



Stadiontavle.no
Informasjonstavle for idrettslag

Bachelorprosjekt

VÅREN 2022





Institutt for Informasjonsteknologi
 Postadresse: Postboks 4 St. Olavs plass, 0130 Oslo
 Besøksadresse: Holbergs plass, Oslo

PROSJEKT NR.

62

TILGJENGELIGHET

Offentlig

Telefon: 22 45 32 00

HOVEDPROSJEKETS TITTEL Stadiontavle.no - Presentasjon av aktiviteter	DATO 25.05.2022
	ANTALL SIDER/BILAG 137/36
PROSJEKTDELTAKERE Dani Tran - s346177@oslomet.no Kisan Srirangan - s345873@oslomet.no Linda Kadrijaj - s333751@oslomet.no Marthe V. Marstrander - s333930@oslomet.no	INTERN VEILEDER Roza Abolghasemi, stipendiat rozaabol@oslomet.no +47 67 23 54 87
OPPDRAKGIVER Digimaker AS post@digmaker.no +47 467 07 009	KONTAKTPERSON Christian Randulf Bull, daglig leder christian@digmaker.no +47 915 75 899

SAMMENDRAG

Prosjektet ble utført av studenter ved studiene Dataingeniør, Informasjonsteknologi og Anvendt datateknologi ved OsloMet, med Digimaker som oppdragsgiver. Det blir presentert en ny løsning for oppsettet av tavler som skal ligge lett tilgjengelig i hovedmenyen på administrasjonssiden til stadiontavle.no, der adminbruker enkelt kan sette opp informasjon som skal vises på tavlene for idrettsklubber.

3 STIKKORD

Presentasjon

Scrum

Angular

Forord

Denne rapporten tilhører Bachelorgruppe 62 fra OsloMet - Storbyuniversitet, og er en fullstendig rapport i forbindelse med bachelorprosjektet vi har gjennomført hos Digimaker AS.

Vi vil gi en stor takk til Digimaker AS for å ha gitt oss muligheten til å skrive oppgave hos dere. Spesielt vil vi takke prosjektveileder Christian Randulf Bull for tips og råd, og samtidig bidro Christian med tett oppfølging gjennom hele prosjektprosessen. I tillegg er vi veldig takknemlig for utviklingsteamet hos Digimaker AS som har gitt oss opplæring og hjulpet oss i praksis med problemer vi har støtt på under arbeidet. Dette teamet består av Aksel Susegg, Chen Xiongie og Morten Sydow Mo. Til slutt vil vi takke vår interne veileder hos OsloMet, Roza Abolghasemi, for å ha brukt tid til å gi oss gode tilbakemeldinger i forbindelse med rapporten selv under hektiske perioder.

Rapporten kan leses på papir, men anbefales å leses elektronisk. Med tanke på at det inneholder lenker som gjør det enklere å navigere seg til de ulike vedleggene i rapporten.

Leserveiledning

Dette er slutt dokumentasjonen for bacheloroppgaven ved OsloMet – Storbyuniversitet, som en del av studieprogrammet i vårsemesteret 2022 i samarbeid med Digimaker.

Presentasjon av prosjektet

Den første delen av rapporten er [Kapittel 1: Presentasjon av prosjektet](#). Dette kapittelet inneholder en introduksjon av prosjektgruppen, veileder og oppdragsgiver. Her blir det gitt en generell beskrivelse av problemstillingen, målgruppen og løsningen. Leseren vil etter å ha lest denne delen ha en bedre forståelse av hva dette prosjektet inneholder.

Kravspesifikasjon

Den andre delen er [Kapittel 2: Kravspesifikasjon](#). Dette kapittelet inneholder en oversikt med kravspesifikasjonen som ble satt av oppdragsgiver, med kartlegging og analyse av kravene. Denne delen gir også oversikt over skisser som ble utformet under utviklingen for å kunne møte kravene på en tilfredsstillende måte. Leseren vil etter å ha lest dette kapittelet få en bedre forståelse av hvordan kravspesifikasjonen ble brukt under utviklingen.

Prosessdokumentasjon

Den tredje delen er [Kapittel 3: Prosessdokumentasjon](#). Dette kapittelet inneholder prosjektets prosess, planlegging og metodikk. Videre går vi inn på hvilken arbeidsmetode som ble brukt, med detaljert beskrivelse av de forskjellige fasene i prosjektet. Leseren vil etter å ha lest dette kapittelet kunne forstå prosessen til prosjektet.

Produktdokumentasjon

Den fjerde delen er [Kapittel 4: Produktdokumentasjon](#). Dette kapittelet inneholder en kort beskrivelse av teknologiene og verktøyene som ble brukt for å utvikle løsningen. Videre gir denne delen en detaljert oversikt over selve utviklingen av produktet. Leseren vil etter å ha lest dette kapittelet få bedre forståelse av utviklingsprosessen til produktet.



Testdokumentasjon

Den femte delen er [Kapittel 5: Testdokumentasjon](#). Dette kapittelet inneholder de forskjellige testene vi utførte under og etter utviklingen av koden, samt resultatene på disse testene.

Brukerveiledning

Den sjette delen er [Kapittel 6: Brukerveiledning](#). Dette kapittelet inneholder en brukerveiledning for hvordan en tar i bruk løsningen som ble utviklet, samt at en får et visuelt innblikk i hvordan designet på brukergrensesnittet ser ut. Leseren vil etter å ha lest dette kapittelet ha bedre forståelse av hvordan løsningen fungerer.

Konklusjon og refleksjon

Den syvende delen er [Kapittel 7: Konklusjon og refleksjon](#). Dette kapittelet gir en generell konklusjon av hele prosjektet.

Vedlegg

Den åttende delen er [Kapittel 8: Vedlegg](#). I dette kapittelet ligger diverse vedlegg fra prosjektet som blant annet prosjektdagboken, kontrakten, installasjonsveiledning mm.

Kildehenvisning

Den niende delen er [Kapittel 9: Kildehenvisning](#). I denne delen ligger kildene som ble brukt i rapporten, og disse er henvist som fotnoter.

Innholdsfortegnelse

Forord	3
Leserveileddning	4
Innholdsfortegnelse	6
Kapittel 1: Presentasjon av prosjektet	9
1.1 Innledning	9
1.2 Problemstilling	9
1.2.1 Prosjektets mål	10
1.2.2 Målgruppe	10
1.3 Teoretisk bakgrunn	10
1.3.1 Systemutvikling	11
1.3.2 Visualisering	11
1.3.3 Interaksjon	11
1.3.4 Testing av programvare	11
1.4 Presentasjon av prosjektgruppen	12
1.5 Presentasjon av veiledere	13
1.6 Presentasjon av oppdragsgiver	13
Kapittel 2: Kravspesifikasjon	14
2.1 Innledning	14
2.2 Kravspesifikasjon og dens rolle	14
2.2.1 Kartlegging	14
2.2.2 Analyse og funksjonalitet/design	16
Kapittel 3: Prosessdokumentasjon	27
3.1 Innledning	27
3.2 Forarbeid	27
3.2.1 Hvorfor Digimaker	27
3.2.2 Arbeidsplan	28
3.3 Metodikk	29
3.3.1 Fossefall vs smidig metode	29
3.3.1.1 Smidig arbeidsmetode	30
3.3.2 Scrum	30
3.3.2.1 Implementasjon av Scrum	30
3.4 Planning Poker	31
3.5 Sprinter	32
3.5.1 Sprint 1	33
3.5.2 Sprint 2	35



3.5.3 Sprint 3	37
3.5.4 Sprint 4	39
3.5.5 Sprint 5	41
3.6 Plattform og verktøy	43
3.6.1 Discord	43
3.6.2 Microsoft Teams	43
3.6.3 GitHub	43
3.6.4 Google Drive	44
3.7 Refleksjon	44
3.7.1 Refleksjon av arbeidsmetodikk og sprinter	44
3.7.2 Refleksjon av plattform og verktøy	45
3.8 Avslutning	46
Kapittel 4: Produktdokumentasjon	47
4.1 Innledning	47
4.2 Systemarkitektur	47
4.2.1 Overordnet arkitektur	48
4.2.2 Kommunikasjon	49
4.2.3 Logisk arkitektur	49
4.2.3.1 Frontend	49
4.2.3.2 Backend	50
4.3 Teknologier og verktøy	50
4.3.1 Angular	50
4.3.2 Laravel	51
4.3.3 MySQL	51
4.3.4 Visual Studio Code	51
4.3.5 PhpStorm	51
4.3.6 Node JS	52
4.3.7 Composer	52
4.3.8 phpMyAdmin	52
4.4 Frontend	53
4.4.1 Filstruktur	53
4.4.2 Implementasjon	57
4.5 Backend	67
4.5.1 Filstruktur og database	67
Kapittel 5: Testdokumentasjon	71
5.1 Innledning	71
5.2 Testplan	71
5.3 Enhetstesting	72

5.3.1 Hvilke funksjoner ble testet	72
5.3.2 Resultat av enhetstest	73
5.4 Systemtesting	75
5.4.1 Hvilke funksjoner ble testet	75
5.4.2 Resultat av systemtest	75
5.5 Akseptansetest	79
5.5.1 Hvilke funksjoner ble testet	79
5.5.2 Resultat av akseptansetest	79
5.6 Konklusjon	82
Kapittel 6: Brukerveiledning	83
6.1 Innledning	83
6.2 Nettside - Stadiontavle.no	83
6.2.1 Logg inn	84
6.3 Dashboard	85
6.3.1 Navigasjonsbar	85
6.4 Implementering av nye visninger	86
Kapittel 7: Konklusjon og refleksjon	94
7.1 Konklusjon	94
7.2 Potensielle endringer	95
7.2.1 Klubb-ID	95
7.2.2 Maler	95
7.2.3 Design	96
7.3 Refleksjon	96
Kapittel 8: Vedlegg	97
8.1 Prosjektdagbok	97
8.2 Oppgavebeskrivelse fra Digimaker	110
8.3 Installasjonsveileder	111
8.4 Feilmelding	124
8.5 Fremdriftsplan	125
8.6 Kontrakter	127
Begrepsliste og språk	131
Kapittel 9: Kildehenvisning	134

Kapittel 1: Presentasjon av prosjektet

1.1 Innledning

Gruppen består av fire studenter som utfører prosjektet som en del av bachelorstudiet ved OsloMet. Dette prosjektet blir utført hos Digimaker AS som er en ekstern oppdragsgiver. Prosjektet skal danne et grunnlag for bacheloroppgaven. Gruppen skal jobbe sammen som et team av utviklere hos bedriften, og har dermed ansvaret for prosjektets resultat og fremdrift.

Digimaker har utviklet en elektronisk informasjonstavle for idrettsplasser kalt "Stadiontavle". Informasjonstavlen viser relevant informasjon for diverse idrettslag. Systemet kan blant annet benyttes til å planlegge og vise bane-, garderobefordeling. I tillegg blir det vist kollektivtilbud fra idrettsplassen på skjermen. Man kan også se kamprapportering, reklame, annonser og sponsorinformasjon, og kommunisere dette til lagledere og spillere. Det er relevant for idrettslag å ha en strukturert oversikt slik at alle medlemmer av laget kan samarbeide på best mulig måte.

1.2 Problemstilling

For at en digital løsning generelt skal bli benyttet av en kunde, er det viktig å skape et godt brukergrensesnitt som ikke krever for mye forkunnskap av de som skal ta det i bruk.

Dagens løsning på stadiontavle.no er ikke spesielt brukervennlig, da det å sette opp elementene som skal vises på tavlene hverken er intuitivt eller oversiktlig, etter tidligere tilbakemeldinger og testing fra kundene. Etter analyse gjort av gruppen, kom vi frem til at det er få layouts å velge mellom, samt lite designvalg i editoren. Fargevalg og struktur har ikke vært et fokus under utvikling og det er ikke tilstrekkelig med informasjon på nettsiden om hva slags oversikt ett idrettslag får, og hva informasjonen som sendes ut brukes til.

Digimaker ønsker en ny løsning for oppsettet av tavlene som ligger lett tilgjengelig i hovedmenyen på administrasjonssiden, der brukeren enkelt kan sette opp informasjonen

som skal vises på tavlene. Denne editoren skal ha flere valg for layout og design, samt ha muligheten til å se en forhåndsvisning av tavlen.

Det var av den grunn viktig at vi undersøkte:

- Hvilke layouts og visninger av informasjon som ville fungere best med dagens løsning.
- Hvordan vi kan utvikle editoren til å ha et godt brukergrensesnitt som vil skape en bedre brukeropplevelse for brukerne.
- Om sluttproduktet kan integreres på en velfungerende måte.

1.2.1 Prosjektets mål

Det overordnede målet med prosjektet er å forbedre dagens løsning ved å gi brukerne flere valgmuligheter for design og layout av den informasjonen som skal vises på stadiontavlen. Ved å øke funksjonaliteten i editoren på administrasjonssiden, vil brukerne få tilgang til en mer fleksibel tjeneste som vil bidra til å gi en bedre visuell presentasjon på tavlene.

1.2.2 Målgruppe

Det er ønsket at denne webbaserte løsningen skal gi brukere en mer oversiktlig og strukturert informasjon om idrettslag og deres ressurser. Målgruppen for denne løsningen er derfor fotballklubber og andre selskaper hvor det er relevant at denne løsningen blir brukt. Digimaker tilbyr utvidet funksjonalitet som ressurshåndtering, måltavle med klokke, integrasjon FIKS og egne annonser/sponsorer. For de fleste klubber gir denne løsningen mulighet til å vise nødvendig informasjon, der hver klubb kan ha en felles oppdatert oversikt til enhver tid.

1.3 Teoretisk bakgrunn

Teorien fra flere av fagene vi har hatt under studiet har vært sentrale under dette prosjektet. Fagene som har vært mest aktuelle under utviklingen er systemutvikling, visualisering, interaksjon og testing av programvare.



1.3.1 Systemutvikling

Systemutvikling handler om det å utvikle digitale produkter som har tilstrekkelig kvalitet, innen gitt tid og kostnadsrammer. Det er viktig at resultatet av en løsning blant annet er funksjonell, brukervennlig og pålitelig. Typiske aktiviteter under utviklingen er alt fra planlegging, innsamling av kravspesifikasjon, utforming av design, koding, og ferdigstilling av en løsning¹.

1.3.2 Visualisering

Visualisering blir brukt for å presentere data på en forståelig måte. Ved å bruke visualisering kan data gå fra en mengde med tall til å bli informasjon som lettere kan bli oppfattet av hjernen ved bruk av en god presentasjon².

For nye brukere av en løsning kan mye informasjon på en gang føre til kognitiv overbelastning. Visuelle virkemidler som bruk av ikoner eller farger gjør det lettere å skille informasjonen fra hverandre, noe som kan skape en bedre brukeropplevelse.

1.3.3 Interaksjon

Brukergrensesnittet er det brukeren har å forholde seg til når vedkommende har interaksjon med den digitale løsningen, det er derfor viktig at brukeren får tillit til brukeropplevelsen³. Utformingen på grensesnittet er viktig å ta hensyn til når man utvikler en løsning, da det er dette som gjør at brukeren kan kommunisere med maskinen. Grensesnittet skal gi brukeren inntrykk av funksjonene som er tilgjengelig i produktet. Det har av den grunn blitt tatt hensyn til blant annet valg av tekst, fargebruk, valg av ikoner mm. når frontend delen av løsningen ble utviklet.

1.3.4 Testing av programvare

Testing skal kunne redusere risikoen for dårlig kvalitet av et system, og det skal kunne skape tiltro til kvalitetsnivaet ved hjelp av testobjektene⁴.

¹ (Lindsjørn, 2021a)

² (Ramsøy, 2021)

³ (Sandnes, 2018)

⁴ (Krattebøl, 2020a)

Ved å gjennomføre testing av løsningen kan man sjekke at koden fungerer på en ønskelig måte, samt at en kan undersøke om kravspesifikasjonene har blitt tilfredsstilt. Det er også viktig å teste løsningen for å rette opp eventuelle feil, noe som vil spare tid under utvikling da eventuelle feil kan bli rettet fort opp.

1.4 Presentasjon av prosjektgruppen

Dani Tran er en Informasjonsteknologistudent på OsloMet. Han gikk rett inn på studiet etter videregående skole, ettersom IT-fagene på skolen fanget interessen hans. Gjennom studiet ble også interessen for front-end utvikling større. Han har alltid likt design, og samtidig er han veldig opptatt av at brukeren skal ha en god opplevelse. Dette har gitt oss en positiv virkning i utviklingen av prosjektet. Med økende interesse for front-end og design, ønsker Dani å gå videre med en master i informatikk innen design, bruk og interaksjon.



Kisan Srirangan studerer Dataingeniør på OsloMet. Interessen for IT og teknologi har alltid vært i livet hans. Valget av studiet og karriere var dermed enkel. På videregående hadde han valgfagene IT1 og IT2, som ga han et grunnlag for hva han ville møte på som IT-student. I løpet av studietiden har han fått en økt mengde med interesse og erfaring, som har ført til at han vurderer å studere master etter endt bachelorstudium.



Linda Kadrijaj studerer informasjonsteknologi på OsloMet. Hun fikk interesse for IT etter at hun var på yrkesmesse. Hun startet på studiet i 2018 og begynte kort tid etter å jobbe på OsloMet som studentassistent ved IT support førstelinje. Der økte også erfaringen. Etter to år fikk hun 100% stilling i Sopra Steria og tar bachelorgraden på siden. Etter endt bachelor ønsker hun å fortsette i stillingen som IT-konsulent og bygge videre rundt dette, muligens innenfor [Cloud](#).



Marthe V. Marstrander går siste året på Anvendt datateknologi, og har spesialisert seg innen front-end utvikling og [UX/UI](#) design. Hun studerte jus tidligere, men bestemte seg for å bytte over til IT da interessen for data og teknologi alltid har vært stor. Hun brenner for å gjøre teknologiske løsninger mest mulig brukervennlige for folk flest, og ønsker etter endt studie å jobbe med web/apputvikling, da med spesielt fokus på å skape gode brukeropplevelser og designe intuitive [brukergrensesnitt](#).

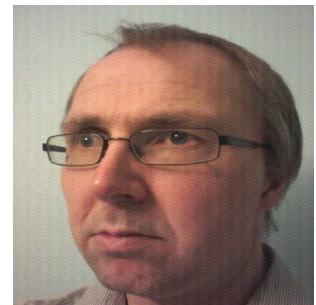


1.5 Presentasjon av veiledere

Roza Abolghasemi er veilederen vår fra OsloMet. Hun er en stipendiat og har sin tilhørighet til Applied Artificial Intelligence, fakultet for teknologi, kunst og design (TKD), og Institutt for informasjonsteknologi på universitetet.



Christian Randulf Bull innehavar stillingen som daglig leder i Digimaker AS. Han er prosjektveilederen vår i bedriften. Vi har hatt møter med Christian ukentlig, og han har fulgt opp gruppen med tips og hjelp gjennom hele prosjektperioden.



1.6 Presentasjon av oppdragsgiver

Digimaker leverer løsninger på flere plattformer, for eksempel Digimaker [CMS](#), Digimaker [CMF](#), [eZ Publish](#) og [Wordpress](#). De har god erfaring med Angular og React. Løsningene til Digimaker kjennetegnes ved at de er skreddersydde og brukervennlige. Disse løsningene er portalløsninger som knytter sammen informasjon fra ulike interne systemer, eksterne kilder og eksterne mottakere. Selskapet bygger og digitaliserer ulike virksomheter sine [kjerneprosesser](#), og gjør det mulig for ansatte, kunder og andre brukere å se all nødvendig informasjon. Eksempler på virksomheter er entreprenørselskapet Selvaag AS og Kongsberg Maritime. Selskapet har publiseringssystem som organisasjoner kan dra nytte av ved å publisere innhold online på en mer innovativ måte, noe som gjør navigering enklere for kundene.

Kapittel 2: Kravspesifikasjon

2.1 Innledning

For å kunne utvikle en IT-løsning er det viktig for en kunde å fremlegge ønsker om hvordan en slik prosess skal løses. Det utarbeides dermed en kravspesifikasjon. Det fremstilles behov, avgrensing og definisjon av det ønskede systemet. Det er viktig å ha med kartlegging av behovet, det tekniske, analyse, funksjonalitet, arkitektur og metodikk. Disse elementene kan komme godt med for å utarbeide en god kravspesifikasjon, ettersom at utvikling av en IT løsning er forskjellig fra kunde til kunde.

Vi kan dele kravspesifikasjoner inn i funksjonelle og ikke-funksjonelle krav. De funksjonelle kravene definerer hvordan systemet skal funksjonere og kan forklares med systemutfallet etter at en bruker utfører en oppgave på systemet. De ikke-funksjonelle kravene definerer hvordan systemet skal være i motsetning til hva det skal gjøre, for eksempel ytelse eller brukervennlighet.

For gruppen var det viktig å kunne sette seg raskt inn i prosessen for utviklingen. Vi bestemte oss tidlig for å sette opp behovet til kunden, hvordan oppgaven skulle løses ut ifra disse behovene og hvordan ønsket løsning skulle være. Kravspesifikasjonen er utarbeidet på bakgrunn av møter som ble holdt mellom Digimaker og gruppen. De funksjonelle og ikke-funksjonelle kravene er basert på inspirasjon fra systemutvikling, fra tidligere bacheloroppgaver og listen med krav fra Digimaker. Formålet med kravspesifikasjon var å ha oversikt over alle aspektene av utviklingen.

2.2 Kravspesifikasjon og dens rolle

2.2.1 Kartlegging

Under kartleggingen ønsket vi å konkretisere kravene som Digimaker hadde til prosjektet. Innad i gruppen kom vi frem til at det var ønsket en videreutvikling av eksisterende webløsning der det skulle implementeres nye visninger med minimalt behov for programmering.

