pålogging) på minst 3 nivåer. Administrator skal kunne knytte disse til funksjoner/tjenester i portalen.

- **Sikring av innhold**

  Alt innhold og tjenester skal kunne sikres i henhold til «need to know» prinsippet.

- **Funksjoner og tjenester**

  Lagring av bilder, videoer og tekst basert på tagging av ulike typer innhold.

- **Nyhetspublisering (posts)**

  Nyhetspublisering med både bilder, video og tekst.

- **Bloggfunksjon**

  Denne skal brukes til kommunikasjon mellom brukerne.

- **Navigasjon**

  Design avtales nærmere etter forslag fra utvikler. Det samme gjelder antall og navn på aktuelle faner.

- **Faviconer**

  Lenking til sosiale medier og mail til eier av nettsted.

- **Arkiv med søkefunksjoner**

  Både statisk og dynamisk innhold bør være søkbart.

- **Tegneverktøy**
  Det er ønskelig med implementering av et enkelt tegneverktøy med et utvalg av symboler/objekter for løypelegging på kart-filer. Dette som et alternativ til mer omfattende verktøy som er tilgjengelig innen o-sporten. Denne innebærer i all hovedsak: Starttrekant, post-ringer, mål-ringer og streker mellom postene. Tilknyttet en slik funksjon er det ønskelig med en automatisk utregning av løyens lengde samt en enkel postbeskrivelse.

6 Drift og vedlikehold

- **Tilgjengelighet**
  Beskrivelse av vinduer for tilgjengelighet og drift/vedlikehold. Tilgjengelighet til driftspersonell.

- **Beskrivelse av teknisk og systemteknisk sikkerhet**

  Ved valg av ISP leverandør skal det tas hensyn til sikring av web/applikasjonserver og database/filserver mot inntrengning, hacking og overbelastning/DOS. Likeledes skal backup-rutiner beskrives.

7 Videre kontakt

Prosjektleder for Finn Frem i Østfold, Svend Sondre Frøshaug blir kontaktpersonen i forbindelse med alle spørsmål. Han vil også bidra til å fremskaffe all nødvendig informasjon og materiell som trengs i forbindelse med det idrettsspesifikke.

Noen tanker om plattformens bruksområde og kontekst.

1. Kart

Målet for Finn frem i Østfold er blant annet å tegne skolegårdskart for alle skoler i Østfold. Hver skole skal få tegnet et skolegårdskart til bruk I undervisningen som kan finnes I plattformen. Skolegårdskartene skal være av lik størrelse og format for alle skoler. Dette skal gjøre det lettest mulig å sammenlikne og samkjøre undervisningsopplegg. I portalen burde det også ligge en mulighet for å finne også andre type kart for området der dette eksisterer/er mulig, eksempelvis nærmiljøkart (som er en annen type orienteringskart med noe større målestokk). Disse kratrene gjør seg mer egnet for elever i de eldste klassetrinnene på skolene slik at de får utfordret sine orienteringstekniske ferdigheter også utenfor skolegården.

3. Løypeleggerfunksjon

Vi ser for oss at plattformen skal inneholde et enkelt tegneverktøy av løyper som lærerne selv kan bruke for å lage undervisningsopplegg. Dette innebærer å legge til starttrekant, målpost, enkle postringer samt streker mellom postene for å danne en orienteringsløype. En slik funksjon må også kunne regne ut og vise lengden på løypen som er tegnet. På siden av kartet/løypa vil vi også at det skal vises en enkel postbeskrivelse. På skolenivå trenger vi kun en beskrivelse av hvilken kartdetalj posten ligger på og hvor på denne detaljen posten ligger. Eks, på innsiden eller utsiden av et gjerde.

4. Aktivitetsforslag

I portalen ser vi for oss at vi kan legge inn enkle beskrivelser av aktiviteter/øvelser som gjør seg egnet i undervisningssammenheng og som anbefales å benytte i så mate. Dette vil være ulike øvelser som baserer seg på orientering/stafetter/kartøvelser. Disse øvelsene kan beskrives med bilder og tekst, men om mulig også en video der aktivitetene illustreres. Prosjektleder for Finn frem i Østfold er ansvarlig for å fremskaffe denne informasjonen slik at den kan legges inn i portalen/system designet.

Ellers eksisterer det en god portal for dette fra før på:
http://aktivitetsbanken.no/orientering/ovelser/ Det vil være en fordel om det kan ligge inne en snarvei til denne siden I portalen.

5. Klubbens rolle

Målet med Finn frem i Østfold er på sikt å rekruttere flere unge til orienteringsidretten. For å få til dette er vi helt avhengig av at de lokale orienteringsklubbene har en link inn mot skolen. Et ønske er derfor at orienteringsklubbene også skal ha tilgang til plattformen der de kan legge inn ferdige o-tekniske opplegg/treninger for å hjelpe lærere med sine undervisningsopplegg. På denne måten kan lærere finne ferdige undervisningsopplegg i plattformen som kan benyttes, samtidig som at de fremdeles har muligheten til å tegne egne løyper.