

Tittelside

Skal minst inneholde:

- Navn på dokumentet (prosjektet)
- Gruppenummer
- Navn på gruppemedlemmer
- Info om at dette gjelder bacheloroppgaven ved HiØ/IT
- Sted
- Tid

Ingen krav på design/formatering, men bruk gjerne bilder eller annen grafikk... husk at sensor ofte vil se denne forsiden, og at den er med på å “selge” oppgaven.



HØGSKOLEN I ØSTFOLD

Avdeling for informasjonsteknologi
Remmen
1757 Halden
Telefon: 69 21 50 00
URL: www.hiof.no

BACHELOROPPGAVE

Prosjektkategori: Bachelor Thesis	X	Fritt tilgjengelig
Omfang i studiepoeng: 20		Fritt tilgjengelig etter:
Fagområde: Information Systems		Tilgjengelig etter avtale med arbeidsgiver

Tittel Map Management system for orienterings sport in Østfold	Dato: 15. May 2018
Forfattere: Jonas Vestgarden, Michael M. Simon, Adis Jasarevic	Veileder: Børre Stenset
Avdeling / Program: Department of computer science (Information Systems)	Gruppenummer: BO18-G34
Oppdragsgiver: Halden Orienteringskrets	Kontaktperson hos oppdragsgiver: Svend Sondre Mjølnerød Jens Erik Mjølnerød

Ekstrakt:

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

Østfold orienteringskrets planned to avail technology aiming to facilitate a system for sharing orienteering sport map, enhance competence of the teachers and increase interests around orienteering sport in the schools.

3 emneord:

Web application
Agile
Orienteringskart

Forord

Dette er en mal beregnet til bruk i Bacheloroppgaven ved HiØ/IT. Malen gir en pekepinn om både ° struktur og innhold, og hvordan ting kan løses rent skriveteknisk, typisk ved å klippe og lime. ° Malen er utformet som dokumentasjon på et fiktivt prosjekt, der form ° alet er ° å gjøre det lettere ° og enklere å dokumentere en bacheloroppgave (og liknende prosjekter). De fleste kapitler er ° innledet med generelle retningslinjer for hva som skal med (dette er uthevet i grønt). ° Det er tenkt at malen skal kunne brukes i alle de ulike prosjektypene: utvikling, utredning og medieproduksjon. Dermed er mange overskrifter generiske, og må selvfølgelig tilpasses de enkelte ° prosjektene. Det kan også være aktuelt ° å sl ° å sammen enkelte deler av malen, eller legge til kapitler. ° Det er ikke obligatorisk å bruke malen.

Sammendrag

Sammendraget er hele rapporten komprimert til max 1 side. Sammendraget skal gi leseren et godt og tilnærmet komplett bilde av innholdet i dokumentet. Akademiske sammendrag kalles på engelsk for "Abstract", og i mer kommersielle sammenhenger heter det gjerne "Executive Summary". I det siste tilfellet har sammendraget som hensikt å gi ledelsen i en bedrift nok informasjon til å ta økonomiske og/eller administrative avgjørelser. . . uten å lese hele rapporten (!). Tradisjonelt blir sammendraget formattert som et sammenhengende avsnitt. I et bachelorprosjekt, vil hovedformålet være å gi leseren (kanskje i første rekke sensor?) et informativt (og appetittvekkende) bilde av prosjektet. Det er ikke vanlig å bruke litteratur- eller kryssreferanser i sammendraget. Som en regel kan vi si at alt som står i sammendraget, kan det leses mer om i rapporten. Dermed blir utfordringen å belyse alle viktige hovedpunkter, kort og presist. For denne rapporten, kan det f.eks. bli som dette: De nye retningslinjene for evaluering av bacheloroppgaver ved Høgskolen i Østfold/IT legger større vekt på hoveddokumentet enn før. Denne rapporten er resultatet av et prosjekt der formålet var å gi studentene en mulighet for å forenkle og forbedre dokumentproduksjonen. Rapporten er en selvforklarende mal som tar for seg innhold, struktur og layout av hoveddokumentet i bacheloroppgaven. I tillegg er den et konkret eksempel på hvordan man kan bruke LaTeX som dokumentverktøy. Dokumentet er en mal, dvs. et stilsett som brukes for å gi dokumentet ønsket layout. Det blir gitt eksempler på de viktigste teknikkene, slik som bruk av kryssreferanser, kildereferanser, figurer og tabeller, og eksempler på formattering av spesielle elementer, som lister, sitater, definisjoner og matematiske uttrykk. I de tilfellene eksemplene ikke er selvforklarende, blir det gitt råd om hvordan man skal få det til. Intensjonen er at malen kan brukes for alle de tre hovertypene av bachelorprosjekter ved HiØ/IT: Utredninger, mediaproduksjoner, og utvikling av programvare, maskinvare eller systemer. Der det er naturlig å differensiere innholdet i de enkelte kapitlene, blir det skissert mulige løsninger for alle typene prosjekt. Formgivingen er enkel, oversiktlig og tradisjonell. Utgangspunktet for strukturen er den generiske oppbyggingen av et teknisk-vitenskapelig dokument, slik det er beskrevet i Mayfield Handbook of Technical & Scientific Writing. Innholdet i denne rapporten er en (kanskje forvirrende) blanding av generiske retningslinjer og konkret eksemplifisering relatert til prosjektet med å utvikle malen.