Kravene som ble spesifisert av Digimaker delte inn i både funksjonelle og ikke-funksjonelle krav:

Funksjonell	Ikke-funksjonell
Flere valgmuligheter for visning av aktiviteter i form av ferdige layouts	
Flere parametere i editoren på administrasjonssiden for brukerne. Valg av visningstype, farger mm.	
Etablering av nye maler med HTML og koder for aktiviteter for administrator	
Det kan bli aktuelt å se på den generelle bruken av editoren.	
	Det skal brukes samme kodespråk som den eksisterende løsningen, men det er åpning for forslag av andre løsninger som kan iverksettes inn i den nåværende.

Før vi begynte å jobbe på selve prosjektet valgte vi å bruke tid på å avdekke kravene ved å skissere dem slik at vi fikk en bedre forståelse av hvordan kravene ville utforme seg visuelt. Skissene ble laget digitalt og sendt til vurdering mellom hver skissering. Det ble deretter gitt tilbakemelding på hver skisse. På denne måten kunne Digimaker være med på å få kravene sine konsise og fullstendige før videre arbeid med prosjektet.

2.2.2 Analyse og funksjonalitet/design



Onsdag 09. februar 2022

Tid	Kunstgress	Mini	Gress
10:00			
10:30	Fenstad G2010/2011		
11:00			
11:30			
12:00		Trening G10	
12:30	FenstadJ2011/2012		
13:00			Trening J17
12:30			
13:00	Trening G13		
13:30			
14:00			
14:30			

Figur 2.2.2.1 - tabell

Her har vi en mal som ligner på den eksisterende malen som Digimaker har fra tidligere.



Onsdag 18. mai 2022

	17.00	17.30	18.00	18.30	19.00	19.30	20.00	20.30	21.00	21.30	22.00	22.30
Kunstgress			Trening G14/15									
			Trening G12									
					J11 7'er: FUVO-Skjettene 2							
											6.div : FUVO 2-Fet 2	
Mini												
Gress				Trening A-lag								

Figur 2.2.2.2 - figuren viser den eksisterende tabellen som Digimaker har for Stadiontavle

Forskjellen her er at vi har endret x- og y-aksene om hverandre, og dermed har man en tabell som heller er lengre langs y-aksen hvor tiden står. Dette fungerer bedre, med tanke på at treninger vanligvis pleier å være på ettermiddager og kvelder, og dermed ikke er så lang på y-aksen. Derfor blir det mer oversiktlig å ha tabellen slik som vi har skissert ut.

Kommentaren vi fikk fra Christian angående denne skissen var at vi heller skal fargekode basert på klubbene som skal spille kamp eller trenere, enn å farge basert på hvilkenbane det er. Denne kommentaren angående farger gjelder også på resten av skissene.



Onsdag 23.02

- Kunstgress 1
 - Fenstad G2010/2011 (10:30 - 11:30)
 - Fenstad J2011/2012 (12:30 - 13:00)
 - Trening G13 (13:00 - 14:00)
- Mini
 - Trening G10 (12:00-13:00)
- Gress
 - Trening J17 (13:00 - 14:00)

Figur 2.2.2.3 - listevisning utkast #1

Denne malen lagde vi for å vise all dataen i listeform. Denne skissen viser hvordan vi først så for oss at denne listevisningen kunne se ut. Dag og dato øverst, og liste basert på baner. Under banene er det en oversikt med lag som skal trenere og når de skal trenere, skrevet i parentes.



Onsdag 23.02

- **10:30 - 11:30**
 - Fenstad G2010/2011 (Kunstgress 1)
- **12:00-13:00**
 - Trening G10 (Mini)
- **12:30 - 13:00**
 - Fenstad J2011/2012 (Kunstgress 1)
- **13:00 - 14:00**
 - Trening J17 (Gress)
 - Trening G13 (Kunstgress 1)

Figur 2.2.2.4 - listevisning utkast #2

Videre tenkte vi at det hadde vært bedre å få tiden til å skille seg ut. Slik at man ser tiden tydeligere. Samtidig kan du se hvor treningen foregår, og kan deretter gå rett til treningsøkta i følge planen.

CB
Christian R. Bull
i går

Til: Dani Tran >

🔗
🚩

SV: Skisse Stadiontalve Sandefjord

Dette ser greit ut, men jeg tror eksempelet deres med visning Onsdag 23.02 ikke bør grupperes på Bane. Dette er en oversikt for at lagene fort skal finne ut hvor de skal trenе, så kolonne 1 bør være klokkeslett, som også bør være sorteringskolonne, deretter bør det vises lag/gruppe, aktivitet og bane/rom

Christian

[Se mer](#) ⬅

Figur 2.2.2.5 - tilbakemelding fra Christian angående skisse 2.2.2.4

Da vi sendte *Figur 2.2.2.4* til Christian, mente han at vi bør gjøre om listevisningen til en slags form for tabell. Kolonne 1 med klokkeslett, og kolonne 2 med lag/gruppe, aktivitet og bane/rom.



Onsdag 23.02	
10:30 – 11:30	Fenstad G2010/2011 (Kunstgress 1)
12:00 – 13:00	Trening G10 (Mini)
12:30 – 13:00	Fenstad J2011/2012 (Kunstgress 1)
13:00 – 14:00	Trening J17 (Gress)
13:00 – 14:00	Trening G13 (Kunstgress 1)

Figur 2.2.2.6 - listevisning utkast #3

Dermed endte vi opp med *Figur 2.2.2.6*, som er en listevisning satt opp i tabellform.



Uke 2	Mandag 11/1	Tirsdag 12/1	Onsdag 13/1	Torsdag 14/1	Fredag 15/1
8					
9					
10					
11	Kunstgress 1 10:30-11:30 Fenstad G2010/2011			Kunstgress 1 9:30-11:00 Fenstad J2011/2012	
12		Mini 12:00-15:00 Fenstad G2010/2011		Mini 11:30-13:00 Fenstad G2010/2011	
13	Kunstgress 1 13:30-15:30 Fenstad J2011/2012		Gress 12:00-13:00 TreningsJ17		Kunstgress 1 12:00-14:30 Fenstad J2011/2012
14					
15					
16	Gress 16:00-18:30 TreningsJ17		Kunstgress 1 14:30-15:30 Fenstad J2011/2012		
17					
18					
19					

Figur 2.2.2.7 - Kalenderform

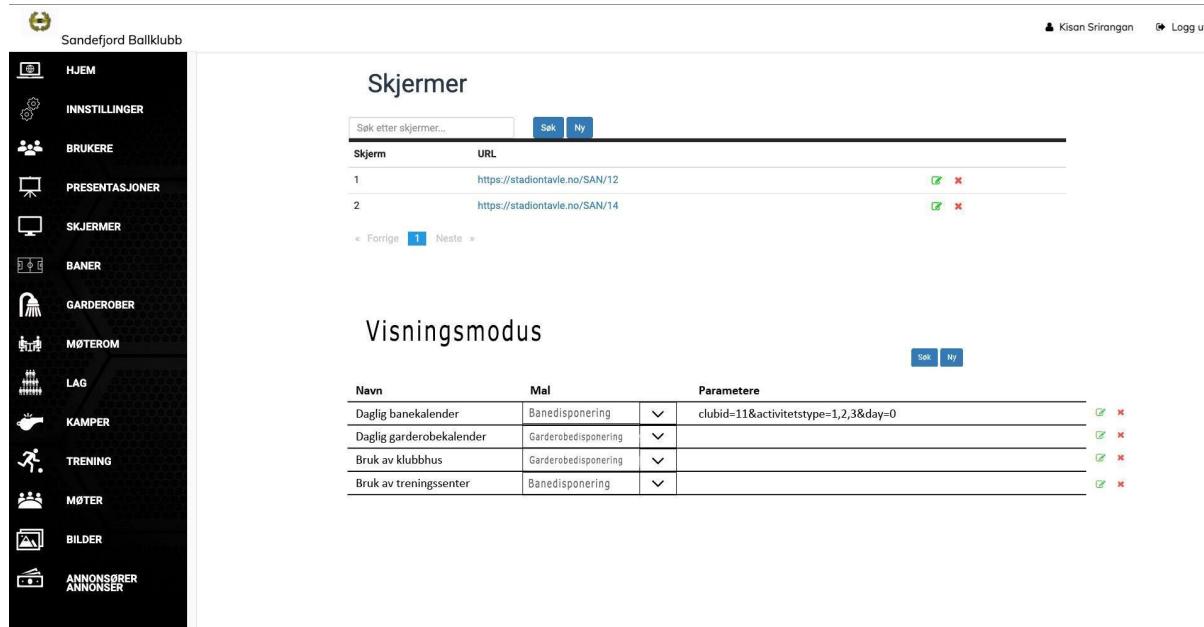
Vi hadde enda en idé for en mal, og den var inspirert av OsloMets timeplan for rom.

Ukekalender med trening satt opp under dag, og basert på tid.

Navn	Mal	Parametere
Daglig banekalender	Banedisponering	clubid=11&aktivitetstype=1,2,3&day=0
Daglig garderobekalender	Garderobedisponering	
Brak av klubbhus	Garderobedisponering	
Brak av treningssenter	Banedisponering	

Figur 2.2.2.8 - Aktivitetvisning

For å finne en måte på å kunne velge visning for aktiviteter, skisserte vi opp en ny side kalt ”Aktivitetsvisning” med en egen knapp på menyen. Her er det oversikt over navn på mal, type mal, og parametere.



The screenshot shows a navigation menu on the left with items like Hjem, Innstilling, Brukere, Presentasjoner, Skjermer, Baner, Garderober, Møterom, Lag, Kamper, Trening, Møter, Bilder, and Annonsører. The main content area is titled 'Skjermer' and shows a table of two screens:

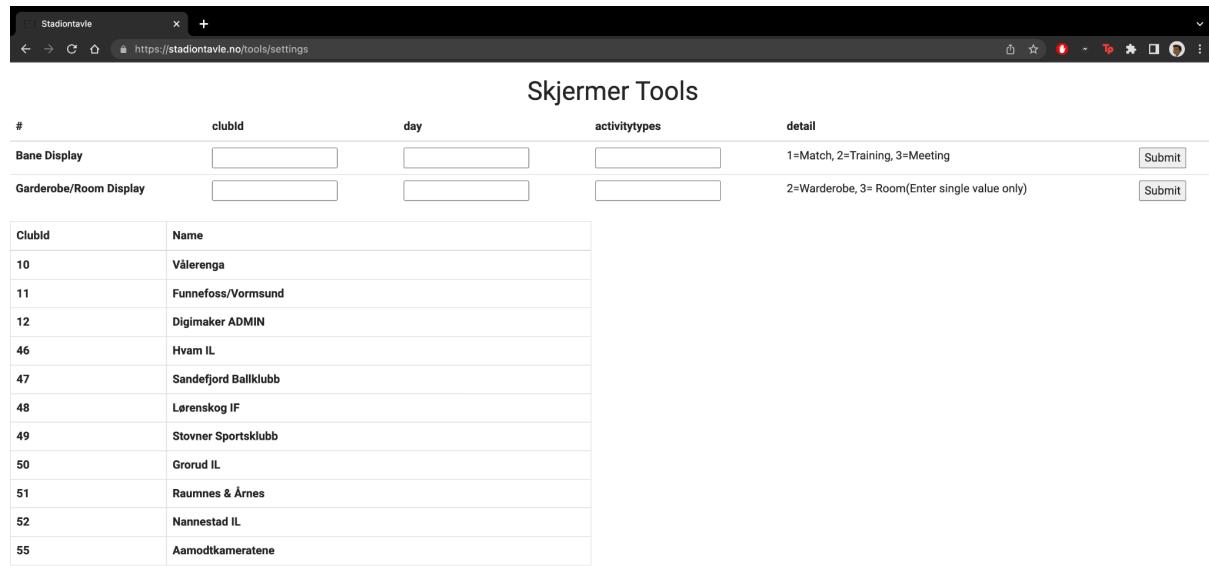
Skjerm	URL	
1	https://stadiontavle.no/SAN/12	
2	https://stadiontavle.no/SAN/14	

Below this is another table titled 'Visningsmodus' with four entries:

Navn	Mal	Parametere
Daglig banekalender	Banedisponering	clubid=11&aktivitetstype=1,2,3&day=0
Daglig garderobekalender	Garderobedisponering	
Bruk av klubbhus	Garderobedisponering	
Bruk av treningsenter	Banedisponering	

Figur 2.2.2.9 - Visningsmodus utkast #1

Vi kom senere frem til at det å lage en egen side for aktivitetsvisning er unødvendig med tanke på at visningen er en del av ”Skjermer”, da vi skal hente funksjoner fra ”Skjermer tools”-siden, som er en ekstern side av Stadiontavle-løsningen. Se *figur 2.2.2.10* angående den siden. *Figur 2.2.2.9* viser ”Visningsmodus” som skal ligge i ”Skjermer”-siden, under ”Skjermer”. I motsetning til på *Figur 2.2.2.8*, kan man her velge type mal fra en dropdown-menü for mal.

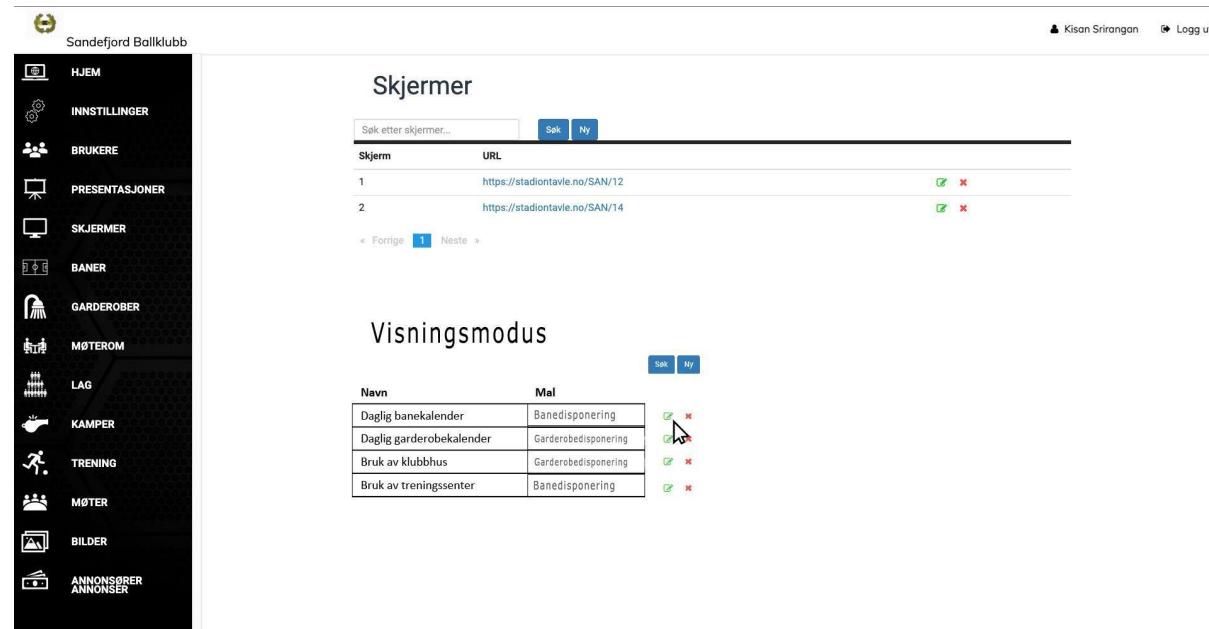


The screenshot shows the 'Tools' section of the Stadiontavle website. At the top, there are input fields for 'Bane Display' and 'Garderobe/Room Display' with dropdown menus for 'clubId', 'day', and 'activitytypes'. Below these are two buttons: 'Submit' and '2=Wardrobe, 3= Room(Enter single value only)'. A table lists various clubs with their names:

#	clubId	day	activitytypes	detail
Bane Display				1=Match, 2=Training, 3=Meeting
Garderobe/Room Display				2=Wardrobe, 3= Room(Enter single value only)
ClubId	Name			
10	Vålerenga			
11	Funnefoss/Vormsund			
12	Digimaker ADMIN			
46	Hvam IL			
47	Sandefjord Ballklubb			
48	Lørenskog IF			
49	Stovner Sportsklubb			
50	Gorud IL			
51	Raumnnes & Årnes			
52	Nannestad IL			
55	Aamodtkameratene			

Figur 2.2.2.10: Den eksterne siden for Stadiontavle kalt "Tools".

Her er "Skjermer tools"-siden som Digimaker har brukt for å vise frem maler. Det er denne sidens funksjoner vi skal ta med over til den interne administrasjonssiden.



The screenshot shows the internal tool for managing screens. On the left is a sidebar with navigation links: HJEM, INNSTILLINGER, BRUKERE, PRESENTASJONER, SKJERMER (selected), BANNER, GARDEROBER, MØTEROM, LAG, KAMPER, TRENING, MØTER, BILDER, and ANNONSØRER/ANNONSER. The main area is titled 'Skjermer' and shows a table of screens with URLs:

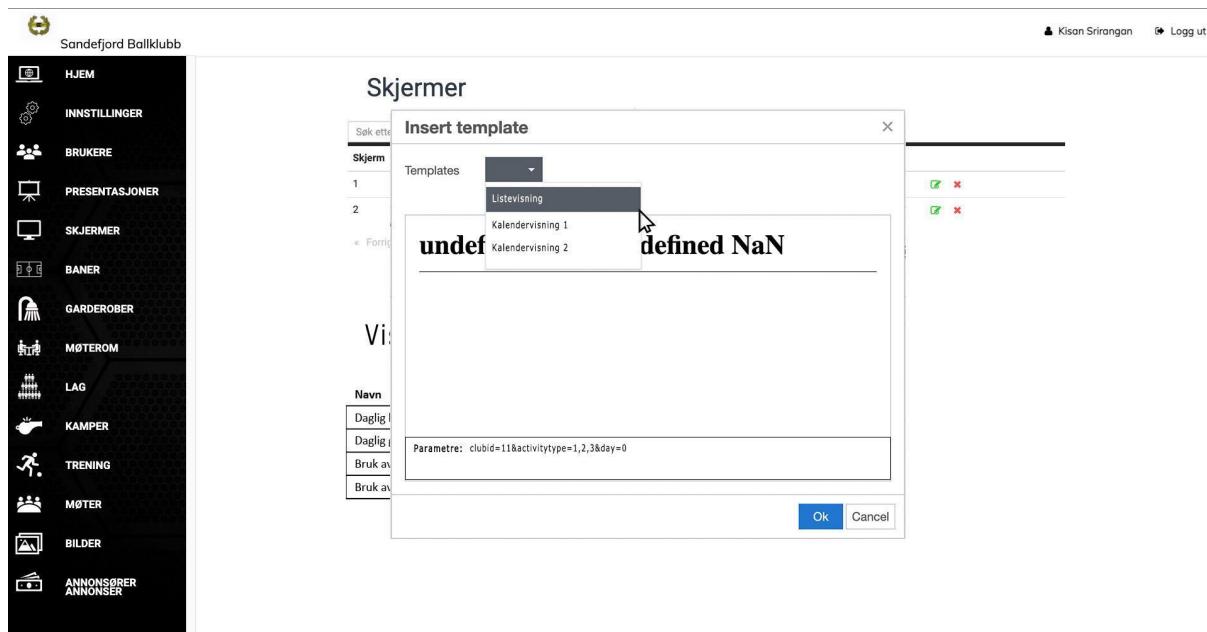
Skjerm	URL
1	https://stadiontavle.no/SAN/12
2	https://stadiontavle.no/SAN/14

Below this is a 'Visningsmodus' section with a table of display modes:

Navn	Mal
Daglig banekalender	Banedisponering
Daglig garderobekalender	Garderobedisponeering
Bruk av klubbhus	Garderobedisponeering
Bruk av treningsenter	Banedisponering

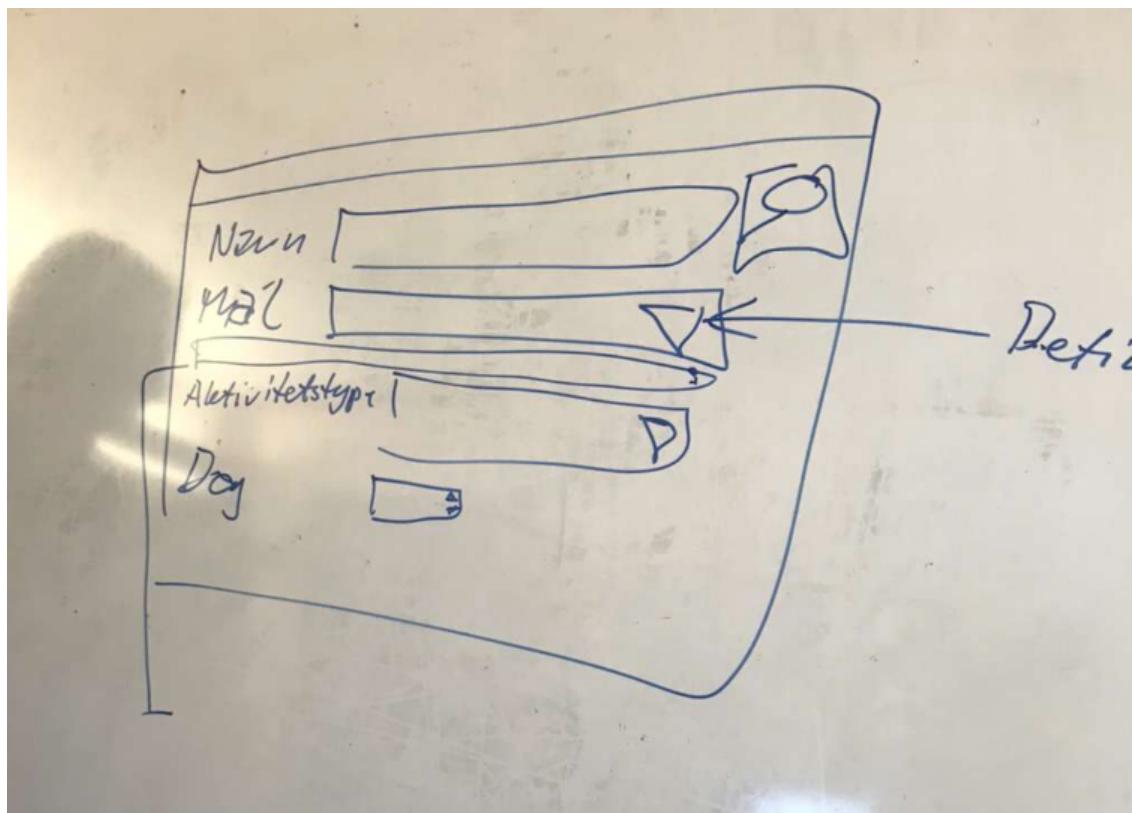
Figur 2.2.2.11 - Visningsmodus utkast #2.