Takk Til

Det er vanlig, men ikke nødvendig, å nevne personer og miljøer som har hatt en positiv betydning for prosjektet, f.eks. på denne måten:

Jeg ønsker å takke gode kolleger ved Høgskolen i Østfold, Universitet i Oslo, og Høgskolen i Oslo og Akershus for interessante og fruktbare diskusjoner om utforming, gjennomføring og evaluering av bachelor- og masterprosjekter. I tillegg retter jeg en varm takk til pansermallene Ole, Dole og Doffen for uvurderlig støtte under arbeidet med prosjektet.

Innhold

Forord	3
Sammendrag	4
Takk Til	5
Figurliste	9
Table list	10
Chapter 1 - Introduction	11
1.1 The group	11
1.2 Project owner	11
1.3 Task description	12
1.4 Goals, deliverables og methods	13
1.4.1 Goal	13
1.4.2 Deliverables	13
1.4.3 Metode	14
1.5 Rapportstruktur	14
Chapter 2 - Terminology	16
2.1 Programming languages and data formats	16
2.1.1 Libraries	16
2.2 Framework	17
2.3 Design	17
2.4 Software Engineering	18
2.5 Orienteering	18
Chapter 3 - Project process	19
3.1 Overall process	19
3.2 Planning and Research	19
3.3 Methodologies	19
3.3.1 Design Thinking	19
3.3.2 Agile development	19
3.4 Implementation of methodology	20
Chapter 4 - Research	21
4.1 Research methods	21
4.1.1 Purpose	21
4.1.2 Goal	21
4.1.3 Interview target groups	22
4.4 Technologies	22
Chapter 5 - Analysis	23

5.x Analysis method?	23
5.x Indexing the interview?	23
5.x Synthesising?	23
5.x Categories?	23
5.x Results?	23
5.x Analysis discussion?	24
5.x.1 Current implementation/Orienteering sport	24
5.1 Technology criterias	24
5.1.1 Programming libraries	24
5.1.2 Database system	24
5.2 Application functionality from a user perspective	24
Chapter 6 - Design	25
Figure nr : The figure above illustrates the general structure of the system.	25
6.1 Domain description	25
6.2 Interface	26
6.2.1 Design Requirements	26
6.2.2 Use case	26
6.2.3 Statement (hva alt skal gjøre)	26
6.2.4 Sequence diagram	26
6.2.5 Design iterations	26
6.2.5.1 First iteration	26
6.2.5.2 Second iteration	26
6.2.5.3 Third iteration	26
6.3 Functionality	26
6.3.1 Functional Requirements	26
6.3.2 Security Requirements	26
6.3.3 Use case	26
6.3.4 User stories	26
6.3.5 Sequence diagram	26
Chapter 7 - Documentation	27
7.1 Javascript documentation	27
7.2 PHP documentation	27
7.3 SQL documentation	27
7.4 HTML/CSS documentation	27
Chapter 8 - Testing	28
8.1 Test plan	28
8.2 Test types	28
8.3 Test criterias	29
8.4 Results and measures taken	29
8.5 Second iteration of testing	29

Chapter 9 - Implementation	30
9.1 Database	30
9.2 Server	30
9.3 Administration	30
9.4 Dependencies	30
Chapter 10 - Discussion	31
Chapter 11 - Conclusion	32
(Chapter 12 - Future work, include this if we are not done)	33
Register	34
Tillegg A	35

Figurliste

4.1 Layout for recto sider	16
A.1 The LATEX “bible”, 2. edition	25
A.2 Input and result from running the Douglas-Peucker line simplification algorithm [1] (from [2])	29

Table list

Chapter 1 - Introduction

In this part we will have an introduction about the content that this specific chapter contains.

1.1 The group

All of the group members are third year Information systems students with different interests and competence.

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.

1.2 Project owner

Østfolds o-krets 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).

Beskriv oppdragsgiver, både firma og kontaktpersoner, snakk med Svend Sondre for mer info.**NOTE:** Finnes ikke veldig mye info om østfold o-krets på nett.

1.3 Task description

Different schools around Østfold implements orienteering sport in their curriculums. The purpose of introducing the sport is to enhance maps and map reading ability of the students by applying enjoyable and inclusive activity.

The purpose of the project FinnFrem i Østfold is to mobilize for physical activity and use of maps among juveniles, as well as to showcase and spread interest around the World Cup in orientation in 2019.

The maps that are currently in use are the old paper based maps. Østfold O-krets wishes to implement/apply effective digital solutions that will help the archive and as well share the maps across all schools located in Østfold.

The system will be designed to solve challenges around map sharing and help those who operate it save and edit on a shared platform. This will make it easier for other schools to enter the platform and download the finished maps for printing.

In addition to sharing, the system will also be used to draw control points(indicates hidden points) and other relevant information for the map. This functionality is suggested by the group members with a hope of reducing use of third party applications for drawing control points and other information that are required in the map for performing orienteering sport.

There are two options of the application platform. One is to build a web application that runs on the browser. A web application (also called web app) is almost the same as a desktop application. They offer very similar functionality, but the main difference is that a desktop app runs locally on your computer, whereas a web app runs on the server. Since the goal is to create a system that allow teachers to get away from using different desktop apps, this solution is very viable. The second option is to combine a web app with a light weight mobile version with different functionalities depending on the platforms performance, applicability and usability.

1.4 Goals, deliverables og methods

1.4.1 Goal

Goal The main goal of the project is to develop a platform for sharing and draw Orienteering maps that will be used in different municipalities of Østfold country.

Users of the platform will be able to...

Subgoal 1 upload, store, download, edit and print Orienteering maps.

Subgoal 2 select a map based on location.

Subgoal 3 suggest activities and materials for supporting the learning process.

Subgoal 4 make maps available to specific user groups.

In addition to the goals mentioned above the final solution is expected to be built focusing on the points mentioned hereunder.

Simple and effective

The platform should be simple and easy to access required information both when navigating and searching with a use of clear and comprehensible language.

User friendly and intuitive

The platform should be developed by taking into consideration vital user experience and usability solutions and standards.

Accessible

The platform is expected to comply to web content accessibility guidelines and principles. The project will be developed following the guideline specified in web content accessibility guidelines(WCAG) 2.0.

Responsive design

The solution will be responsive. The platform in general will be accessed from mobile, tablets and desktop computers and different functionalities may be accessible on different devices depending on the applicability of the devices to perform certain task.

1.4.2 Deliverables

(Jobb Videre med denne delen)

The final product will be a platform for upload, share and draw orienteering map. The website will be used by teachers and activity centers from 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.

Users can upload, share, draw (set annotations), print and retrieve saved maps.

Users can automatically generate annotations on the maps.

Users 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
- Post orienteering map
- Draw points in the maps using the platform
- Print maps
- Admincontrol (post news and learning materials)
- Import maps that are made using other map drawing application, for example OCAD.