På denne skissen har vi valgt å fjerne parametere, for å heller vise det som vises på Figur 2.2.2.12. Dropdown-menyen for valg av type mal er også fjernet, for å gjøre det som vi har valgt å gjøre det på Figur 2.2.2.12.



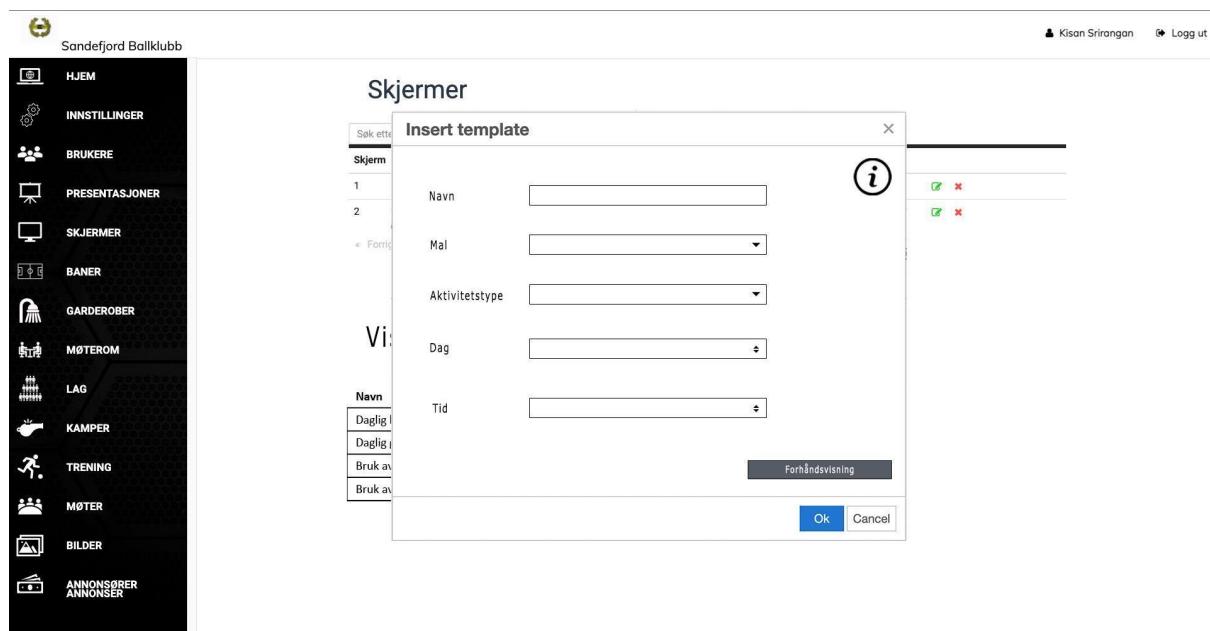
Figur 2.2.2.12 - Sett inn mal utkast #1.

Som dere kan se på *Figur 2.2.2.8, 2.2.2.9, 2.2.2.11*, har vi en grønn redigeringsknapp på høyre side av “Aktivitetsvisning” og “Visningsmodus”-menyene. Dette er for å sende videre til en “sett inn mal”-popup, som skal gi deg en oversikt over type maler, med mulighet for å velge en av de. Parameterne blir vist nederst på visningen.



Figur 2.2.2.13 - Sett inn mal utkast #2.

Denne skissen er skissert av Christian, etter å ha vist han Figur 2.2.2.12. Han mente at vi burde ha med navn, mal, aktivitetstype og dag i "sett inn mal"-popopen. Dette er for å gjøre det mer oversiktlig. Knappen ved siden av *navn*, er en informasjonsknapp som skal gi informasjon om de forskjellige type malene som kan velges.



Figur 2.2.2.14 - Sett inn mal utkast #3.

“Sett inn mal”-popugen endte opp med å bli skissert slik som på *Figur 2.2.2.14*, med navn, mal, aktivitetstype, dag og tid (tid valgte vi senere å fjerne). Popugen har også en informasjonsknapp som nevnt tidligere, og en “forhåndsvisning”-knapp for å vise en forhåndsvisning av den valgte malen.

Teknisk kartlegging/analyse

- Front-end: Angular + HTML/CSS
- Back-end: Laravel/PHP
- Database: MySQL

I utgangspunktet var løsningen satt av Digimaker. Det var lagt opp til at vi skulle videreutvikle eller forbedre løsningen som Digimaker hadde begynt på. Det var ønsket av oppdragsgiver at implementeringen av funksjoner skulle foregå enklast mulig hvor det ikke skulle være nødvendig å kode helt nye funksjoner, men heller bygge videre på og videreutvikle disse funksjonene som da ville resultere i flere valgmuligheter for visning av aktiviteter.

Fordelen ved denne løsningen var at kode enkelt kunne bli gjenbrukt og ikke ville kreve nye funksjoner ved hver layout. En annen fordel som anses som viktig i forhold til

rammebetingelser og bruk av Scrum er tidsbegrensingen. Det ville med denne løsningen om å bygge på eksisterende funksjoner, være lønnsomt tidmessig da vi etter gjennomføringsmetodikken måtte håndtere både store og små endringer raskt.

Det var på forhånd ikke bestemt hvordan brukergrensesnittet skulle være, men etter forslag fra oppdragsgiver kom vi fram til en felles avgjørelse på valg av design. Derimot var det allerede bestemt hva slags kodespråk i front-end/back-end som skulle bli tatt i bruk med tanke på den eksisterende løsningen.

Kapittel 3: Prosessdokumentasjon

3.1 Innledning

Dette kapittelet tar for seg hele utviklingsprosessen av vår bacheloroppgave våren 2022. I dette dokumentet går vi gjennom hele planleggingsfasen, utviklingsmodellene og metodikken vi har brukt under arbeidet med løsningen i dette prosjektet. Det vil bli gjort rede for selve gjennomføringen med detaljoversikt (fremdriftsplan), hvilke verktøy som har blitt brukt, hvilke forhold vi har jobbet etter og selve samarbeidet med Digimaker.

3.2 Forarbeid

Planleggingsfasen startet tidlig høst 2021 etter at gruppen ble sammensatt over nett, gjennom studentplattformen Canvas. Store deler av forarbeidet gikk over nett ettersom at studenter fortsatt var preget av Covid-19 situasjonen. Dette gjaldt også for mange av bedriftene vi kontaktet. Etter en felles vurdering i gruppen kom vi fram til at prosjektet Digimaker tilbød oss å jobbe med var den mest ideelle og tilrettelagte for oss som gruppe å utvikle. Etter at restriksjonene ble opphevet bestemte vi oss for at fysisk oppmøte skulle være en gang i uken hos Digimaker. Statusmøte ble satt hver mandag på nett.

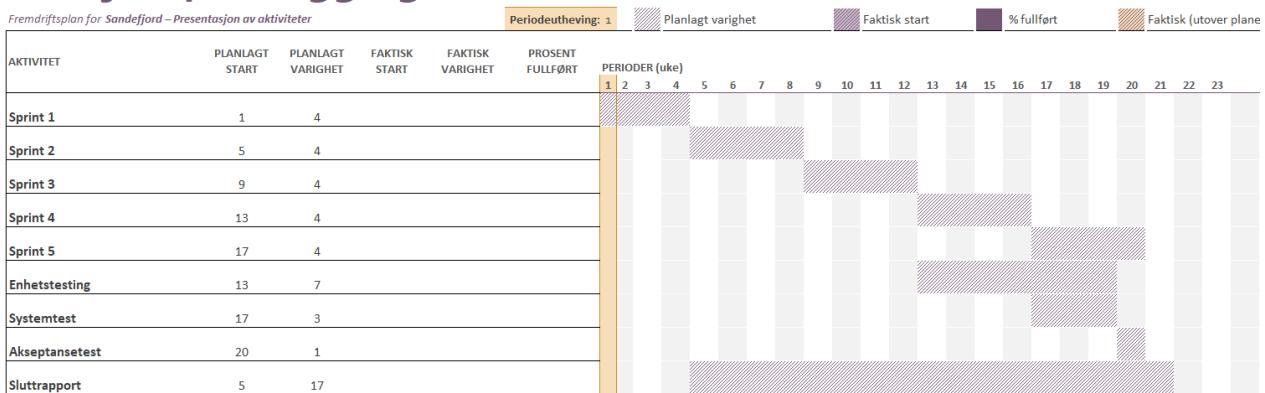
3.2.1 Hvorfor Digimaker

Digimaker hadde et behov for å videreutvikle Stadiontavle slik at det kunne bli et fullstendig informasjonssystem. Det er stadig større behov for at bedrifter skal kunne levere brukervennlige løsninger. I tillegg til dette, fokuserer også Digimaker på å skreddersy løsninger til kundene. For å kunne levere et fullstendig prosjekt er dette et av de viktigste punktene å forholde seg til. Som gruppe fikk vi en litt annen utfordring med å bygge videre på en eksisterende kode. Denne oppgaven utfordret oss på hvordan vi kunne gjøre informasjonstavlen/nettsiden mer brukervennlig og samtidig følge bestemte rammer. Vi sa oss derfor fornøyde med arbeidsmetoden og oppgaven Digimaker la frem for oss.

3.2.2 Arbeidsplan

Et av de første dokumentene som ble utarbeidet i gruppen var prosjektskissen. Det ble beskrevet kort hva prosjektet gikk ut på, litt om ønsket sluttprodukt, hvilket kodespråk som allerede var blitt brukt og som gruppen skulle bygge videre på. Etter å ha laget en grov skisse for prosjektet, diskuterte vi hvordan prosjektet skulle bygges opp. Vi diskuterte også hvilken arbeidsmetode som er mest effektiv med tanke på tid og struktur i utviklingsprosessen. Gruppen ble enige om å sette opp et Gantt-diagram⁵ for å illustrere prosjektets tidsplan og avhengigheter mellom de forskjellige arbeidsoppgavene. Det var viktig for oss å kunne forholde oss til det overordnede diagrammet. Et mer detaljert diagram ville komme senere i utviklingsprosessen, avhengig av det planlagte tidsløpet i prosjektet.

Prosjektplanlegging



3.2.2.1 Gantt - diagram (prosjektplanlegging)

⁵ En større visning av Gantt-diagrammet kan ses på [8.5 Fremdriftsplan](#).

3.3 Metodikk

Arbeidsmetodikk er en viktig del av ethvert prosjektarbeid, og avgjør på hvilken måte en gruppe tilnærmer seg det å bruke gruppemedlemmene kunnskaper og ferdigheter på best måte for å oppnå ønsket resultat.

Scrum er et smidig rammeverk ofte tilknyttet programvareutvikling, der brukeren kan adressere komplekse adaptive problemer, samtidig som de produserer produkter av høyest mulig verdi på en produktiv og kreativ måte⁶.

I dette underkapittelet skal vi gå mer igjennom Scrum og hvorfor vi valgte å jobbe med en smidig arbeidsmetode.

I starten av prosjektet utarbeidet vi en arbeidsplan, som skulle holde oversikt på hvor mye tid vi skulle bruke på hver oppgave, og når nye oppgaver skulle påbegynnes.

Etter møte med vår veileder fra Digimaker, kunne vi presisere hvilke krav som måtte dekkes i prosjektet. Vi kunne deretter legge en mer konkret plan for hvordan oppgavene skulle løses. Vi ble i gruppen enig om å benytte oss av en smidig arbeidsmetodikk, da det ville passe oppgaven best.

3.3.1 Fossefall vs smidig metode

Under planleggingsfasen av prosjektet ble både fossefall (Plandrevet modell - Der utviklingen foregår i veldefinerte faser) og smidig metode vurdert. I fossefallsmetoden er alle aktivitetene planlagt på forhånd, der hver oppgave skal være ferdig før en går videre til neste oppgave. Denne metoden gir en klar struktur på alt som skal gjøres, og kan fungere godt i prosjekter der en har en tydelig idé om alt som skal utføres for å oppnå ønsket resultat uten å måtte gjøre endringer underveis.

Med en smidig metode, gjøres planleggingen litt etter litt, og en ny fase i prosessen avgjøres ut i fra hva man oppnådde i forrige fase. På denne måten kan man tilpasse nye krav fra kunden underveis, og kunden blir mer involvert i de forskjellige fasene av prosjektet⁷.

⁶ (www.scrum.org, 2022)

⁷ (Lindsjørn, 2021b)

3.3.1.1 Smidig arbeidsmetode

Vi valgte en smidig arbeidsmetodikk, da oppgaven ikke var helt klar fra starten av og vi ønsket å kunne tilpasse planen vår etterhvert som prosjektet utviklet seg.

Vi ville ha en mer åpen tilnærming på hvordan prosjektet skulle løses, og hvem som skulle ha ansvar for de forskjellige delene. Vi ønsket videre at utviklingen skulle kunne bli evaluert kontinuerlig av Digimaker, der vi kunne jobbe tett for å få innspill på om kravspesifikasjonene ble dekket på ønsket måte slik at vi eventuelt kunne gjøre endringer underveis.

3.3.2 Scrum

Arbeidsmetoden Scrum er et sentralt tema innen systemutvikling, da det er en av det mest brukte metodene innen smidig utvikling. Gjennom studiet har vi fått kjennskap til ulike arbeidsmetodikker, og det ble avgjort at Scrum var det beste for vårt prosjekt.

Scrum er et smidig prosessrammeverk og gjennomføres på en [inkrementell](#) måte. Dette blir gjort med å fullføre oppsatte oppgaver under en sprint før en går videre til neste sprint.

I smidig utvikling er det rom for at det kan oppstå endringer i kravspesifikasjonen underveis, men dette kan tilpasses da det blir lagt en ny plan for hver ny sprint etter tilbakemelding fra produkteier. Etter hver sprint har man sjansen til å endre productbackloggen, som er en liste med alle oppgavene som gjenstår i prosjektet.

3.3.2.1 Implementasjon av Scrum

Scrum som smidig prosessrammeverk gjennomføres på en inkrementell måte. Med dette rammeverket følger det med en viss struktur, og terminologi. Scrum innebærer at vi har et utviklingsteam, en product backlog, en produkteier, en [scrum-master](#) som holder scrum-møter, og at prosjektet skal foregå i sprints.

Scrum team:

Et Scrum team består av produkteier, Scrum-master og et utviklerteam. Teamet skal bestå av medlemmer som er selvstendige i sitt arbeid, og som har en tverrfaglig kompetanse.

Hvordan oppgavene kan løses på best måte, avgjøres av teamet.

Produkteier:

Hovedansvaret i denne rollen definerer kravene til det som skal utvikles, og skal kontinuerlig passe på at oppgavene løses med hensyn til en liste med prioriteringer.

Digimaker er produkteier i dette prosjektet.

Scrum-Master:

En Scrum-master er ansvarlig for at teamet jobber etter de smidige prinsippene innen Scrum, og at teamet følger planen som har blitt satt. Denne rollen fungerer som mellomleddet til produkteier og utviklingsteamet. Vedkommende er ansvarlig for å holde Scrum-møter, og passe på at prosjektet har en bra prosessdokumentasjon.

I dette prosjektet valgte vi Dani som Scrum-master, som også er en del av utviklingsteamet.

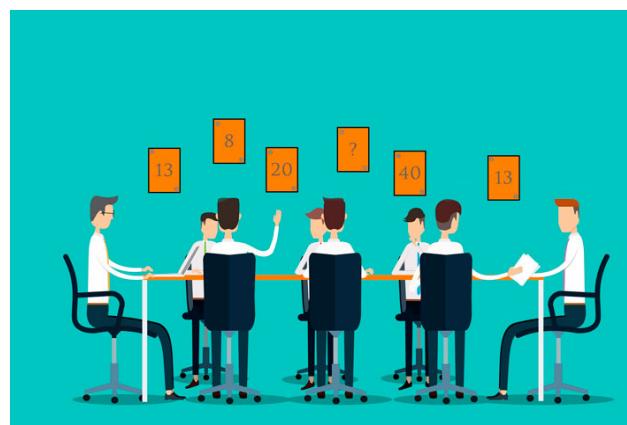
Utviklingsteam:

Utviklingsteamet er en selvorganisert gruppe der alle har rollen som utvikler. Dette teamet besto av fire studenter, der det var en representant for hver studieretning innen IT på OsloMet.

3.4 Planning Poker

For hver av oppgavene som skulle løses var det viktig at teamet kunne estimere hvor mye tid oppgaven ville ta. For at denne planleggingen skulle bli enklere valgte vi å bruke planning poker. Dette utføres med at alle i teamet har et sett med kort som inneholder tall. Når en kravspesifikasjon, funksjon eller oppgave er valgt skal alle i teamet velge et kort som de mener estimerer tidsbruken i antall dager som behøves for å kunne ferdigstille dette.

Alle snur kortet samtidig for hverandre, og hvis alle er enig blir estimatet for tid til den spesifikke oppgaven fastsatt.



Figur 3.4.1: Eksempel på planning poker. Hentet fra agilebuddha.com

3.5 Sprinter

En sprint starter med å holde et Scrum-møte der teamet blir enig om hvilke oppgaver som skal utføres og hvordan oppgavene skal fordeles.

Under selve sprinten har man standup-møter eller daily scrum. Dette gjøres for at en skal ha oversikt på hvordan teamet ligger an.

Da vi alle hadde forskjellige timeplaner der det var vanskelig å finne en tid hver dag som passet for alle, bestemte vi oss for å holde to faste ukentlige møter. En med bare utviklingsteamet på mandager, og så et møte med teamet og produkteier på torsdager.

Utover det planla vi møter etter behov under hver sprint.

Product backlog:

Product backlog er en liste med alle de oppgavene som skal bli gjennomført i løpet av prosjektet. Listen inneholder alle krav, endringer og feilrettinger. Kravene i denne listen blir satt etter prioritet, og hjelper produkteier med å holde oversikt på progresjonen til utviklerteamet.

Sprint backlog:

Dette er en liste over alle oppgavene som skulle utføres under en sprint. Hensikten med dette er å gi teamet oversikt over oppgavene som har blitt utført.

Sprint review:

Målet med å holde en sprint review er å gå igjennom resultatene fra sprinten og se hva som kan forbedres til neste sprint. Under disse møtene ble progresjonen diskutert med produkteier, og nye krav eller endringer som skulle implementeres under neste sprint ble klargjort.

Sprint retrospektiv

Sprint retrospektiv er et møte teamet har etter hver Sprint review. Her blir oppgavene som er utført oppsummert. Her dokumenterte vi hva som hadde gått bra under sprinten, og hva som kunne forbedres.

3.5.1 Sprint 1

Varighet: 4 uker (Uke 1-4)

Mål:

Målet for første sprint var at utviklerteamet skulle sette opp og bli kjent med de forskjellige verktøyene som skulle brukes i prosjektet. Teamet skulle også finne ressurser for å lære mer om kodespråkene som skulle brukes under utviklingen av den digitale løsningen for Digimaker. Dette var også en forutsetning for å være klar til neste sprint. Videre skulle teamet sette av tid til prosjektplanlegging der en arbeidsplan med de div. sprintene ble opprettet, samt at forprosjektrapporten skulle skrives ferdig og innleveres til skolen. Det skulle også planlegges faste møter internt i teamet og med Digimaker.

Deltagere:

- Dani Tran (Scrum-master)
- Kisan Srirangan
- Linda Kadrijaj
- Marthe V. Marstrander

Oppgaver:

- Finne ressurser for Laravel og Angular
- Opprette kanal på Discord
- Opprette backlog på google drive
- Lage arbeidsplan for prosjektet
- Skrive ferdig forprosjektrapport med innlevering til OsloMet
- Planlegge faste møter

Oppsummering:

Den første sprinten gikk som planlagt. Alt av verktøy ble satt opp etter planen, og de relevante ressursene for kodespråkene ble innhentet. Teamet samarbeidet godt med å skrive forprosjektrapporten og denne ble levert i god tid. Første arbeidsplan for prosjektet ble ikke satt opp helt korrekt med hensyn til å sette opp planen ift. Scrum sprinter, så en ny

arbeidsplan ville bli utarbeidet under neste sprint. Det er denne arbeidsplanen vi endte med å bruke i sluttrapporten. Teamet hadde ikke fått helt rutine på å ha faste møter under denne sprinten.

Retrospektiv:

Hva gikk bra?

- Arbeidet med forprosjektrapporten
- Oppsettet av div. verktøy
- Alle oppgavene ble ferdig under sprinten
- Oppsettet av backlog

Hva kunne vært bedre?

- Oppsettet av den første arbeidsplanen
- Gjennomføringen av faste møter

Hva skal forbedres?

- Planleggingen av antall fremtidige sprints
- Rutinen på møter

3.5.2 Sprint 2

Varighet: 4 uker (Uke 5-8)

Mål:

Målet for andre sprint var at utviklerteamet skulle bli kjent med kodespråket, på den eksisterende løsningen til Digimaker. Ved bruk av ressursene teamet innhentet i første sprint skulle teamet ha individuell opplæring, der hovedfokuset var å få en bedre forståelse av Laravel og Angular. Teamet skulle ha møte med Digimaker for å få en mer helhetlig og omfattende kravspesifikasjon, samt få tilgang til Digimaker sin [repository](#) på Github der den eksisterende løsningen lå. Malen for sluttrapporten skulle også bli satt opp på Google Docs.

Deltagere:

- Dani Tran (Scrum-master)
- Kisan Srirangan
- Linda Kadriaj
- Marthe V. Marstrander

Oppgaver:

- Bli kjent med de forskjellige kodespråkene Laravel (PHP) og Angular (TypeScript)
- Ha møte med Digimaker for å få en mer omfattende kravspesifikasjon
- Få tilgang til koden av Digimaker via repository på GitHub
- Sette opp strukturen på sluttrapporten på Google Docs

Oppsummering:

Teamet fikk en god innføring i kodespråkene, men innlæringen måtte fortsette videre da ingen på teamet hadde nok erfaring med språkene på dette tidspunktet. Møtet med Digimaker var veldig godt da temaet fikk en bedre forståelse av hvilke funksjoner Digimaker ønsket i den nye løsningen. Alle fikk tilgang til den eksisterende koden, og teamet er klar for neste sprint der utviklingsmiljøet skal settes opp. Sluttrapporten ble påbegynt, og alle fikk tilgang til å kunne redigere rapporten. Teamet har fått en god rutine på å ha faste møter internt, og med Digimaker.

**Retrospektiv:**

Hva gikk bra?

- Alle de planlagte oppgavene ble ferdig under sprinten.
- Møtet med Digimaker
- Rutinen på faste møter

Hva kunne vært bedre?

- Opplæringen av kodespråkene

Hva skal forbedres?

- Kompetansen på kodespråkene innad i teamet

3.5.3 Sprint 3

Varighet: 4 uker (Uke 9-12)

Mål:

Målet for tredje sprint var å sette opp utviklingsmiljøet på teamet sine individuelle datamaskiner, dette for å kunne utvikle den nye løsningen i den eksisterende løsningen som lå på GitHub. Videre skulle teamet lage skisser i Figma etc, for å ha prototyper på hvordan den nye løsningen skulle se ut. Etter skissene er ferdige, skulle det bli holdt nytt møte med Digimaker for å se om skissene tilfredsstilte kravspesifikasjonene. Arbeidet med skissene ville fortsette etter møtet hvis Digimaker ønsket endringer.

Deltagere:

- Dani Tran (Scrum-master)
- Kisan Srirangan
- Linda Kadrijaj
- Marthe V. Marstrander

Oppgaver:

- Sette opp utviklingsmiljø
- Installere all nødvendig programvare
- Lage skisser
- Ha møte med Digimaker for å godkjenne skissene

Oppsummering:

Teamet arbeidet på både PC og Mac, så oppsettet av utviklingsmiljøet tok lengre tid enn beregnet. Vi måtte ha hjelp av en av utviklerne fra Digimaker før alle hadde det oppsettet de trengte for å kunne jobbe med koden.

Arbeidet med skissene gikk bra, og etter noen møter med Digimaker hadde vi prototyper som skal være utgangspunktet for utviklingen av løsningen.

Retrospektiv:

Hva gikk bra?

- Arbeidet med skissene
- Møter med Digimaker for å få godkjent skisser

Hva kunne vært bedre?

- Tidsberegningen for oppsettet av utviklingsmiljøet

Hva skal forbedres?

- Kompetansen for å bruke utviklingsmiljøet

3.5.4 Sprint 4

Varighet: Uke 4 (Uke 13-16)

Mål:

Målet for fjerde sprint var å begynne på utviklingen av koden, da hovedsakelig frontend delen av løsningen med skissene fra forrige sprint som utgangspunkt. Det skulle utføres enhetstest kontinuerlig under denne utviklingen. Teamet skulle også bli kjent med backend delen, og databaseoppsettet.

Arbeidet med sluttrapporten skulle begynne, og alle delene som kunne skrives skulle fullføres under denne sprinten.

Deltagere:

- Dani Tran (Scrum-master)
- Kisan Srirangan
- Linda Kadrijaj
- Marthe V. Marstrander

Oppgaver:

- Begynne utviklingen av frontend
- Utføre enhetstest
- Få kjennskap til back-end og databasen
- Skrive på sluttrapporten

Oppsummering:

Utviklingen av frontend ble begynt, og ville fortsette under neste sprint. Enhetstest har blitt utført for å unngå oppsamling av feil i koden. Backend delen vil kreve videre opplæring av teamet for å kunne gjennomføres på en effektiv måte. Det vil settes opp mer tid til opplæring under neste sprint.

Arbeidet med sluttrapporten ble satt igang, og store deler av rapporten var på dette tidspunktet skrevet ferdig.

Retrospektiv:

Hva gikk bra?

- Starten på utviklingen av frontend
- Enhetstesten
- Arbeidet med sluttrapporten

Hva kunne vært bedre?

- Forståelsen på backend delen innad i teamet

Hva skal forbedres?

- Kodekompetansen i teamet

3.5.5 Sprint 5

Varighet: 4 Uker (Uke 17-20)

Mål:

Målet for femte sprint var å ferdigstille arbeidet med frontend delen, samt fullføre backend delen av den nye løsningen. Det skulle brukes tid på å øke back-end kompetansen innad i teamet, og dette skulle gå parallelt med selve utviklingen.

Når koden var ferdig, skulle det på nytt utføres en enhetstest. Hvis denne ble godkjent skulle det utføres en systemtest. Når løsningen var ferdig skulle teamet ha et møte med Digimaker der løsningen skulle vises fram, samt at det skulle gjennomføres en akseptansetest.

Resultatene skulle så dokumenteres i sluttrapporten.

Deltagere:

- Dani Tran (Scrum-master)
- Kisan Srirangan
- Linda Kadrijaj
- Marthe V. Marstrander

Oppgaver:

- Ferdigstille frontend
- Ferdigstille backend
- Enhetstest
- Systemtest
- Akseptansetest
- Dokumentere resultater i sluttrapporten

Oppsummering:

Teamet brukte mye tid på å sette seg inn i backend og database delen av kodingsprosessen, men løsningen vi hadde valgt for oss viste seg å være tungvint. Dermed måtte vi inngå kompromisser, og nedprioritere noen av funksjonene vi hadde tenkt til å utvikle. Imidlertid

klarte vi å komme frem med en løsning som Digimaker kan videreutvikle. Det ble også gjort testing av kode, og resultatene fra testingen kommer i [Kapittel 5: Testdokumentasjon](#).

Retrospektiv:

Hva gikk bra?

- Enhetstesten og systemtesten

Hva kunne vært bedre?

- Div. backend funksjoner
- Resultatet av akseptansetesten

Hva skal forbedres?

- Backend delen av løsningen
- Visningsmodus
- Lagring av data valgt av bruker
- Automatiseringen av ID

3.6 Plattform og verktøy

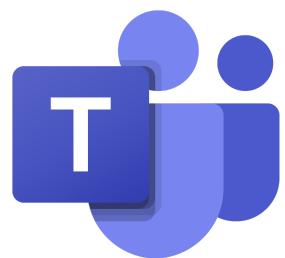
3.6.1 Discord

Et viktig verktøy som ble brukt er Discord. Denne digitale distribusjonsplattformen ble brukt av teamet for å sette sammen gruppen og bli kjent under planleggingsfasen (forarbeidet). Det ble delt dokumenter, kilder og annet nødvendig informasjon om bacheloroppgaven. Videre i prosjektet brukte vi denne plattformen til å ha møter internt i gruppen, og for å ha oversikt over kommunikasjonen som foregikk over chat.



3.6.2 Microsoft Teams

Ettersom at en del av møtene gikk over nett på grunn av digital undervisning og lite oppmøte på universitetet, valgte vi å bruke Microsoft Teams som hovedplattform for kommunikasjon. Det ble hovedsakelig opprettet ukentlige møter med veileder i Digimaker, men også mellom oss i gruppen. Vi lagde også et eget Team slik at vi hadde styr på informasjon som ble delt under møtene.



3.6.3 GitHub

Selve systemutviklingsprosessen førte vi opp i GitHub. Dette er det mest brukte verktøyet gjennom studiet vårt og dermed den enkleste måten å holde kontroll på kontinuerlige endringer i koden. GitHub gjør det tidsmessig besparende å jobbe med et prosjekt med tidsramme, da vi alle jobber med det samme prosjektet til forskjellig tid og sted. Det er enkelt å slå sammen hver enkelt branch (et arbeidsområde per medlem), til et fullstendig prosjekt (repository).





3.6.4 Google Drive

Gruppen kom frem til at Google Drive ville være det beste alternativet å samarbeide på. Allerede under forarbeidet opprettet vi en felles mappe der vi lastet opp og opprettet dokumenter som var viktige og relevante under arbeidet med bacheloroppgaven. Dette inkluderte alle rapporter, skisser, diagram, arbeidsplan, product backlog og selve sluttrapporten. Mappen er lett tilgjengelig uansett hvilken enhet som blir brukt så lenge gruppemedlemmet har en Google-konto. Det er mulig å jobbe samtidig på dokumenter og endringene synkroniseres raskt underveis.



3.7 Refleksjon

3.7.1 Refleksjon av arbeidsmetodikk og sprinter

I startfasen av sprintene var Scrum i praksis ukjent for teamet, da ingen av oss hadde erfaring med å jobbe med smidig arbeidsmetodikk fra tidligere prosjekter. Det å estimere hvor mye tid hver enkel oppgave behøvde, samt å finne den beste fordelingen av oppgaver ift. kompetansen i gruppen ble bedre etter første sprint. Istedetfor daglige standup møter hadde teamet to fastsatte møter i uken, der vi heller avtalte å møtes utover det etter behov. Vi valgte å gjøre det på den måten da forskjellige timeplaner gjorde det vanskelig å finne et tidspunkt der alle var tilgjengelig hver dag.

Etter hver sprint holdt vi sprint review og sprint retrospektiv møter både internt i gruppen og med oppdragsgiver, der vi gikk igjennom resultatene og la plan for neste sprint.

Det å dele prosjektet inn i mindre deler ved bruk av sprinter fungerte bra for teamet, da det ga bedre struktur på arbeidsplanen og oppgavene som skulle løses.

Kodespråkene viste seg å være den største utfordringen for gruppen da ingen hadde mye erfaring med disse fra tidligere. Frontend delen av arbeidet gikk som planlagt, mens backend delen ble ikke komplett under den siste sprinten.



3.7.2 Refleksjon av plattform og verktøy

De utvalgte verktøyene gjorde at kommunikasjonen og samarbeidet under arbeidsprosessen fungerte på en god måte. Discord ble en plattform der meste av kommunikasjonen innad i teamet foregikk, samt at det var en oversiktlig måte å dele linker eller annet materiale med hverandre. Da Discord gir mulighet til å opprette både individuelle møterom og chatterom, ble samtalene mer organiserte.

Microsoft Teams ble for det meste brukt til møtene teamet hadde med oppdragsgiver, da dette var den plattformen Digimaker var vant til å bruke. Møtevirksomheten fungerte godt ved bruk av Teams, da pandemien satte stopper for fysiske møter i startfasen av prosjektet. Vi gikk senere over til å ha møter på Digimaker sine lokaler på Helsfyr i Oslo.

GitHub var et sentralt verktøy i utviklingsfasen da alt vi gjorde av endringer i den eksisterende koden til Digimaker ble lagt i egen [branch](#) i Stadiontavle sitt repository. Dette for at endringene vi gjorde ikke skulle skape problemer i den eksisterende koden før en av utviklerne på Digimaker godkjente arbeidet.

Dette ga også teamet bedre oversikt over alle endringer som ble gjort i koden, og gjorde samarbeidet lettere under utviklingen. Vi hadde alle erfaring med GitHub fra tidligere prosjekter, og vi fikk verdifull erfaring fra å bruke det her i samarbeid med en oppdragsgiver.

Google Drive var verktøyet som ble brukt mest til å samarbeide om sluttrapporten, samt å ha oversikt over hvilke oppgaver som skulle gjøres under en sprint. Vi lagde et Google dokument der oversikt over oppgaver og gjøremål ble satt opp. Vi valgte å gjøre det på denne måten da teamet hadde god erfaring med denne metoden av arbeidsfordeling fra tidligere. Denne løsningen fungerte også som product backlog til prosjektet.

3.8 Avslutning

I dette bachelorprosjektet har vi fått lære mye om rollen som utviklere og konsulenter, samt hvordan det er å arbeide smidig. Det å opprettholde god kommunikasjon og godt samarbeid innad i et team, samt med oppdragsgiver har gitt oss mulighet til å utvikle oss både faglig og profesjonelt som fremtidige arbeidstakere.

Vi har hatt god støtte i de fra Digimaker under hele prosessen, samtidig som vi har fått stor frihet til å komme med idéer og utvikle løsningen slik vi ønsket. Det å være en del av utviklingsprosessen til den nye løsningen har gitt oss innblikk i hvordan en systemutviklingsprosess foregår, samt hvor nyttig et prosessrammeverk er for å skape en oversiktig og effektiv arbeidsprosess.

Kapittel 4: Produktdokumentasjon

4.1 Innledning

I dette kapittelet beskriver vi de tekniske aspektene til webapplikasjonen, altså dens struktur og funksjonalitet. I tillegg blir det presentert hvilke teknologier og verktøy som har vært brukt, og i hvilken sammenheng, for å bygge løsningene. Dette er ment for å gi vedkommende en dyp og oversiktlig forståelse av applikasjonen, som kan gjøre det enklere å markedsføre produktet videre eller drive brukerstøtte.

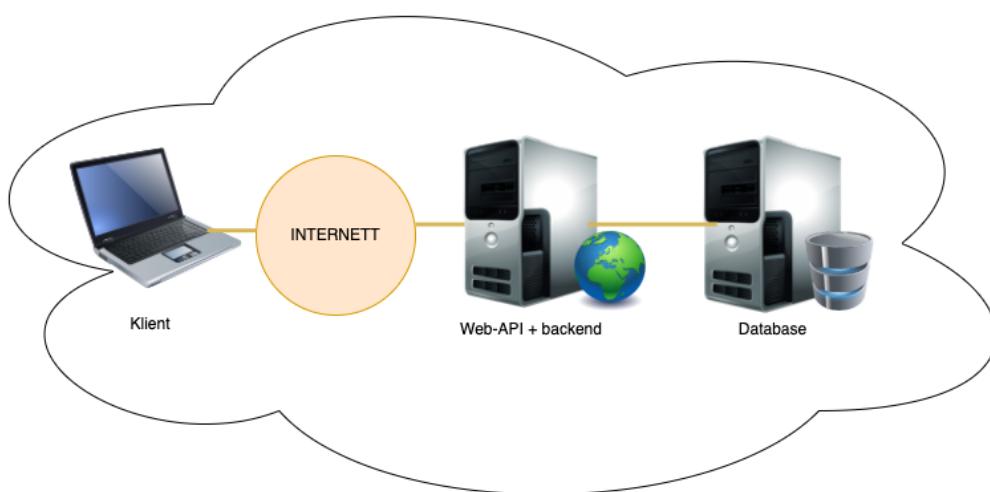
Under videreutviklingen til webapplikasjonen, blir både kode av [frontend](#) og [backend](#) implementert. Forskjellen på disse er at frontend-utvikling er en stil med dataprogrammering som setter søkelys på koding og opprettelse av elementer og funksjoner på en nettside som vil bli sett av brukeren. Den sørger for at de visuelle aspektene ved et nettsted er funksjonelle, vi kan også kalle dette klientsiden av webapplikasjonen. Mens backend (serversiden) derimot fokuserer på den siden av nettstedet brukeren ikke kan se. Altså det som gjør nettstedet interaktivt. Vi vil derfor ta for oss både filstrukturen til frontend og backend, og gi en beskrivelse på de viktigste komponentene, samt de viktigste hovedfunksjonalitetene i webapplikasjonen.

4.2 Systemarkitektur

Systemarkitektur er en modell som gir et innblikk på strukturen og funksjonene til flere komponenter og undersystemer. I følge arkitekturprinsipper har vi valgt å dele det inn i overordnet arkitektur, kommunikasjon og logisk arkitektur.

4.2.1 Overordnet arkitektur

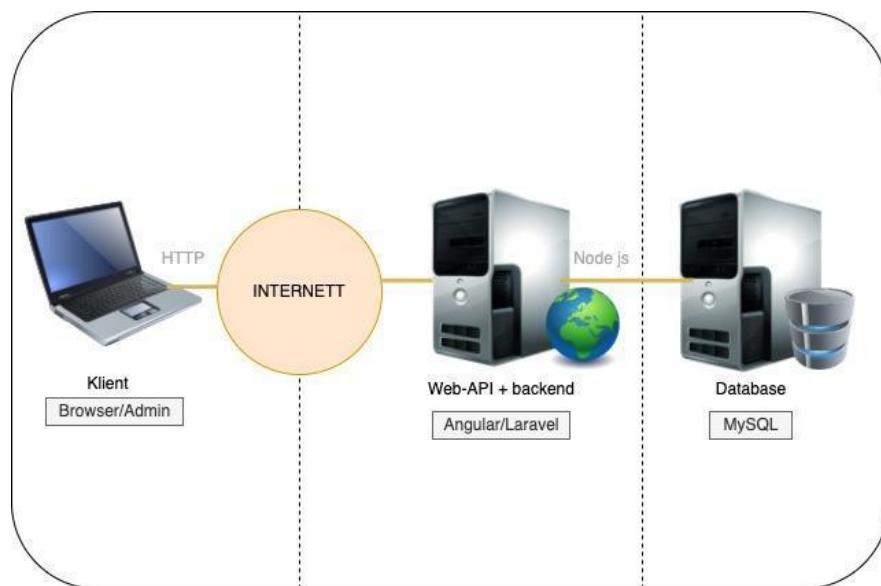
Figur 4.2.1.1 viser den fysiske arkitekturen til systemet. Dette er den overordnede flyten i løsningen som er en klient-server arkitektur, som de fleste webapplikasjoner følger i dag. Løsningen vår består av at klienten og serveren kommuniserer med hverandre ved hjelp av et web-[API](#). API-et gir en standardisert måte for to applikasjoner å sende data frem og tilbake, så når brukeren utfører en interaksjon i brukergrensesnittet vil klienten sende en forespørsel til APIet for å hente informasjon eller oppdatere noe i applikasjonen. Serveren som inneholder ressurser som klienten ønsker, vil da motta dette klientforespørslet. Deretter samhandler disse to enhetene, og returnerer en respons uten å gi klienten direkte tilgang til innhold som er lagret i [databasen](#).



Figur 4.2.1.1: Den fysiske arkitekturen til systemet

4.2.2 Kommunikasjon

Figur 4.2.2.1 viser en standart API arkitekturdiagram, hvor designet er basert på vår oppgave. API brukes til å koble sammen enheter og tillate deling av data, samtidig utveksle data mellom server og klient⁸. Den grå skriften viser hvordan frontend og backend kommuniserer med hverandre. Samt viser de grå tekstboksene kodespråket som blir brukt og rammeverket for hver av de forskjellige lagene i webapplikasjonen.



Figur 4.2.2.1: Kommunikasjonen innad i systemet

4.2.3 Logisk arkitektur

Målet med logisk arkitektur er å styre de funksjonelle kravene. Under beskrives den logiske arkitekturen til både frontend og backend.

4.2.3.1 Frontend

Alt brukeren ser på webapplikasjonen er lagret som et objekt, kodet i Angular, som er et frontend-rammeverk med åpen kildekode. Det er et JavaScript basert Typescript programmeringsspråk som eliminerer unødvendige funksjoner og kode for å sikre letttere og

⁸ (Jain, 2019)

raskere applikasjoner⁹. For å kjøre de forskjellige funksjonene i applikasjonen må vi sette i gang en action. I dette tilfellet blir det brukt et [REST-API](#) fra Digimaker for at samhandlingen mellom klienten og serveren skal gi respons.

4.2.3.2 Backend

Backend-arkitekturen består av fire deler: API, backend applikasjon, server og database. APIet sender klientforespørslene til serveren som kjører backendapplikasjonen, og henter den spesifikke informasjon fra databasen og Laravel. Databasen inneholder samling av all relatert data fra webapplikasjonen og Laravel er backendapplikasjonen. Når APIet har fått hentet informasjonen som trengs, vil det bli lagret og tilsendt tilbake til klienten, og brukeren vil da få en respons fra webapplikasjonen.

Eksempel på dette er når brukeren skal lage sin egen visning på administrasjonssiden. Da kommer brukeren til å hente en ferdiglaget visning fra databasen og legge til sine egne parametere for at visningen skal tilpasse brukerens behov, så det blir deres unike visning. Dette vil bli lagret og endret i databasen ved hjelp av backendapplikasjonen. Kommunikasjonen mellom dem er ved hjelp av APIet.

4.3 Teknologier og verktøy

4.3.1 Angular

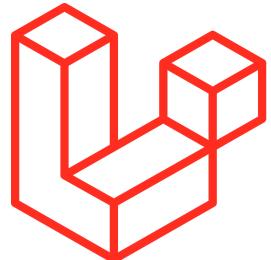
Angular er en plattform og et rammeverk for å bygge enkeltside klientapplikasjoner ved hjelp av HTML og TypeScript. Det er skrevet i TypeScript, og lar oss implementere kjerne og valgfri funksjonalitet som et sett med TypeScript-biblioteker som du importerer til applikasjonene dine. Meningen med Angular er å bygge interaktiv og dynamiske enkeltsideapplikasjoner gjennom sine avgjørende funksjoner som RESTful API-håndtering.



⁹ (Luzniak, 2021)

4.3.2 Laravel

Laravel brukes først og fremst til å bygge tilpassede nettapplikasjoner ved hjelp av PHP. Det er et rammeverk som håndterer ting som er tidkrevende å bygge selv, som eksempel HTML-maler og autentisering.