1.4.3 Metode

Beskriv konkret hvilken utvikling metodologi som skal benyttes, hvordan skal gruppen skal komme frem til slutt løsning, hvordan design thinking prinsippet skal implementeres (detaljert om de fem steger i design thinking prosess- research, define, ideate, prototype, testing). Hvordan gruppen skal benytte scrum og levering av sprint, møter og arbeidsfordeling.

Verktøy: Hvilken verktøyer som er benyttet for å strukturere, utføre, kontrollere arbeidsoppgaver. (Trello, github).

1.5 Rapportstruktur

This document is structured based on the template delivered by HiOF.

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: Here the different terminology used throughout the document will be explained.

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

Chapter 4 - Research: The research that was done prior to development, how it was conducted and the outcome of the research.

Chapter 5 - Analysis: An analysis of the results of the previous chapter, and what those results were used for. What does the user need, and what did we need to implement to achieve this.

Chapter 6 - Design: The system design, process, implementation and related decisions taken based on the requirements specification.

Chapter 7 - Documentation: Documentation of the most important code written.

Chapter 8 - Testing: The testing that was done during development, the criterias for those tests, how they were performed and what the results were and used for.

Chapter 9 - Implementation: How we implemented the solution and a guide for others to do the same.

Chapter 10 - Discussion: A discussion of deliverables, implementation, the solution, further work and maintenance.

Chapter 11 - Conclusion: The conclusion of the project.

Chapter 2 - Terminology

In this chapter we will talk about different definitions and terminology used in this document

2.1 Programming languages and data formats

The application is mainly written in Javascript using the MEAN stack in NodeJS.

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.

CSS

CSS stands for Cascading Style Sheets, and is primarily used to modify 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 an 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.

JSON

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

SVG

SVG stands for Scalable Vector Graphics and is a 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.

2.1.1 Libraries

SVG.JS

SVG.JS is a "lightweight library for manipulating and animating SVG." (<http://svgjs.com/>) 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 framework used for the application is the MEAN stack in NodeJS, a platform built on Google's V8 javascript engine. It uses a NoSQL database called MongoDB to store data in JSON-like documents. MEAN stands for **M**ongoDB, **E**xpressJS, **A**ngularJS, **N**odeJS.

MongoDB

MongoDB is a 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.

ExpressJS

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

AngularJS

AngularJS is an open source javascript framework developed by google. It is "what HTML would have been, had it been designed for applications".

(<https://docs.angularjs.org/guide/introduction>)

NodeJS

NodeJS is a framework built with the intent to provide an easy way to build scalable web applications. It's based on Google's V8 engine, which allows it to handle thousands of concurrent connections on a single thread.

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.

Material Light

Material Light is a CSS library that is developed based on the android android material design principles. Material light is developed with a purpose of creating new interactions of user and elements by transforming the behaviour 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 contains 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.

2.4 Software Engineering

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.

Unit Testing

Unit testing is facilitate testing of small parts of the application called units.

2.5 Orienteering

Orienteering is a kind of sport that can be performed 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.

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).

Orienteering. (2018). *About orienteering*. Hentet fra <http://www.orienteering.ca/about/orienteering/>.
Norsk Orientering. (2018). *Østfold*. Hentet fra <http://www.orientering.no/kretser/ostfold/>.

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 as a tool for learning use of compass and map reading. The activity helps increasing social activities and enhance interaction between students.

Chapter 3 - Project process

The purpose of this chapter is to explain how we planned working towards the goal, what decisions we took to achieve this, how we ended up working and what we failed to do according to the plan.

3.1 Overall process

At the start of the process, first and foremost, we thought to gather information about which programming language we would use. The conclusion became javascript in collaboration with nodejs. Nodejs is an open cross platform for javascript. We found that this combination turned out to be easier to use than the others considering the functions of the requirement specification. After this thinking process, we had a conversation with the project supervisor on which image-format was the best for animating maps. We searched for javascript libraries that would suit the test process and landed on svg.js, which is a lightweight library for manipulating and animating svgs. This helped us move forward to the next process which is to test code the idea of the functions that are present in the specification. We made the first version of the program and it showed results.