4.3.3 MySQL

MySQL er et databasehåndteringssystemer for relasjonsdatabaser utviklet av Oracle som er basert på det strukturerte populære spørrespråket, SQL. MySQL er en av de mest gjenkjennelige teknologiene i det moderne store økosystemet når det gjelder data. Med MySQL kan man umiddelbart bygge raske, kraftige og sikre datalagringssystemer.



4.3.4 Visual Studio Code

Visual Studio Code er en gratis åpen kildekode editor av Microsoft, og gir støtte til utviklere for oppgaver som feilsøking, oppgavekjøring og versjonskontroll. Programvaren tar sikte på å bidra til å gi utviklere en lettare jobb når det kommer til rask kode bygging og feilsøking.



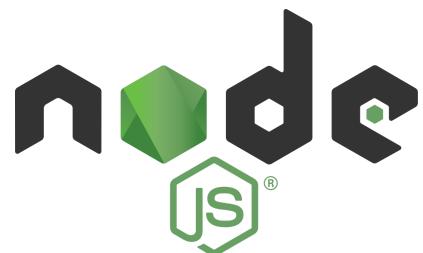
4.3.5 PhpStorm

PhpStorm er en kildekode editor bygget av selskapet JetBrains. Denne editoren gir støtte for PHP, HTML, JavaScript. Et par medlemmer i gruppa valgte også å bruke denne editoren ved siden av Visual Studio Code, siden PhpStorm er mer oversiktlig og bedre egnet for PHP.



4.3.6 Node JS

Node JS er en digital plattform bygget på Chromes JavaScript kjøretid, for å gjøre det enkelt å bygge raske og skalerbare nettverksapplikasjoner. Det er perfekt for dataintensive sanntidsapplikasjoner som kjører på tvers av distribuerte enheter.



4.3.7 Composer

Composer er en administrator som er avhengig av PHP. Den laster ned, installerer og laster inn avhengigheter du spesifiserer i en composer.json fil. Composer gjør det enklere å holde andre biblioteker utenfor ditt oppbevaringssted, som betyr at bare applikasjonskoden går i ditt *git repository*. Samtidig er det enklere å vedlikeholde alle nødvendige biblioteker, fordi du kan enkelt kjøre Composer oppdatering for å få nye versjoner av biblioteker.



4.3.8 phpMyAdmin

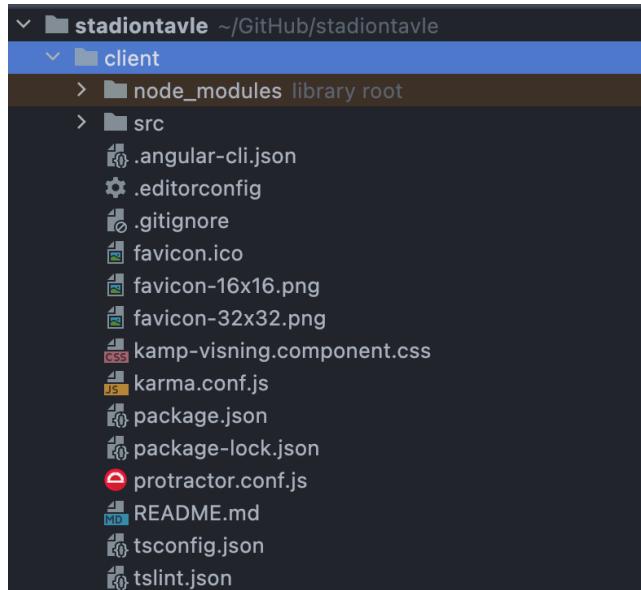
phpMyAdmin er et åpent og gratis programvareverktøy på nettet skrevet i PHP som er ment til å håndtere administrasjonen av en MySQL. Du kan bruke phpMyAdmin til å utføre de fleste administrasjonsoppgaver, inkludert å opprette en database, utføre spørrslinger gjennom SQL-kommandoer og legge til brukerkontoer.



4.4 Frontend

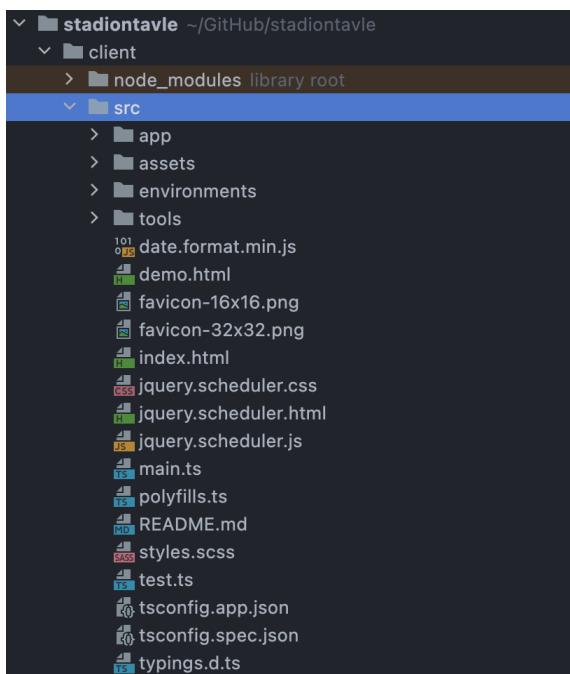
4.4.1 Filstruktur

Mappestruktur



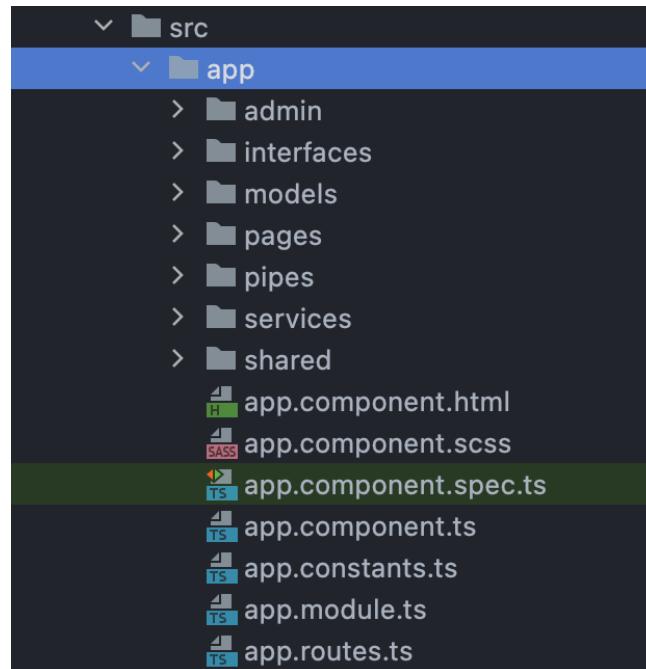
Figur 4.4.1.1: client-mappen

Rot-mappen er den mappen som ligger øverst av et filsystem eller et hierarki. I dette prosjektet er det “stadiowntavle” som er rot-mappen. I den ligger det to mapper, “client” og “server”, og en “README.me” fil. I denne frontend-delen ligger hovedfokuset på “client”-mappen.



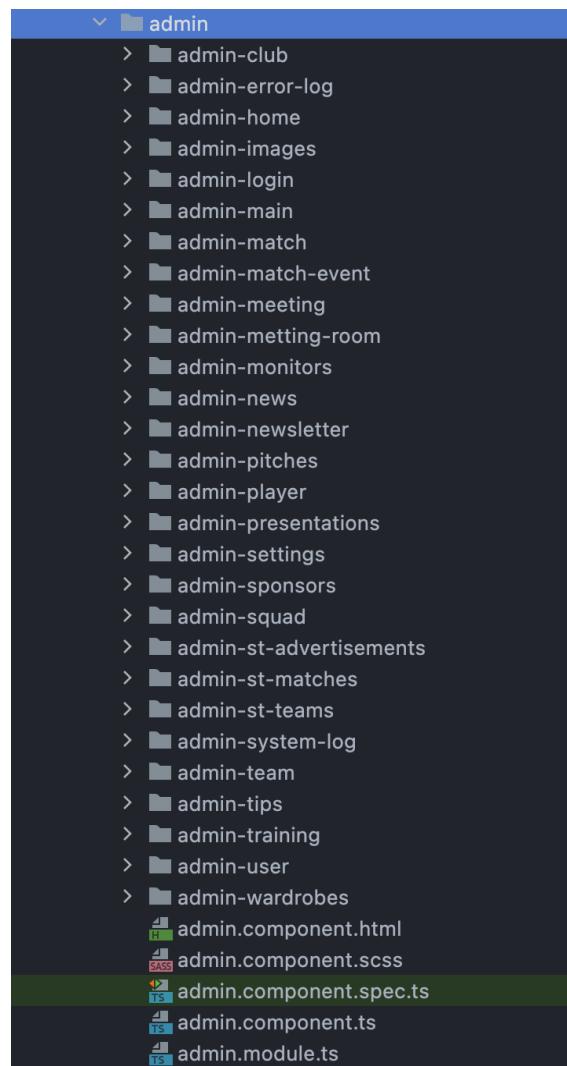
Figur 4.4.1.2: src-mappen

“src”-mappen i “client” kan ses på som “ryggraden” til prosjektet vårt, og det er stort sett der all koden ligger for vår del. Denne mappen inneholder fire undermapper: “app”, “assets”, “environments” og “tools”. Med en slik struktur som “src”-mappen har blitt delt inn i, får vi en bedre helhetlig oversikt av prosjektet og det gjør det lettere for de som skal videreutvikle prosjektet, å navigere seg til riktig kode.



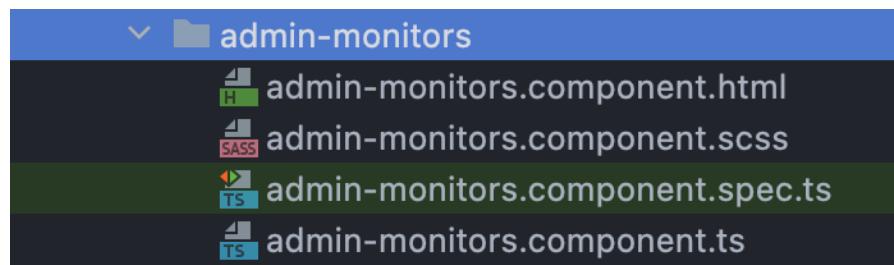
Figur 4.4.1.3: app-mappen

Den første undermappen kalt “app” inneholder enda flere undermapper. Her er den viktigste mappen for vår oppgave - “admin”-mappen. Denne skal bli nevnt senere i dette delkapitlet. Denne mappen inneholder også noen grunnleggende Angular-filer, som “app.component.ts”, “app.constants.ts”, “app.module.ts”, og “app.routes.ts”. Disse filene er grunnmuren av hele prosjektet, og disse kodelinene er komponenter som blir brukt for å bygge opp hele Angular-prosjektet.



Figur 4.4.1.4: admin-mappen

Den viktigste mappen for oss i sammenheng med oppgaven vår, er “admin”-mappen. Denne mappen inneholder koder for administrasjonssiden av Stadiontavle, og her ligger det undermapper for blant annet “skjermer”, “baner”, og “presentasjoner”. Disse mappene er for de forskjellige sidene på administrasjonssiden, og mer om det nevner vi i [Kapittel 6: Brukerveiledning](#).



Figur 4.4.1.5: admin-monitors-mappen

Våre endringer kommer hovedsakelig inn i “admin-monitors”-mappen, fordi vår oppgave har hovedfokus på “Skjermer”-siden (*Figur 4.4.2.6*). Grunnen for hvorfor vi endte opp med å ha løsningen vår i “Skjermer”-delen, står i [2.2.2 Analyse og funksjonalitet/design](#). Forklaring om hver kode hver for seg kommer nedenfor.

Skjerm	URL
1	https://stadiontavle.no/SAN/12

Navn	Mal
Eksempl 1	test

Figur 4.4.1.6: Skjermer-siden med Visningsmodus-funksjonen under

4.4.2 Implementasjon

admin-monitors.component.html

```

117  <ng-template #insertModal>
118  <div class="modal-header">
119    <h2 class="modal_header"> Sett inn mal </h2>
120
121    <button (click)="modalRef.hide()" aria-label="Close" class="close pull-right" type="button">
122      <span aria-hidden="true">&amptimes</span>
123    </button>
124  </div>
125  <div class="modal-body">
126    <form (ngSubmit)="submit()" [formGroup]="contactForm">
127      <div class="row p-5">
128        <div class="col-md-12">
129          <p style="... ">
130            <label for="<input ngModel> value_name" class="form-control" formControlName="value_name">
131            <!-- <button (click)="addCountry()">Add</button> -->
132          </p>
133          <p style="... ">
134            <label for="<input ngModel> value_type" class="form-control" formControlName="value_type">
135              <select class="form-control" formControlName="value_type">
136                <option [ngValue]="" disabled>Velg aktivitetstype</option>
137                <option *ngFor="let country of countries" [ngValue]="">{{country.name}}</option>
138              </select>
139
140              <!-- <button type="submit">Submit</button> -->
141            </p>
142            <p style="... ">
143              <label for="<input ngModel> value_temp" class="form-control" formControlName="value_temp">
144                <select class="form-control" formControlName="value_temp">
145                  <option [ngValue]="" disabled>Vis type mal</option>
146                  <option *ngFor="let temp of temps" [ngValue]="">{{temp.name}}</option>
147                </select>
148
149              <p style="... ">
150                <label for="<input ngModel> value_date" class="form-control" formControlName="value_date" type="number">
151                </p>
152              </div>
153            </div>
154          </form>
155        </div>
156      </div>
157    </div>

```

Figur 4.4.2.1: Skjema som dukker opp for å lage ny visning

```

160  <div class="modal-footer">
161    <div class="col-md-12">
162
163      <h1 *ngIf="textShow"><a
164        href="http://localhost:4200/tools/{{temp}}?clubId={{name}}&day={{date}}&activitytypes={{type}}"
165        target="_blank">https://stadiontavle.no/{{shortName}}/{{item.id}}</a></h1>
166      <button (click)="showText()" class="btn btn-primary" type="button">Vis link</button>
167
168    </div>
169  </div>
170

```

Figur 4.4.2.2: Skjema som dukker opp for å lage ny visning

Her har vi lagt til visningen som du kan se på Figur 6.4.2 i [6.4 Implementering av nye visninger](#). Den første figuren viser selve skjemaet og alle inputfeltene, mens den andre figuren viser knappen man kan klikke for å få opp en link hvor man kan se på malen man har satt inn.



admin-monitors.component.ts

```

16   export class AdminMonitorsComponent implements OnInit {
17
18     contactForm: FormGroup;
19
20
21     selectedLevel;
22     country_name = "";
23     set_country = "";
24
25     countries = [
26       {id: 1, name: "Kamp"}, 
27       {id: 2, name: "Trening"}, 
28       {id: 3, name: "Møte"} 
29     ];
30
31     temps = [
32       {id: 1, name: "Tabell", linksurl: "banedisp.html"}, 
33       {id: 2, name: "Liste", linksurl: "temp.html"}, 
34       {id: 3, name: "Kalender", linksurl: "temps.html"} 
35     ];
36

```

Figur 4.4.2.3: Inputverdier

man kan velge i feltene i skjemaet som dukker opp for å kunne lage en visning

Her kan man se hvordan vi har lagt inn verdier til inputfeltene, og hvordan vi har linket opp inputverdiene til mal, med html-filene som vi har for de ulike visningene.

```

67   selected() {
68     this.valueLink = this.selectedValue.id;
69     this.textShow = true;
70     // alert(e.target.value)
71   }
72
73   showText() {
74
75     this.valueLink = this.name + this.template + this.type + this.date;
76
77     this.textShow = true;
78   }
79
80   hideText() {
81     this.textShow = false;
82   }
83
84   ngOnInit() {
85     this.getMonitorList();
86     this.contactForm = this.fb.group(controlsConfig: {
87       value_type: [null],
88       value_name: [null],
89       value_template: [null],
90       value_date: [null],
91     });
92
93     this.setType();
94     this.setName();
95     this.setTemplate();
96     this.setDate();
97

```

Figur 4.4.2.4: Kode som viser showText()-metoden som skal få opp url-linken når man trykker på knappen som vist på figur 4.4.2.2.

```

119   submit() {
120     console.log("Form Submitted")
121     console.log(this.contactForm.value)
122   }
123
124   setType() {
125     this.contactForm.get("value_type").patchValue( value: null);
126   }
127
128   setName() {
129     this.contactForm.get("value_name").patchValue( value: null);
130   }
131
132   setTemp1() {
133     this.contactForm.get("value_temp1").patchValue( value: null);
134   }
135
136   setDate() {
137     this.contactForm.get("value_date").patchValue( value: null);
138   }
139

```

```

99   this.contactForm.get("value_type").valueChanges
100    .subscribe( next: f => {
101      this.onTypeChanged(f);
102    })
103
104   this.contactForm.get("value_name").valueChanges
105    .subscribe( next: f => {
106      this.onNameChanged(f);
107    })
108
109   this.contactForm.get("value_temp1").valueChanges
110    .subscribe( next: f => {
111      this.onTemp1Changed(f);
112    })
113
114   this.contactForm.get("value_date").valueChanges
115    .subscribe( next: f => {
116      this.onDateChanged(f);
117    })
118

```

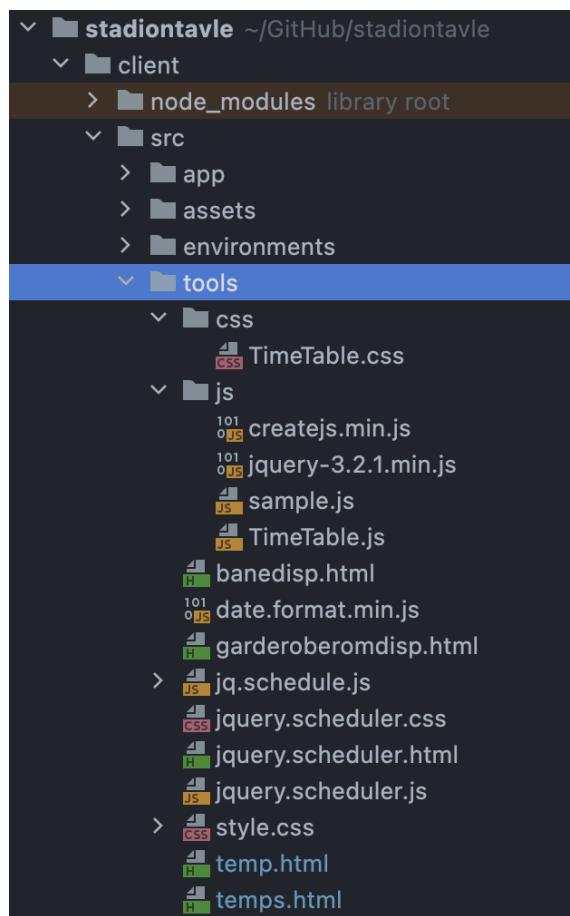
```

140   onNameChanged(value) {
141     console.log('onCountryChanged')
142     console.log(value)
143     this.name = value;
144   }
145
146   onTypeChanged(value) {
147     console.log('onCountryChanged')
148     console.log(value)
149     this.type = value;
150   }
151
152   onTemp1Changed(value) {
153     console.log('onCountryChanged')
154     console.log(value)
155     this.temp1 = value;
156   }
157
158   onDateChanged(value) {
159     console.log('onCountryChanged')
160     console.log(value)
161     this.date = value;
162   }

```

Figur 4.4.2.5, 4.4.2.6, og 4.4.2.7: Kode som viser get- og set-metoder for å få det overnevnte skjemaet til å fungere optimalt. Disse kodene er skrevet for å kunne velge de forskjellige inputverdiene i feltene. Det er også en submit()-metode som henter inn all data.

tools-mappen



Figur 4.4.2.8: tools-mappen

Tools-mappen inneholder html-kode for den eksisterende malen til Digimaker for Stadiontavle. Disse filene er kalt ”banedisp.html” og ”garderoberomdisp.html”. Malens design kan dere se på *Figur 2.2.2.2 i 2.2.2 Analyse og funksjonalitet/design*. I mappen har vi også ”temp.html” og ”temps.html” som er maler som vi har jobbet med. De andre filene blir også nevnt under.

- [date.format.min.js](#), [jquery.scheduler.html](#), [jquery.scheduler.js](#), [jquery.scheduler.css](#)

Disse filene er hentet fra <https://github.com/phstc/jquery-dateFormat>, og denne repositoryen er en jQuery Plugin, som er skrevet av GitHub-brukeren phstc (Pablo Cantero). Pluginen brukes til å formatera datoutdata ved hjelp av JavaScript. Fordelen med å bruke disse filene er at det er det minste datoformatbiblioteket som er tilgjengelig i dag¹⁰.

Nedenfor kan dere se koden til den første malen vi jobbet med: "temp.html". Denne malen er i listeform, som dere kan se på *Figur 6.4.11* i [6.4 Implementering av nye visninger](#). "temps.html" som er malen i kalenderform bruker jQuery-plugin filene nevnt i forrige avsnitt.

temp.html

```

1  <html>
2
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <style>
7          .parentDiv {
8              padding: 44px 100px !important;
9          }
10
11         #weekDay {
12             text-decoration: underline;
13         }
14
15         li {
16             margin-top: 10px;
17             font-size: 20px;
18         }
19
20     </style>
21
22     <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
23     </script>
24
25     <div class="parentDiv">
26         <div id="ddd"></div>
27         <h1 id="weekDay" style="...></h1>
28
29         <div id="tree">
30
31     </div>

```

Figur 4.4.2.9: temp.html

¹⁰ (<https://github.com/phstc/jquery-dateFormat>, 2019)

```

51   <h1 id="msg" style="...">></h1>
52
53   </html>
54   <script type="text/javascript" language="javascript">
55     function getParamValue(paramName) {
56       var url = window.location.search.substring(1); //get rid of "?" in querystring
57       var qArray = url.split( separator: '&' ); //get key-value pairs
58       for (var i = 0; i < qArray.length; i++) {
59         var pArr = qArray[i].split( separator: '=' ); //split key and value
60         if (pArr[0] == paramName)
61           return pArr[1]; //return value
62     }
63   }
64   $(document).ready( handler: function() {
65     var clubId = getParamValue( paramName: 'clubId' );
66     var day = getParamValue( paramName: 'day' );
67     var activitytypes = getParamValue( paramName: 'activitytypes' );
68     var rowData = getParamValue( paramName: 'cc' ) ? getParamValue( paramName: 'cc' ) : 4
69
70
71     var weekday = ["Søndag", "Mandag", "Tirsdag", "Onsdag", "Torsdag", "Fredag", "Lørdag"];
72     const monthNames = ["januar", "februar", "mars", "april", "mai", "juni",
73      "juli", "august", "september", "oktober", "november", "desember"
74    ];
75
76     var myCurrentDate = new Date();
77     var myPastDate = new Date(myCurrentDate);
78     if (day > 0) {
79       myPastDate.setDate(myPastDate.getDate() + Math.abs(day));
80     } else {
81       myPastDate.setDate(myPastDate.getDate() - Math.abs(day));
82     }
83
84     var custom = weekday[myPastDate.getDay()] + ' ' + String(myPastDate.getDate()).padStart(2, '0') + '.' + monthNames[myPastDate.getMonth()]
85     $( selector: "#weekDay" ).append(custom);
86
87

```

Figur 4.4.2.10 og 4.4.2.11: temp.html - Denne malen er inspirert av banedisp.html som er den malen som Digimaker hadde opprinnelig på Stadiontavle.

```

88   $.ajax( settings: {
89     url: "http://127.0.0.1:8000/api/stadionAdmin/GetMatixActivityData",
90     //url: "https://stadiontavle.no/api/api/stadionAdmin/GetMatixActivityData",
91     type: "POST",
92     data: JSON.stringify( value: {
93       ref_club_id: clubId,
94       day: day,
95       activitytypes: activitytypes,
96       row: rowData ? rowData : 4
97     }),
98     contentType: "application/json",
99     success: function(response) {
100       if (response.data.timeData[0].StartFrom === null) {
101         $( selector: "#msg" ).append( content1: "Ingen aktivitet" );
102       } else {
103         var thData = response.data.timeData[0];
104         var activityData = response.data.PitchData;
105         var tbody = $( selector: '#activityData tbody' );
106         var tr = $( selector: '<tr>' );
107         var tr1 = $( selector: '<tr>' );
108         var timeGap = [];
109         $(document).ready( handler: function() {
110

```



På Figur 4.4.2.10 og 4.4.2.11 kan man se hvordan vi definerer ukedager og måneder. Det blir også laget parametre for clubID, dag og aktivitetstype. Man kan også se en \$.ajax()-metode. Denne blir brukt for å kunne oppdatere visningen (malen) med data som er lagt inn.

```

112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
      var treeParentArr = activityData;
      if (activityData.length) {

          var $ulLevel1 = $( selector: "<ul />").appendTo($( selector: "#tree"));
          //level 1
          for (var i = 0; i < activityData.length; i++) {
              var treeParentVal = activityData[i];
              var listChildLevel1 = $ulLevel1.append($( html: "<li />", attributes: {
                  text: treeParentVal.displayName
              }));
              var childSubVal1 = treeParentVal.activity;
              if (childSubVal1) {
                  var treechildLevel1 = childSubVal1;

                  //level 2
                  //check the condition data is there or not
                  if (treechildLevel1) {
                      console.log(i);
                      var $ulLevel2 = $( selector: "<ul />").appendTo($(listChildLevel1));
                      for (var j = 0; j < treeParentVal.activity.length; j++) {
                          var treeChildLevelData1 = treechildLevel1[j];
                          var d = treeParentVal.activity[j].FromTime;
                          d = d.split(' ')[1];
                          var e = treeParentVal.activity[j].ToTime;
                          e = e.split(' ')[1];
                          var f = treeParentVal.activity[j].Date;
                          f = f.split(' ')[0];

                          var listChildLevel2 = $ulLevel2.append($( html: "<li />", attributes: {
                              text: treeParentVal.activity[j].Title
                          })).css( propertyName: 'color', treeParentVal.activity[j].farge);
                          var listChildLevel2 = $ulLevel2.append($( html: "<li />", attributes: {
                              text: d + ' TO ' + e
                          })).css( propertyName: 'color', treeParentVal.activity[j].farge);
                          var listChildLevel2 = $ulLevel2.append($( html: "<li />", attributes: {
                              text: f
                          })).css( propertyName: 'color', treeParentVal.activity[j].farge);
                      }
                  }
              }
          }
      }
  
```

Figur 4.4.2.12: temp.html - Kode for å få dato og tid opp på visningen

```

168
169
170
171
172
173
174
175
176
177
178
179
180
181
    var jData = 0;
    var sta = 0;
    var dynamicRow = parseInt(rowData) / 2;
    for (var i = 0; i < timeGap.length + 1; i++) {
        if (i > 0) {
            for (var jj = 0; jj <= 1; jj++) {
                $( selector: '<td colspan=' + dynamicRow + '' style="width: 6.45624%; height: 18px; ' +
                'border: 1px solid black; text-align: center;">').html( htmlString: jj == 0 ? timeGap[i - 1] + ".00" : timeGap[i - 1] + ".30").attr(
                attributeName: "id", value: timeGap[i] + "-" + i).appendTo(tr1);
            }
        } else if (i == 0) {
            $( selector: '<td>').html( htmlString: "").attr( attributeName: "id", value: timeGap[i] + "-" + i).appendTo(tr1);
        }
        tbody.append(tr1);
    }
  
```

Figur 4.4.2.13: temp.html - Kode for listevisning



```

184     $.each(activityData, callback: function(i, item) {
185         if (item.activity.length <= 0) {
186
187
188             var tr = $( selector: '<tr style="height: 30px;">');
189             for (var i = 1; i <= thData.columnCount; i++) {
190                 if (i === 1) {
191                     $( selector: '<td>').html( htmlString: "" ).attr( attributeName: "id", i).appendTo(tr);
192                     $( selector: '<td style="width: 9.46915%;height: 18px; border: 1px solid black;vertical-align: top;">').html( htmlString: "<span style=" );
193                     tbody.append(tr);
194                 } else {
195                     $( selector: '<td>').html( htmlString: "" ).attr( attributeName: "id", i).appendTo(tr);
196                     tbody.append(tr);
197                 }
198             }
199         } else {
200             var tr = $( selector: '<tr style="height: 30px;">');
201             for (var m = 0; m < item.count; m++) {
202                 if (m === 0) {
203                     var count = thData.columnCount;
204                     for (var j = 1; j <= count; j++) {
205                         if (j === 1) {
206                             $( selector: '<td rowspan=' + item.count + ' style="width: 9.46915%;height: 18px; border: 1px solid black;vertical-align: middle; background-color: #e0e0e0;">').appendTo(tr);
207                             tbody.append(tr);
208                         } else {
209                             if (j === item.activity[m].startColumn) {
210
211                                 $( selector: '<td colspan=' + item.activity[m].colspan + ' style="text-align: center; vertical-align: middle; background-color: #e0e0e0;">').appendTo(tr);
212                                 count = thData.columnCount - item.activity[m].colspan + 1;
213                             } else {
214                                 $( selector: '<td style="width: 1%;">').html( htmlString: "" ).attr( attributeName: "id", value: "data.Title").appendTo(tr);
215                             }
216                         }
217                     }
218                 }
219             }
220         }
221     }

```

Figur 4.4.2.14: temp.html - Kode for listevisning

```

221     }
222
223     } else {
224         var tr1 = $( selector: '<tr style="height: 30px;">');
225         var count = thData.columnCount;
226         for (var k = 1; k <= count; k++) {
227             if (k !== 1) {
228                 var $ulLevel2 = $( selector: "<ul />").appendTo($(listChildLevel1));
229
230                 var treeChildLevelData1 = treechildLevel1[j];
231                 var listChildLevel2 = $ulLevel2.append($( html: "<li />", attributes: {
232                     text: item.activity[m].Title
233                 }));
234                 if (k === item.activity[m].startColumn) {
235                     $( selector: '<td colspan=' + item.activity[m].colspan +
236                         ' style="text-align: center; vertical-align: middle; background-color: ' +
237                         item.activity[m].farve + 'color: ' + item.activity[m].text_color +
238                         '">').html(item.activity[m].Title).attr( attributeName: "id",
239                         value: "data.Title").appendTo(tr1);
240                     tbody.append(tr1);
241                     count = thData.columnCount - item.activity[m].colspan + 1;
242                 } else {
243                     $( selector: '<td style="width: 1%;">').html( htmlString: "" ).attr( attributeName: "id",
244                         value: "data.Title").appendTo(tr1);
245                     tbody.append(tr1);
246                 }
247             }
248         }
249     });
250 };
251 });
252 });
253 
```

Figur 4.4.2.15: temp.html - Kode for listevisning



temps.html

```

67   <body>
68
69
70     <div id="scheduler"></div>
71     <h1 id="msg" style="..."></h1>
72
73
74   <div id="footer">
75
76     </div>
77
78     <!-- Latest compiled and minified JavaScript -->
79     <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
80     </script>
81     <script type="text/javascript" src="date.format.min.js">
82     </script>
83     <link rel="stylesheet" href="jquery.scheduler.css">
84     <script src="jquery.scheduler.js"></script>
85

```

Figur 4.4.2.16: temps.html - Kode for ”kalender”visning, denne malen er inspirert av både
banedisp.html og temp.html

På Figur 4.4.2.16 kan man se at vi importerer filer fra den nevnte GitHub-repoen for
jQuery-pluginen.

```

88
89   <script type="text/javascript">
90     <!-- Function to get parameter value from URL -->
91     function getParamValue(paramName) {
92       var url = window.location.search.substring(1); //get rid of "?" in querystring
93       var qArray = url.split( separator: '&' ); //get key-value pairs
94       for (var i = 0; i < qArray.length; i++) {
95         var pArr = qArray[i].split( separator: '=' ); //split key and value
96         if (pArr[0] == paramName)
97           return pArr[1]; //return value
98       }
99     }
100
101    $(document).ready( handler: function() {
102      var clubId = getParamValue( paramName: 'clubId' );
103      var day = getParamValue( paramName: 'day' );
104      var activitytypes = getParamValue( paramName: 'activitytypes' );
105      var rowData = getParamValue( paramName: 'cc' ) ? getParamValue( paramName: 'cc' ) : 4
106      $( selector: '.hour-header hour').text( text: 'The replaced text.' );
107      $( selector: '#hour-header hour').text( text: 'The replaced text.' );
108
109
110      var weekday = ["Søndag", "Mandag", "Tirsdag", "Onsdag", "Torsdag", "Fredag", "Lørdag"];
111      const monthNames = ["januar", "februar", "mars", "april", "mai", "juni",
112                          "juli", "august", "september", "oktober", "november", "desember"
113      ];

```

Figur 4.4.2.17: temps.html - vi definerer ukedager og måneder. Det blir også laget parametre
for clubID, dag og aktivitetstype.

```

114     var myCurrentDate = new Date();
115     var myPastDate = new Date(myCurrentDate);
116     if (day > 0) {
117         myPastDate.setDate(myPastDate.getDate() + Math.abs(day));
118     } else {
119         myPastDate.setDate(myPastDate.getDate() - Math.abs(day));
120     }
121
122     var custom = weekday[myPastDate.getDay()] + ' ' + String(myPastDate.getDate()).padStart(2, '0') + '.' +
123         + monthNames[myPastDate.getMonth()] + ' ' + myPastDate.getFullYear();
124     $( selector: "#weekDay" ).append(custom);
125

```

Figur 4.4.2.18: temps.html - Her bruker vi get- og set-metoder for å tak i dag og tid.

```

130     $.ajax( settings: {
131         url: "http://127.0.0.1:8000/api/stadionAdmin/GetMatixActivityData",
132         //url: "https://stadiontavle.no/api/api/stadionAdmin/GetMatixActivityData",
133         type: "POST",
134         data: JSON.stringify( value: {
135             ref_club_id: clubId,
136             day: day,
137             activitytypes: activitytypes,
138             row: rowData ? rowData : 4
139         }),
140         contentType: "application/json",
141         success: function(response) {
142             if (response.data.timeData[0].StartFrom === null) {
143
144
145
146                 var thData = response.data.timeData[0];
147                 var activityData = response.data.PitchData;
148                 // var tbody = $('#activityData tbody');
149                 // var tr = $('<tr>');
150                 // var tr1 = $('<tr>');
151                 // var timeGap = [];
152                 $( selector: "#msg" ).append( content: "Ingen aktivitet");
153
154
155
156             } else {
157                 var thData = response.data.timeData[0];
158                 var activityData = response.data.PitchData;
159

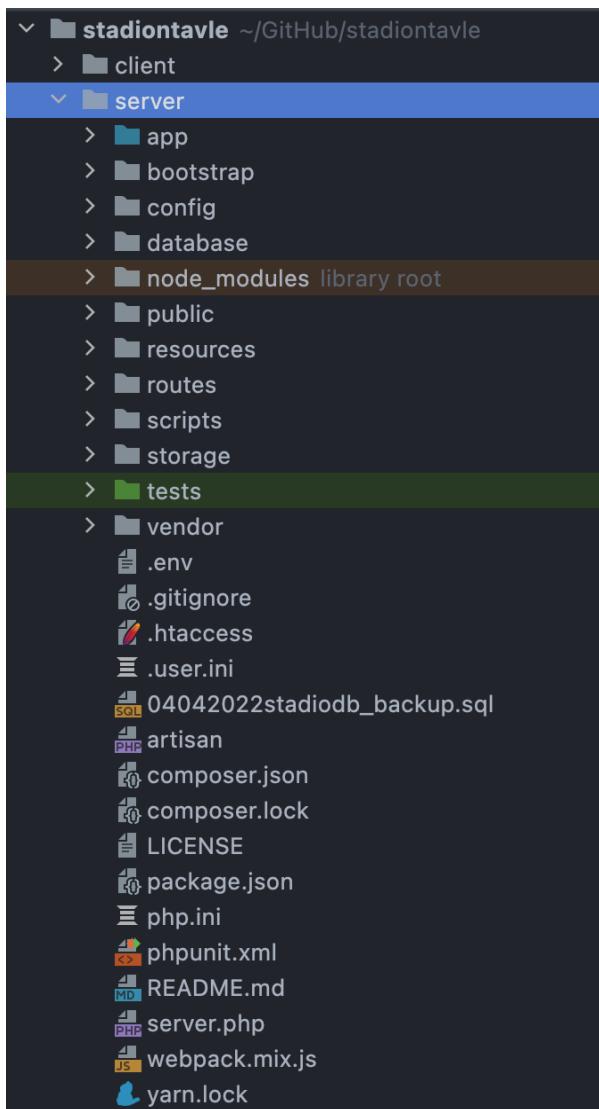
```

Figur 4.4.2.16: temps.html - en \$.ajax()-metode, for å kunne oppdatere visningen (malen)
med data som er lagt inn

4.5 Backend

4.5.1 Filstruktur og database

Mappestruktur

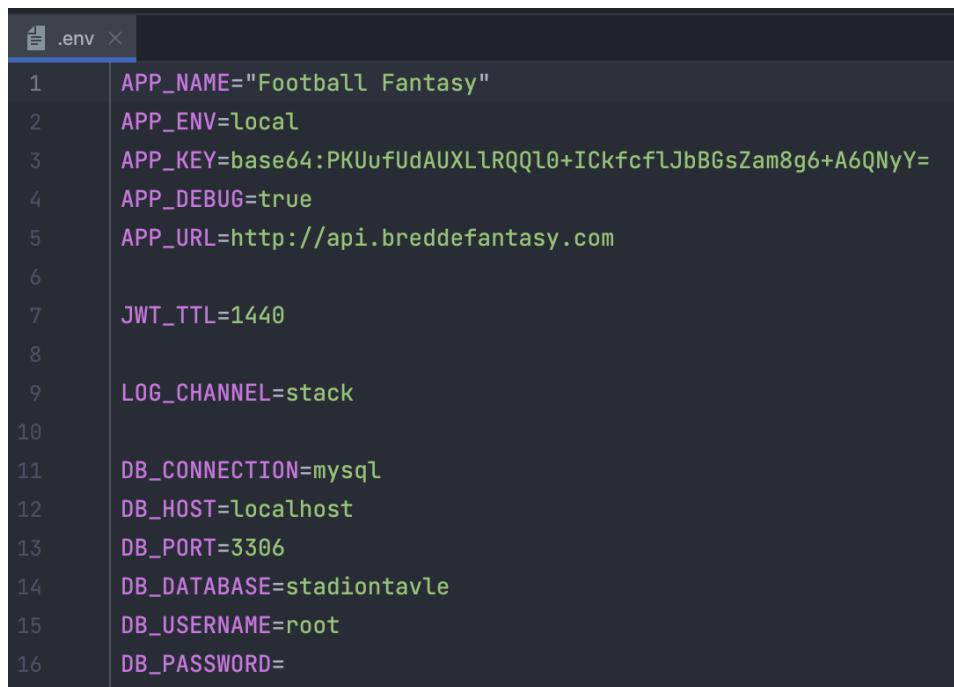


```
stadiontavle ~/GitHub/stadiontavle
  client
  server
    app
    bootstrap
    config
    database
    node_modules library root
    public
    resources
    routes
    scripts
    storage
  tests
  vendor
    .env
    .gitignore
    .htaccess
    .user.ini
    04042022stadiodb_backup.sql
    artisan
    composer.json
    composer.lock
    LICENSE
    package.json
    php.ini
    phpunit.xml
    README.md
    server.php
    webpack.mix.js
  yarn.lock
```

Figur 4.5.1.1: server-mappen

Siden vi endte opp med å ikke jobbe stort på backend delen, kommer vi ikke til å nevne mye om det. Det er om database vi skal gå mest innpå her. Allikevel vil vi vise server-mappen og hvordan strukturen er satt opp i den, for at vi skal kunne kjøre Stadiontavle. Server-mappen består av flere undermapper, hvor "app", "database", og "routes" er blant de viktigste mappene som inneholder mesteparten av tjenestens funksjoner. Mens man jobber med et team og i forskjellige miljøer, er det forskjellige innstillinger for hver maskin. Til det har vi

.env-filen. I Laravel-hovedmappen bør en ha en slik fil som .env som inneholder forskjellige innstillingar. For å tilkoble seg til en database, må man endre data i denne filen. Dette gjelder de siste linjene på *Figur 4.5.1.2*, "DB_...". Andre filer som er viktige for oss i dette prosjektet er sql-filen med data for Stadiontavle.



```

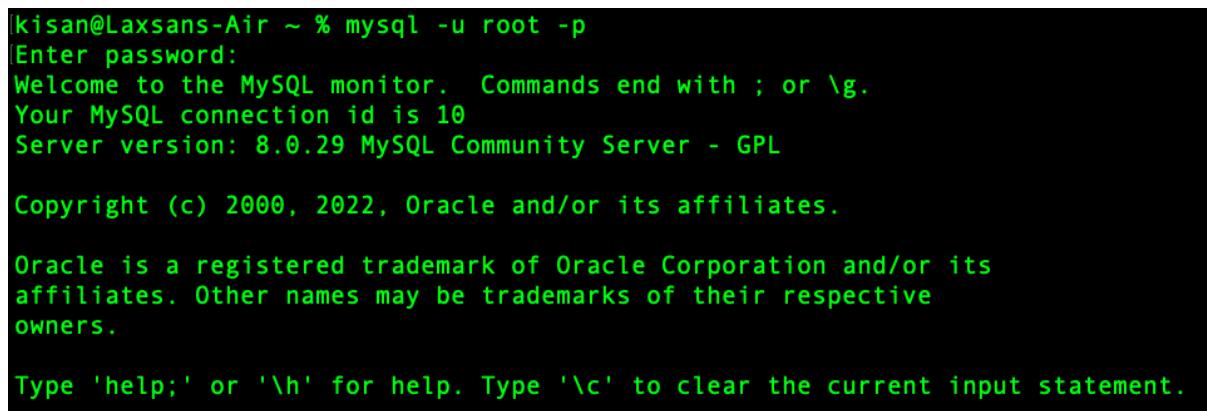
1 APP_NAME="Football Fantasy"
2 APP_ENV=local
3 APP_KEY=base64:PKUufUdAUXLlRQQl0+ICKfcfLJbBGsZam8g6+A6QNyY=
4 APP_DEBUG=true
5 APP_URL=http://api.breddefantasy.com
6
7 JWT_TTL=1440
8
9 LOG_CHANNEL=stack
10
11 DB_CONNECTION=mysql
12 DB_HOST=localhost
13 DB_PORT=3306
14 DB_DATABASE=stadiontavle
15 DB_USERNAME=root
16 DB_PASSWORD=

```

Figur 4.5.1.2: .env-filen

Oppsett av database

Start databaseserveren i terminal og lag en ny database. Importer **.sql-filen** fra servermappen. Deretter kan du se SQL-dataene i terminal. Under følger figurer for å vise hele prosessen.



```

kisan@Laxsans-Air ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.29 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```

Figur 4.5.1.3: Bruk kommandoen mysql -u root -p for å komme deg inn på mysql som en root.