3.2 Planning and Research

Planning and research are responsible for putting the group on the right path, by creating different plans, such as strategic choices, and making important decisions in terms of sources and knowledge.

Here the group will take important steps. Based on these important phases, the group will gain insight into how they have planned research before making a choice and what kind of knowledge they have used and received before they have made a choice.

3.3 Methodologies

Methodology is a research strategy in which the person applies methods to a field of study.

3.3.1 Design Thinking

3.3.2 Agile development

3.4 Implementation of methodology

Hvordan gikk det, avvik osv.

Chapter 4 - Research

4.1 Research methods

http://eprints.ncrm.ac.uk/3276/1/complete_proofs.pdf

Qualitative interview information

There are three main ways to perform a qualitative interview: Structured, semi-structured and unstructured interview.

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.

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.

The last one, a unstructured interview, doesn't necessarily have a set of questions, but rather a topic or subject to cover. This allows the subject to answer in a way that's comfortable to them, using their own ideas and is not influenced by the questions you ask. This is probably the hardest interview to hold as you as the interviewer need to keep the conversation going, while trying not to influence the subject when asking about a particular topic.

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

4.1.1 Purpose

The purpose of the research is to acquire the necessary data that helps discovering the user needs and define the functionalities that are required. The research will also be used when mapping the prioritised functionalities that are needed for the application to perform the required tasks.

4.1.2 Goal

The goal of the research is to acquire relevant information that helps defining and prioritising core functionalities of the application.

4.1.3 Interview target groups

The research groups will mainly be end-users of the application, these are teachers, orienteering sport clubs and ski centers. The contact with research groups will be continuous throughout the development and testing phase of the application modules.

4.4 Technologies

Our decisions based on the results from the interview

From the results above in addition to the requirements specification, we can decide what functionality we need implemented, and how we should design the interface. Some technology was already decided prior to this (SVG.js and MEAN stack) but further development will be influenced heavily by this data.

Write how we come up to the choice of technologies used grouped by functionality, database, interface etc.

Chapter 5 - Analysis

Titles listed as 5.X are placeholder titles we may or may not use, and most certainly will be renamed.

5.x Analysis method?

In this process, the analysis is performed with synthesis. To start off we have to

- code(index) the data, make the data manageable and draw out the most important data.
- We need to identify, examine what we see and decide which codes are the most important.
- Having the codes that are the most important and create categories to make the work more on a abstract level.
- Label the categories and decide which are the most relevant and find out how they are connected to each other. The categories and the connection between them are the core of the whole study and the perspective of the participant is the new knowledge.
- In this step we have to decide if there are any hierarchy among the categories and decide if one category is more important than the other.

5.x Indexing the interview?

5.x Synthesising?

5.x Categories?

5.x Results?

5.x Analysis discussion?

5.x.1 Current implementation/Orienteering sport

Sort out the current practices from different interviews that will be performed and describe here.

5.1 Technology criterias

A description of what led to us using certain libraries, frameworks etc. Things that led to this would be research done (reading about different ways to implement the solution), going through the requirements specification (ultimately the things we decide on needs to be compatible with the requirements), interviews done with the end users (teachers, clubs etc.), talking to experts on different technologies (lecturers in our school etc.) and continuously iterating on the requirements with the project leader.

Community support of the programming languages, libraries and frameworks.

Their ability of solving the problem.

Time saving.

Future popularity of libraries and maintainability.

Availability of the technology

Easy to learn in short amount of time.