```

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.01 sec)

mysql> CREATE DATABASE stadiontavle;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| stadiontavle    |
| sys            |
+-----+
5 rows in set (0.00 sec)

```

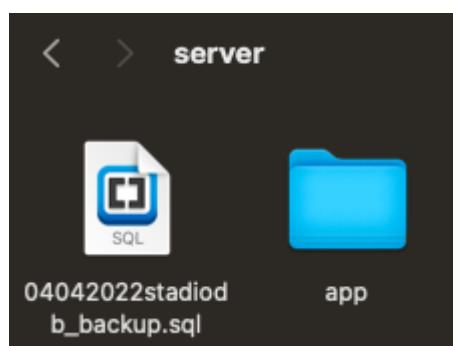
*Figur 4.5.1.4: Bruk kommandoen SHOW DATABASES; for å se databasene som finnes på din MySQL server. Skriv deretter CREATE DATABASE *stadiontavle*; for å lage en ny database, med navn, Stadiontavle. Hvis du bruker kommandoen SHOW DATABASES; nå, vil du se "stadiontavle" som en av databasene.*

```

mysql> USE stadiontavle;
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)

```

*Figur 4.5.1.5: Bruk kommandoen USE *stadiontavle*; og nå kan du lage tabeller i den valgte databasen, "stadiontavle". Med kommandoen SHOW TABLES; kan du se at databasen vår er tom, siden det ikke er lagt til noen verdier i den enda.*



Figur 4.5.1.6: Finn sql-filen som ligger i server-mappen.

```
mysql> source /Users/kisan/Documents/GitHub/stadiontavle/server/04042022stadiodb_backup.sql
```

Figur 4.5.1.7: Bruk kommandoen source + filbanen til sqlfilen.

```
mysql> SHOW TABLES;
+-----+
| Tables_in_stadiontavle |
+-----+
| articles
| chips
| clubs
| invite_participants
| krets
| league_points_matrix
| leagues
```

Figur 4.5.1.8: Med kommandoen SHOW TABLES; kan man nå se at vi har importert tabellene fra sql-filen inn til databasen vår.

```
mysql> SELECT * FROM st_pitches;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name | displayName | subParentId | created_at | updated_at
| parent_id | parent_name | fiks_id | status | ref_club_id | typeCheck |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Funnefoss gress | Funnefoss gress | 2504 | 2020-12-21 09:19:09 | 2020-12-21 09:19:09
| 12053 | Funnefoss stadion | NULL | 0 | 10 | 0 |
| 2 | Funnefoss kunstgress | Funnefoss kunstgress | 2510 | 2020-12-21 09:19:09 | 2020-12-21 09:19:09
| 12053 | Funnefoss stadion | NULL | 0 | 10 | 0 |
```

Figur 4.5.1.9: Ved hjelp av kommandoen SELECT * FROM "tabellnavn", kan man se innholdet av tabellene.

Kapittel 5: Testdokumentasjon

5.1 Innledning

Testing av løsningen på webapplikasjonen til Digimaker ble utført for å sikre at løsningen fungerte som den skulle. Dette var en viktig del av prosjektet ettersom utviklingsteamet skulle kunne kvalitetssikre den ferdige løsningen, avdekke feil og sikre at løsningen ble utviklet etter oppdragsgiverens kravspesifikasjoner. Testing skal kunne redusere risikoen for dårlig kvalitet av et system, og det kunne skape tiltro til kvalitetsnivaet ved hjelp av testobjektene¹¹. Testobjektene som ble tatt i bruk er enhetstest, systemtest og akseptansestest. Da integrasjonstest blant annet går ut på å teste grensesnittet opp mot et API uteble denne testen, da vi ikke jobbet opp mot et API på vår løsning hos Digimaker.

I løsningen på administrasjonssiden skal en ha tilgang til å sette opp nye maler for hvordan diverse aktiviteter skal vises på stadiontavlen. Dette innebærer blant annet å opprette nye visninger av enten kamper, treninger eller møter ved de forskjellige arenaene, der klubb-ID skulle komme opp som klubbnavn, og ikke som ID-nummer. Det var disse funksjonalitetene teamet ønsket å teste når løsningen skulle valideres.

5.2 Testplan

Testplanen ble utviklet ut fra hvilke funksjoner som var interessante å teste. Teamet ville teste fortløpende etter hver funksjonalitet som ble lagt til den eksisterende løsningen. Dette er kjent som testdrevet utvikling, noe som betyr at det blir produsert kode og testet parallelt under utviklingen¹². Hver gang vi jobbet med en funksjon skulle det utføres en enhetstest, der utviklingen ikke går videre før hver enkelt test er godkjent med en fungerende kode.

¹¹ (Krattebøl, 2020a)

¹² (Sommerville, 2011)

5.3 Enhetstesting

Dette er den første fasen av testing av programvare, og i denne fasen skulle det utvikles testfunksjoner på alle metodene som utformer løsningen¹³.

Kriteriet her er å teste alle metoder i løsningen med ulike betingelser. Det skal deretter opprettes ulike tester. Dette skal gjøres for å få oversikt over at den valgte metoden er korrekt eller om den returnerer feil. Målet er å ha en [CodeCoverage](#) på 100%.

Verktøyet som ble benyttet for enhetstesting var terminal og Visual Studio Code, samt at det ble brukt CodeCoverage for å analysere dekningsgraden av testene. De røde testene indikerte feil i koden, mens de grønne testene indikerte at koden var korrekt.

5.3.1 Hvilke funksjoner ble testet

Funksjonen vi testet var å få fram listeform visningen på en ekstern side ved å klikke på linken vi får fram når vi fyller ut skjemaet for å lage en ny visning (*Figur 6.4.14*). Deretter bruker vi konsollvinduet på nettleseren Google Chrome fra Chrome DevTools, for feilsøking og testing av klientkode. Konsollen er veldig praktisk for å gi konkrete tilbakemeldinger som inkluderer hva slags verdier og variabler som blir kjørt og ikke. Vi brukte også Visual Studio Code for å kjøre CodeCoverage.

¹³ (Krattebøl, 2020a)

5.3.2 Resultat av enhetstest

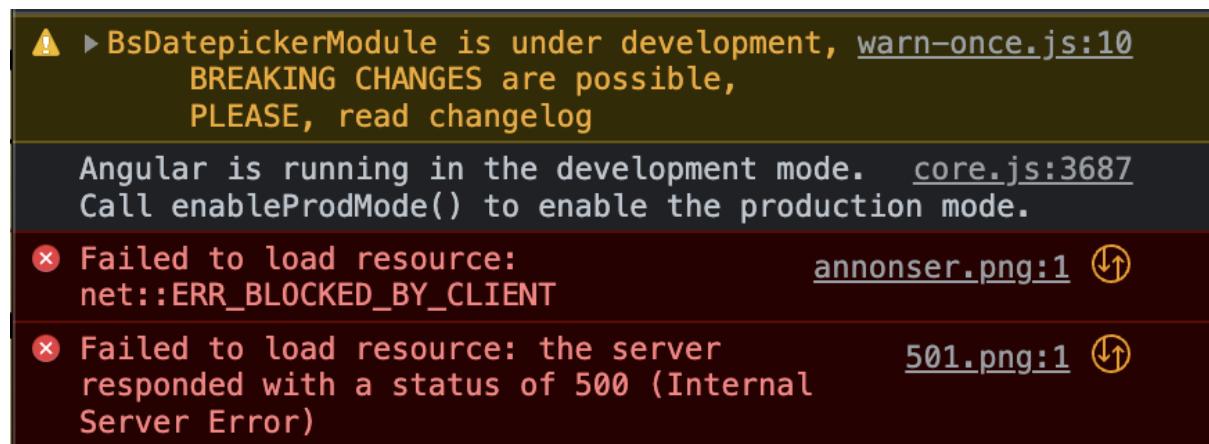
Tirsdag 24. mai 2022

- Bugården parkanlegg
- Kunstgress
 - test trenings
 - 20:00:00 TO 21:00:00
 - 2022-05-24
- Gress
- Mini

Figur 5.4.2.1: Listevisning - Ingen error i console

I vårt tilfelle er det ingen error som dukker opp på den eksterne siden vist på Figur 5.4.2.1.

Men når vi går tilbake til ”Skjermer”-siden (Figur 6.4.1) hvor vi lagde visningen fra, så dukker det noen error på konsollen, vist på Figur 5.4.2.2. Errormeldingen forteller oss:



Figur 5.4.2.2: Errormeldinger i console

Disse feilmeldingene kommer av andre koder som finnes i stadiontavle-mappen. Siden vi jobber med en eksisterende løsning, må vi regne med at det finnes eksterne feil som ikke kommer av vår kode. Det er grunnen i dette tilfellet.

```
{
  "client": {
    "defaults": [
      {
        "project": "src/tsconfig.spec.json",
        "exclude": "**/node_modules/**"
      },
      {
        "project": "e2e/tsconfig.e2e.json",
        "exclude": "**/node_modules/**"
      }
    ],
    "test": {
      "options": {
        "karmaConfig": "./karma.conf.js",
        "codeCoverage": true
      }
    }
  }
}
```

Figur 5.4.2.3: angular-cli.json - Vi legger til “codeCoverage”: true under “test” og “options”, for å kunne kjøre ng test i kommandovinduet på terminal.