5.1.1 Programming libraries

Which libraries we chose and why

5.1.2 Database system

Which database system we chose and why

Probably more sub-titles here to explain some more choices that doesn't fall into these categories

5.2 Application functionality from a user perspective

Functionality from a users perspective will always differ from our perspective as developers, and the project leaders perspective for what they want (Ref. to the what the * wanted etc. picture)

Chapter 6 - Design

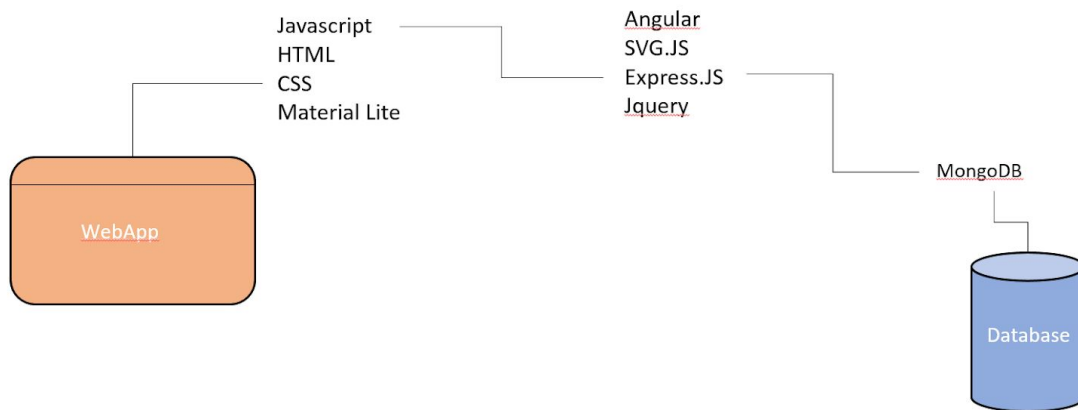


Figure nr : The figure above illustrates the general structure of the system.

6.1 Domain description

Describe the domain structure and how it is constructed. The connections between different elements of subsystems and components.

6.2 Interface

6.2.1 Design Requirements

6.2.2 Use case

6.2.3 Statement (hva alt skal gjøre)

6.2.4 Sequence diagram

6.2.5 Design iterations

6.2.5.1 First iteration

6.2.5.2 Second iteration

6.2.5.3 Third iteration

6.3 Functionality

6.3.1 Functional Requirements

6.3.2 Security Requirements

6.3.3 Use case

6.3.4 User stories

6.3.5 Sequence diagram

Chapter 7 - Documentation

7.1 Javascript documentation

7.2 PHP documentation

7.3 SQL documentation

7.4 HTML/CSS documentation

Chapter 8 - Testing

8.1 Test plan

We will perform iteratively 8 tests which are functionality-testing, usability testing, interface testing, database testing, compatibility testing, performance testing, security testing and crowd testing.

8.2 Test types

Functionality testing

Check if each function of the web application suits the requirement specification intended for it. This is mainly involved as black box testing which is a method for examining the functions without looking into the source-code or the internal structure. The person testing the software is aware of what it is supposed to do but not aware of the mechanism behind it. In other words, the tester is aware that a input returns a output but not aware of the functionality.

Usability and accessibility testing

Usability testing or easy of use falls under the type “functionality testing”. The difference is that the testers are real users and that this type of testing method gives direct input. Its focus is to measure the user's behavior and action and see if it meets the requirement specification.

Accessibility can be tested using automated testers(validations), keyboard and The straw test.

Interface testing

This test requires three areas to be tested. These areas are application, web and database server. Testing the requests sent to the database and see if it outputs a correct display. See if the server is working correctly with coping with the application requests and not denying the service. Last and not least is the database server which is testing if the queries sent outputs expected results.

8.3 Test criterias

What are our criterias for performing the tests listed above? And why they are our criterias.

8.4 Results and measures taken

What we did with the results from testing in the first iteration.

8.5 Second iteration of testing

Same as above, might use under titles or something here. Basically what differs is changes we did to criterias from the first round, what new things did we find out, what issues persisted, what new measures did we take, what didn't work on the first iteration.

Chapter 9 - Implementation

Basically a guide to implement the final result

9.1 Database

9.2 Server

9.3 Administration

9.4 Dependencies

Chapter 10 - Discussion

Chapter 11 - Conclusion

(Chapter 12 - Future work, include this if we are not done)

Register

LaTeX, 25, 26, 28, 29, 33, 34

Recto, 12, 16

Recursion, *Se* Recursion

Tower of Hanoi, 32

Tillegg A