```

PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE GITLENS zsh - client + ×
kisansrirangan@Kisans-MBP client % ng test
Path must be a string. Received undefined
TypeError: Path must be a string. Received undefined
  at assertPath (path.js:28:11)
  at Object.join (path.js:1246:7)
  at Promise (/Users/kisansrirangan/GitHub/stadiontavle/client/node_modules/@angular/cli/tasks/test.js:25:38)
  at new Promise (<anonymous>)
  at Class.run (/Users/kisansrirangan/GitHub/stadiontavle/client/node_modules/@angular/cli/tasks/test.js:23:16)
  at Class.run (/Users/kisansrirangan/GitHub/stadiontavle/client/node_modules/@angular/cli/commands/test.js:122:25)
  at resolve (/Users/kisansrirangan/GitHub/stadiontavle/client/node_modules/@angular/cli/ember-cli/lib/models/command.js:261:20)
  at new Promise (<anonymous>)
  at Class.validateAndRun (/Users/kisansrirangan/GitHub/stadiontavle/client/node_modules/@angular/cli/ember-cli/lib/models/command.js:240:12)
)
  at Promise.resolve.then.then (/Users/kisansrirangan/GitHub/stadiontavle/client/node_modules/@angular/cli/ember-cli/lib/cli.js:140:24)
  at <anonymous>
kisansrirangan@Kisans-MBP client %

```

Figur 5.4.2.4: Feilmeldinger fra når vi kjører ng test

For å få CodeCoverage-rapporter hver gang man tester, setter du følgende alternativ i CLI-konfigurasjonsfilen, angular.json¹⁴, som vist på Figur 5.4.2.3. Problemet for oss er at avhengighetene som vi laster ned når vi installerer prosjektet, er ikke oppdatert. For å kunne kjøre CodeCoverage og andre verktøy, må vi ha oppdatert versjonene til blant annet Node, Angular, TypeScript. Dette kan vi i gruppa ikke gjøre noe med, fordi Stadiontavle er en eksisterende løsning som Digimaker har jobbet med. Mange av metodene som er brukt i koden, er utdaterte. For å få prosjektet til å fungere optimalt som det gjør nå, må vi ha nøyaktig de versjonene som det står i README.me-filen for installasjon av prosjektet. Dermed får vi opp en del feilmeldinger når vi skal prøve å kjøre CodeCoverage, fordi det er forventet at man skal oppdatere til de nye versjonene av tjenestene og bibliotekene.

¹⁴ (Angular, 2022)

5.4 Systemtesting

Testing av systemet er den andre fasen av testingen, der systemet skal testes helhetlig.

Dette innebærer at brukerhistorier og kravspesifikasjoner blir knyttet opp mot de eksisterende funksjonene i systemet, og at disse blir utført korrekt¹⁵.

Testen blir godkjent hvis kravspesifikasjonen tilfredsstilles, og dersom resultatene ikke oppfyller kravspesifikasjonen skal dette dokumenteres og rapporteres.

5.4.1 Hvilke funksjoner ble testet

Liste over brukerhistorier som ble testet:

- Velge tavleeditor i hovedmenyen på adminsiden.
- Opprette ny tavle
- Kunne velge forskjellige layouts/maler
- Kunne navigere frem enkelt til ønsket funksjonalitet
- Kopiering av URL
- Kunne velge klubb ut fra navn på klubben og ikke ID nr
- Forhåndsvisning av tavle viser klubb, område og tid korrekt
- Tavlen blir lagret etter endring

5.4.2 Resultat av systemtest

Skjermer

Skjerm		URL	
1		https://stadiontavle.no/SAN/12	
« Forrige	1	Neste »	

Visningsmodus

Søk		Ny
Navn	Mal	
Eksempel 1	test	

Figur 5.4.2.1: Velge tavleeditor i hovedmenyen på adminsiden.

¹⁵ (Krattebøl, 2020c)

Sett inn mal ×

Klubb-ID

Aktivitetstype
 Velg aktivitetstype

Mal
 Vis type mal

Dag

Vis link

Figur 5.4.2.2: Opprette ny tavle.

Sett inn mal ×

Klubb-ID

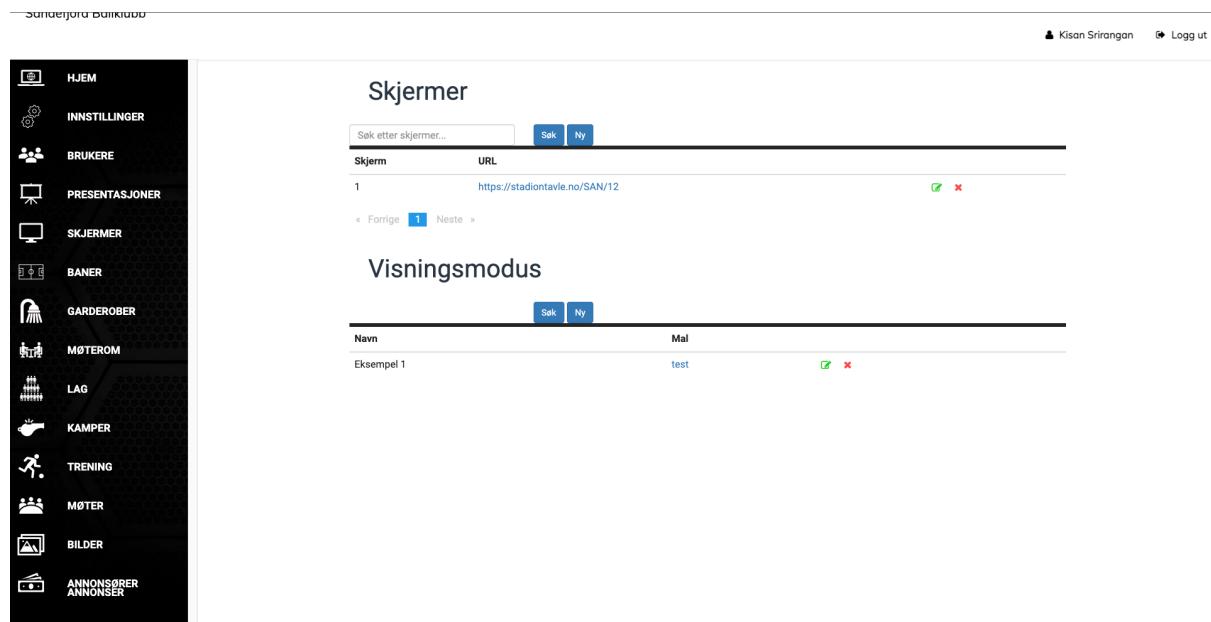
Aktivitetstype
 Velg aktivitetstype

Mal
 ✓ Vis type mal

Tabell
Liste
Kalender

Vis link

Figur 5.4.2.3: Kunne velge forskjellige layouts/maler.



Figur 5.4.2.4: Kunne navigere frem enkelt til ønsket funksjonalitet.

Sett inn mal

Klubb-ID
47

Aktivitetstype
Trening

Mal
Tabell

Dag
0

<https://stadiontavle.no/>

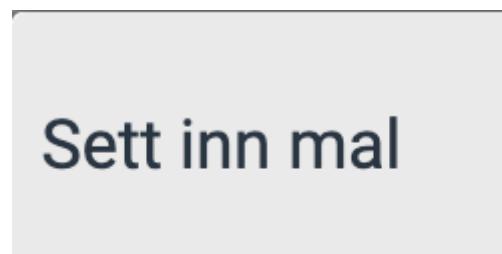
Vis link

← → C ⌂ http://localhost:4200/tools/banedisp.html?clubId=47&day=0&activitytypes=2

Onsdag 25. mai 2022

Bugården parkanlegg	18.00	
Kunstgress		test trening
Gress		
Mini		

Figur 5.4.2.5 og 5.4.2.6: Kopiering av URL.



Figur 5.4.2.7: Kunne velge klubb ut fra navn på klubben og ikke ID nr.

Onsdag 25. mai 2022

- Bugården parkanlegg
- Kunstgress
 - test trening
 - 18:00:00 TO 19:00:00
 - 2022-05-25
 - test 2
 - 18:00:00 TO 19:00:00
 - 2022-05-25
- Gress
- Mini

Trening

date	Kommende	Alle lag	Ny			
Tittel	Lag	Dato	Fra	Til	Pitch	
test trening	VIF	2022-05-25	18:00	19:00	Kunstgress	<input checked="" type="checkbox"/> <input type="checkbox"/>
test 2	LSK	2022-05-25	18:00	19:00	Kunstgress	<input checked="" type="checkbox"/> <input type="checkbox"/>

« Forrige 1 Neste »

Figur 5.4.2.8 og Figur 5.4.2.9: Forhåndsvisning av tavle viser klubb, område og tid korrekt.

Visningsmodus

		Søk	Ny
Navn	Mål		
Eksempel 1	test	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figur 5.4.2.10: Tavlen blir lagret etter endring.

5.5 Akseptansetest

Dette var den siste fasen av testingen, der oppdragsgiver skulle gi godkjenning av produktet vi har utviklet. Under en akseptansetest tester en brukerhistorier som beskriver betingelsene for hva en funksjon i løsningen skal gjøre¹⁶.

Brukerhistoriene tar her utgangspunkt i kravspesifikasjonen.

Digimaker fikk teste funksjonene i løsningen, og ga oss deretter tilbakemelding på hvilke funksjoner som ble ansett som bestått eller ikke bestått.

5.5.1 Hvilke funksjoner ble testet

- Brukerhistorier med utgangspunkt i kravspesifikasjonen.

5.5.2 Resultat av akseptansetest

Brukerhistorier	Implementert	Funksjonalitet	Design	Kommentar	Resultat
Velge tavleeditor i hovedmenyen på admin siden.	Ja	Bestått	Bestått	Vi har løst dette ved å lage en egen del "Visningsmodus", som skal gi muligheten til å ha forskjellige visninger. Denne delen kan	<input checked="" type="checkbox"/>

¹⁶ (Krattebøl, 2020d)

				man finne ved å velge "Skjermer" på hovedmenyen.	
Opprette ny tavle	Ja	Bestått	Bestått	Man kan opprette en ny tavle/visning, ved å bruke popugen "Sett inn mal" som man får opp fra "Visningsmodus".	<input checked="" type="checkbox"/>
Kunne velge forskjellige layouts/maler	Ja	Bestått	Bestått	Det har blitt gitt forskjellige valgmuligheter i nedtrekksmenyen for valg av maler.	<input checked="" type="checkbox"/>
Kunne navigere enkelt frem til ønsket funksjonalitet	Ja	Bestått	Bestått	Valgalternativene i editoren er oversiktlig, og enkle å forstå.	<input checked="" type="checkbox"/>
Kopiering av URL	Ja	Bestått	Bestått	"Vis link"-knappen får opp en URL, som man kan klikke på for å få opp visningen. Så kan man kopiere URL-en fra fanen.	<input checked="" type="checkbox"/>
Kunne velge klubb ut i fra navn på klubben og ikke ID nr	Ja	Ikke bestått	Ikke bestått	Vi har bare fått til å vise club-ID, og velge klubb ut fra det.	<input type="checkbox"/>
Forhåndsvisning av tavle viser	Ja	Ikke bestått	Ikke bestått	Lag, område og tid vises, men det blir	<input type="checkbox"/>

klubb, område og tid korrekt				ikke vist forskjell på farge mellom lagene.	
Tavlen blir lagret etter endring	Ja	Ikke bestått	Ikke bestått	Akkurat nå blir visningene bare lagret midlertidig.	

Oppleves den nye løsningen bedre enn den gamle løsningen?

Den nye løsningen der editoren for tavlevisning ligger i hovedmenyen på adminsiden, fungerer bedre enn den gamle løsningen der en måtte gå til en spesifikk URL for å komme til editoren.

Hva er bedre?

Den nye løsningen oppleves som mer brukervennlig da den er enklere å navigere til fra hovedsiden. Selve editoren har en bedre visuell utforming enn tidligere løsning, der den står i stil til resten av adminsiden. Dette gir en mer helhetlig brukeropplevelse av løsningen.

Hva kunne vært forbedret?

Navn på klubb kommer ikke opp i nedtrekksliste på editoren, da det bare var klubbid som kom opp. Visningen på valgt tavle fungerer ikke optimalt da klubbfarger ikke vises noe som gjør det vanskelig å skille de valgte aktivitetene fra hverandre. Visningen blir bare lagret midlertidig, noe som gjør at løsningen på dette tidspunktet ikke kan bli tatt i bruk av potensielle brukere av Stadiontavle.

5.6 Konklusjon

Vi utførte tre tester til sammen i testdokumentasjonen, hvor den første testen var enhetstesting, som ble utført for å teste samspillet mellom koden på klientsiden og serversiden ut fra funksjonen som ble testet. Testresultatene viste til at funksjonen ble kjørt og var optimal, selv om det var noen avvik.

Den andre testen vi gjorde var systemtesting, som ble utført for å teste sluttproduktet i sin helhet. Dermed ble brukerhistoriene opprettet. Resultatet til systemtestingen viser at systemet samsvarer med de spesifikasjonene som er satt for applikasjonen, men den er ikke helt optimal enda. Grunnen er fordi at det er en del funksjoner som kan bli videreutviklet, med tanke på at det er et par potensielle endringer som trengs for å gjøre det så lettvinnt som mulig. Les videre på [7.2 Potensielle endringer](#).

Den tredje testen var akseptansetesten der Digimaker fikk testet den ferdige løsningen. Brukerhistoriene var utgangspunktet for det som skulle sjekkes, og som forventet ble noen punkter ikke bestått da vi ikke rakk å implementere disse funksjonene i koden. Fem av åtte funksjoner ble derimot bestått i henhold til kravspesifikasjonen.

Kapittel 6: Brukerveiledning

6.1 Innledning

Brukerveileddingsdelen er med på å bidra til å sikre at brukeren lærer hvordan de får mest mulig ut av webapplikasjonen, og kan fungere som et oppslagsverk hvis de eventuelt trenger hjelp med hvor ting befinner seg. Imidlertid har vi valgt å begrense brukerveileddingen til å sette søkelys på vår funksjon av webapplikasjonen. Så målet med dette dokumentet er å gi brukeren en bedre forståelse og beskrivelse på hvordan de skal lage nye visninger på Stadiontavle.no. På denne måten blir det lettere for brukeren å oppfatte interaksjonens veivalg og visualisering.

6.2 Nettside - Stadiontavle.no

Når du først kommer på nettsiden, er dette det første du ser, vist på *Figur 6.2.1*. Du får en oversikt over hva siden innebærer ved hjelp av teksten på siden. Den eneste interaksjonsmuligheten du har på denne siden er «Logg inn»-feltet som er plassert øverst på høyre hjørne.



The screenshot shows the Stadiontavle.no website. At the top right is a login form with fields for 'Brukernavn' (Username) and 'Passord' (Password), and a 'Logg Inn' (Log In) button. Below the login form is a link 'Ny bruker Glemt passord' (New user/Forgot password). The main content area features a large stadium map with various sections labeled: 'P' (Parking), 'Klubbhus' (Clubhouse), 'Tribune' (Stand), 'MINI' (Minifield), 'Gress' (Grass), 'Kunstgress (KG)' (Artificial grass), and 'Inngang spillere/garderober' (Player/Changing room entrance) and 'Inngang publikum/kiosk' (Public/Shop entrance). A legend indicates that yellow dots represent player/changing room entrances and blue dots represent public/shop entrances. Below the map is a section for 'DIGIMAKER Connecting solutions'. The page also includes sections for 'Informasjonstavle' (Information board), 'Nødvendig utstyr' (Essential equipment), and 'Bruksområder' (Usage areas).

Figur 6.2.1: Stadiontavle.no

6.2.1 Logg inn

Systemet har en standard og enkel «Logg inn»-side. En eksisterende bruker må skrive inn sitt brukernavn som i dette tilfellet tilsvarer e-postadressen, og passordet sitt. For å komme seg videre til administrasjonssiden må brukeren trykke på den blå «Logg inn»-knappen vist på *Figur 6.2.1.1*.

The screenshot shows a login interface with the following fields:

- Username field: kisan@live.no
- Password field: masked with dots (.....)
- Log In button: Logg Inn
- Link below the button: Ny bruker Glemt passord

Figur 6.2.1.1: Logg inn

Ved feil innsetting av verdi på skriffeltene, vil en melding dukke opp (se *Figur 6.2.1.2*). Dette viser til at brukernavn eller passord ikke matcher eller er skrevet feil. Hvis brukeren ikke får logget inn, får hen andre innloggingsmuligheter. Enten ved å trykke på «Ny bruker» eller «Glemt passord» under «Logg inn»-knappen.

The screenshot shows a login interface with the following fields:

- Username field: kisan@live.no
- Password field: masked with dots (.....)
- Log In button: Logg Inn
- Error message box: Error Not accessible! (with a shield icon)
- Link below the button: Ny bruker Glemt passord

Figur 6.2.1.2: Error ved Logg inn.

6.3 Dashboard

Denne siden vil være startsiden når du først får logget deg inn på webapplikasjonen, og består av en vertikal navigasjonsbar på venstre side på *Figur 6.3.1.*



Figur 6.3.1: Hjemmeside

6.3.1 Navigasjonsbar

Navigasjonsbaren (*Figur 6.3.1.1*) på startsiden er en global menylinje, som vil si at navigasjonsbaren vil bli stående på samme sted hvor enn du befinner deg på webapplikasjonen. Denne baren blir brukt for å navigere seg til rundt om i nettsiden de har behov for.



Figur 6.3.1.1: Navigasjonsbar

6.4 Implementering av nye visninger

På menylinjen "Skjermer", vil du kunne lage dine egne valgte visninger som skal vises på Stadiontavle.no. På *Figur 6.4.1* får du et overblikk av denne siden. For å lage en ny visning må du klikke på det grønne ikonet under «Visningsmodus». Da vil det dukke opp et skjema vist på *Figur 6.4.2*.

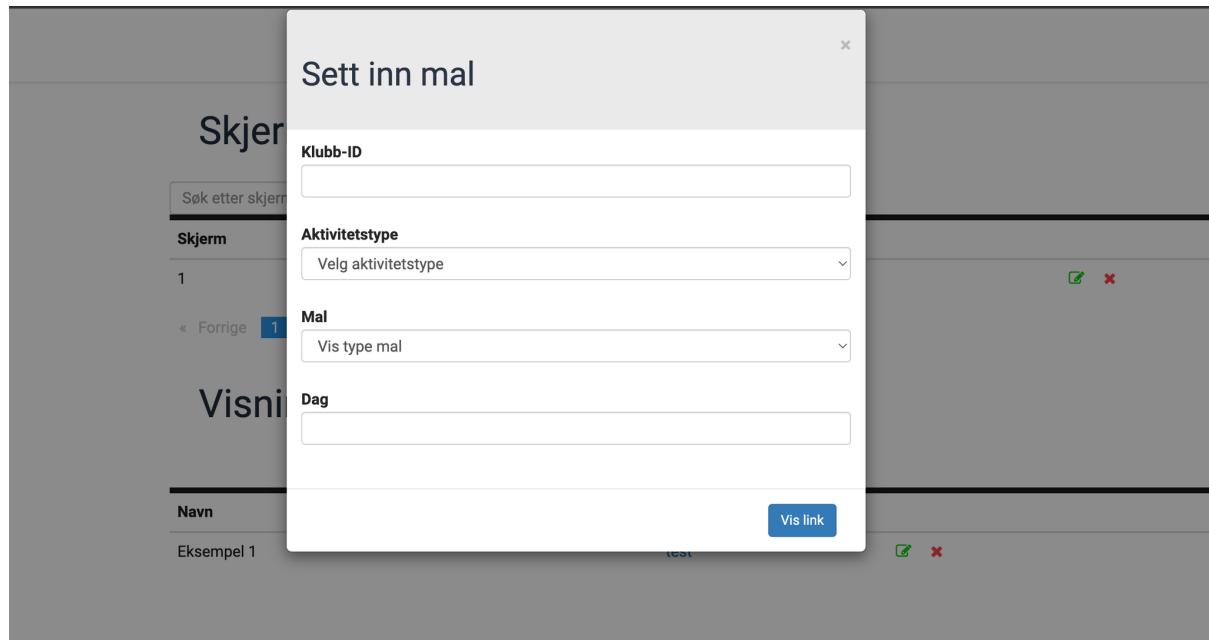


Skjerm	URL
1	https://stadiontavle.no/SAN /12

Visningsmodus

Navn	Mal
Eksempel 1	test

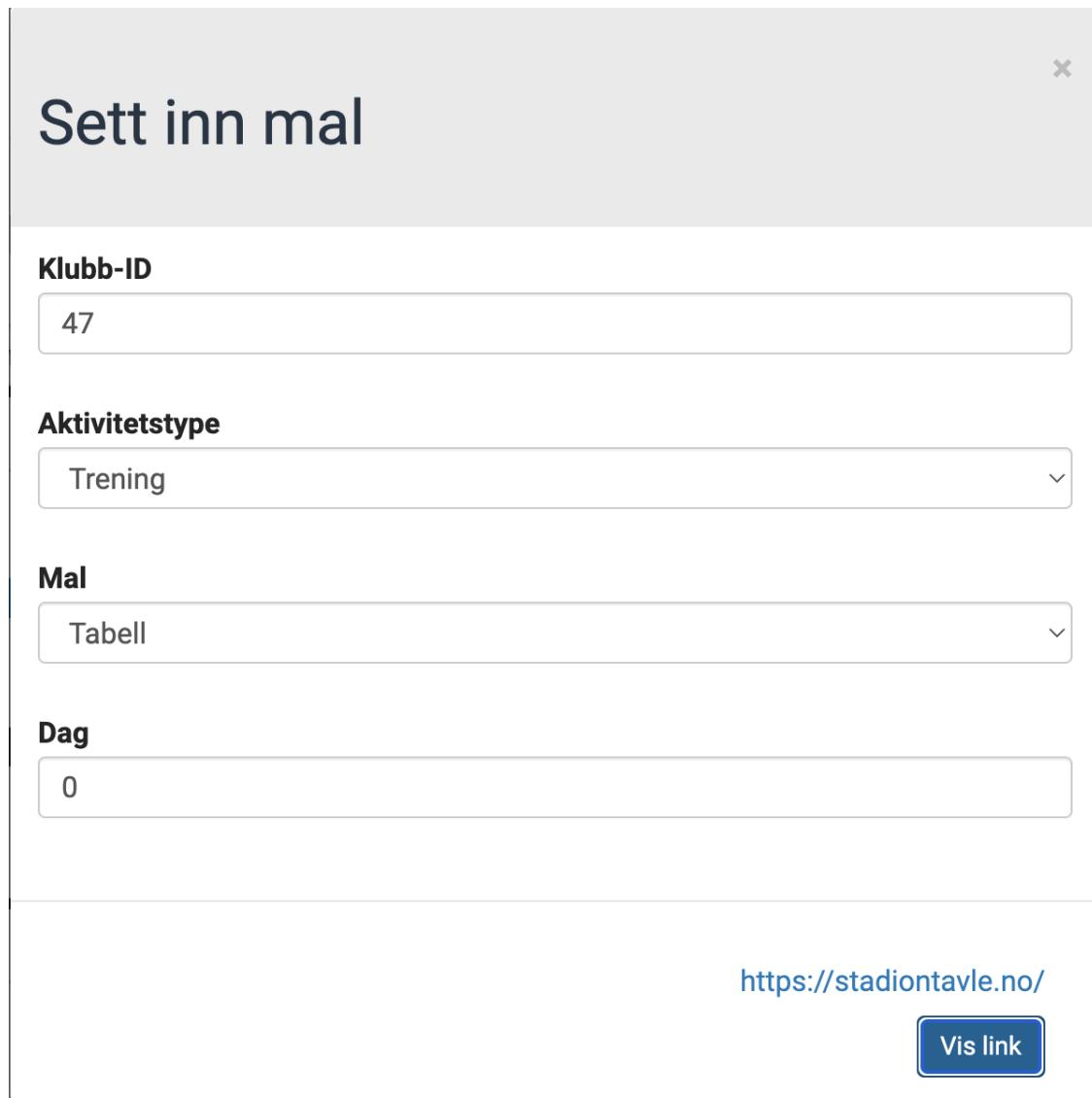
Figur 6.4.1: Skjermer-siden



Navn	Mal	Dag
Eksempel 1	test	

Figur 6.4.2: Skjema som dukker opp for å lage ny visning.

På Figur 6.4.2 er det krav på å fylle alle de fire forskjellig tekstfeltene for å lage en visning. Et eksempel på et utfylt skjema ser du på Figur 6.4.3.



Sett inn mal

Klubb-ID
47

Aktivitetstype
Trening

Mal
Tabell

Dag
0

<https://stadiontavle.no/>

Vis link

Figur 6.4.3: Eksempel på et utfylt skjema.



Klubb-ID
47

Figur 6.4.4: Klubb-ID tilsvarer 47 fordi vi er logget inn i en bruker som er en del av klubben «Sandefjord Ballklubb», som har klubb-ID verdi = 47.

Aktivitetstype

✓ Velg aktivitetstype

Kamp

Trening

Møte

Figur 6.4.5: På Aktivitetstype har jeg valgt Trening, fordi jeg vet jeg har allerede laget treninger på Trening-siden (Figur 6.4.6).

Tittel	Lag	Dato	Fra	Til	Pitch
test 1	VIF	2022-05-16	18:00	19:00	
test 2	LSK	2022-05-16	18:00	19:00	

Figur 6.4.6: Trening-siden, her kan jeg se eller legge til nye treninger ved å trykke på den blå knappen som heter «Ny». For å legge til treninger må du imidlertid habane (Figur 6.4.8) og lag (Figur 6.4.7).

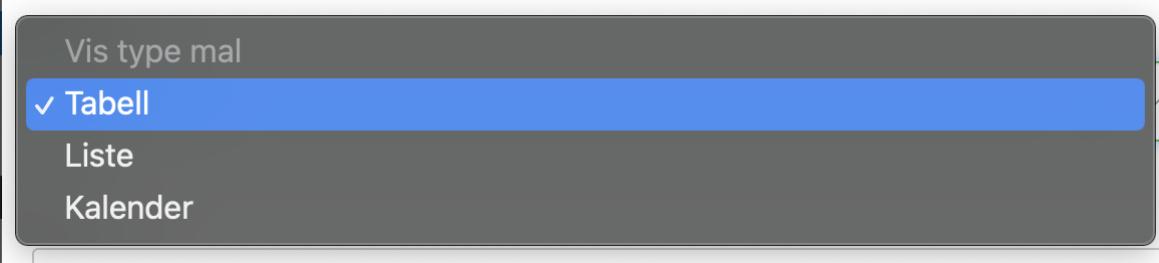
Visningsnavn	Garderobe	Gjestegarderobe
VIF	Garderobe 2	
LSK	Garderobe 1	

« Forrige 1 Neste »

Figur 6.4.7: Under kategorien «LAG» på menylinjen kan du se alle lagene du har lagt til. Samt for å legge til nye lag, gjør du av å trykke på «Nytt» knappen.

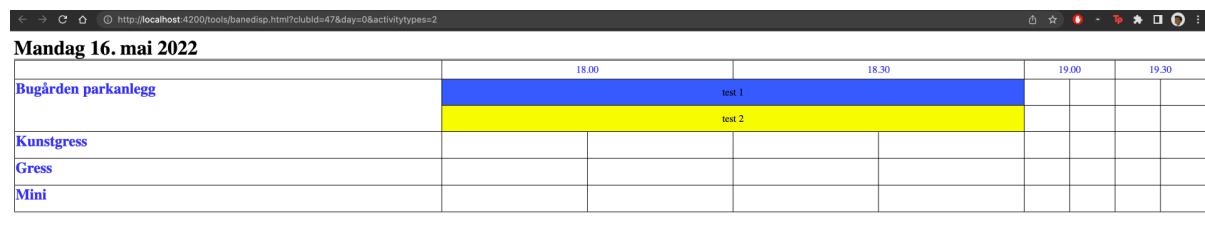
Baneoversikt	Vis inaktive
Bugården parkanlegg	
Kunstgress	
Gress	
Mini	

Figur 6.4.8: Under kategorien «BANNER» på menylinjen får du en oversikt over alle banene som finnes. Du kan legge til en nybane ved å trykke på «Ny»-knappen.



Figur 6.4.9: På mal-parametere har brukeren muligheten til å velge mellom tre ulike design på hvordan disse treningene skal vises. Brukeren kan velge mellom Tabellform (Figur 6.4.10),

Listeform (Figur 6.4.11) eller Kalenderform (Figur 6.4.12).



	18.00	18.30	19.00	19.30
Bugården parkanlegg		test 1		
		test 2		
Kunstgress				
Gress				
Mini				

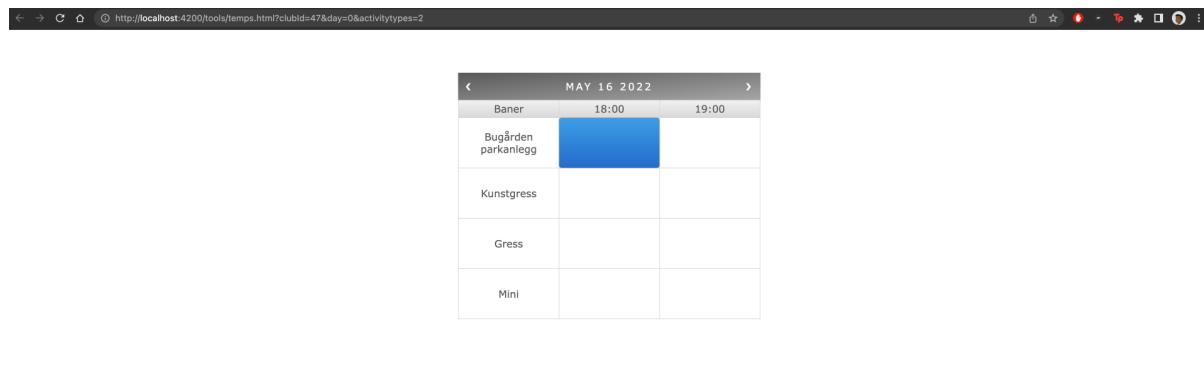
Figur 6.4.10: Visning for Tabellform.



Mandag 16. mai 2022

- Bugården parkanlegg
 - test 1
 - 18:00:00 TO 19:00:00
 - 2022-05-16
 - test 2
 - 18:00:00 TO 19:00:00
 - 2022-05-16
- Kunstgress
- Gress
- Mini

Figur 6.4.11: Visning for Listeform.



Figur 6.4.12: Visning for Kalenderform.

Dag

0

Figur 6.4.13: I dette feltet velger du hvilken dato aktivitetstypen skal bli utført på. Hvis du skriver inn tallet null, vil det automatisk sette nåværende dato, altså datoene i dag. Mens hvis du hadde skrevet inn tallet ett, ville systemet satt datoene for neste dag.

<https://stadiontavle.no/>

Vis link

Figur 6.4.14: Når du har fylt ut skjemaet og trykker på knappen «Vis link», vil du bli gitt en link for å forhåndsvise din visning ut fra parameterne du har satt inn. Av skjemaet på Figur 6.4.3, vil du få en link som tilsvarer visningen på Figur 6.4.10.

Sett inn mal

X

Klubb-ID

Aktivitetstype

Velg aktivitetstype

Mal

Tabell

Dag

<https://stadiontavle.no/>

Vis link

Figur 6.4.15: Du vil fortsatt bli gitt en link selv om det ikke har blitt lagt til noen parametere.

Imidlertid vil du ikke få noe visning når du trykker på linken, som vist på Figur 6.4.16.



Onsdag 18. mai 2022

Ingen aktivitet

Figur 6.4.16: Visning for et skjema som ikke er fylt ut.

Kapittel 7: Konklusjon og refleksjon

7.1 Konklusjon

Som nevnt i [2.2.1 Kartlegging](#) fikk teamet i oppgave å videreutvikle eksisterende webløsning der det skulle implementeres nye visninger med minimalt behov for programmering. Kravene var som følgende:

- Flere valgmuligheter for visning av aktiviteter i form av ferdige layouts
- Flere parametere i editoren på administrasjonssiden for brukerne. Valg av visningstype, farger mm.
- Etablering av nye maler med HTML og kode for aktiviteter for Administrator
- Det skal brukes samme kodespråk som den eksisterende løsningen, men det er åpning for forslag av andre løsninger som kan iverksettes inn i den nåværende.

Ved innlogging som admin får bruker mulighet til å:

- Opprette forskjellige layouts
- Velge mellom forskjellige parametere, i dette tilfellet valgte vi å prioritere visningstype. Endre malene som blir opprettet
- Se de ulike visningene av aktiviteter

Vi konkluderte med at det var viktig å fokusere på å bli ferdig med de funksjonelle kravene først, og deretter de ikke-funksjonelle kravene. Som følge av dette, og mangel på kodekunnskap, ligger forbedringsmulighetene stort sett i backend-delen av prosjektet.

Med tilbakemelding fra veiledere i Digimaker, kan vi si oss delvis fornøyde med oppgaven. Vi har klart å forholde oss til noen av kravene som ble satt, men det er fortsatt forbedringsmuligheter på noen områder.



7.2 Potensielle endringer

Mens teamet jobbet med det endelige produktet, la vi merke til områder med potensielle forbedringer i frontend-delen. Her vil vi gjøre rede for mulige endringer vi kunne gjort med tanke på at applikasjonen er rettet mot kunder. Det er viktig for oss å sette søkelys på hvordan informasjonen på nettsiden blir presentert, fordi det vil ha en stor innvirkning på brukeropplevelsen. Samtidig kan et godt design hjelpe brukeren med å utføre handlingene de ønsker, enklere og mer effektivt, så de fortsatt kan utføre oppgavene de trenger å gjøre.

7.2.1 Klubb-ID

Klubb-ID parametere synes vi er hensiktsløst å ha med på «Sett inn mal» skjemaet vist på *Figur 6.4.2*. Dette er fordi brukeren som allerede har logget inn er en del av en klubb, som er lagret i databasen som brukerens Klubb-ID. Vi hadde med denne parameteren fordi vi gikk ut i fra den eksterne siden for Stadiontavle kalt «Tools» (*Figur 2.2.2.10*). I noen deler av applikasjonen hadde vi begrenset rettigheter for valg av design, som for eksempel databasen. Dette førte til mangel av hvordan informasjonen på nettsiden ble representert. Så vi ville tatt bort Klubb-ID parameteret hvis det var mulig.

7.2.2 Maler

Tabellform (*Figur 6.4.10*) og Kalenderform (*Figur 6.4.12*) har muligheten til å forbedres. Med tanke på at flere aktivitetstyper kan skje samtidig, altså samme dag og tid, vil dette ikke bli så klart representert i visningene. På tabellformen vil to helt forskjellige aktivitetstyper ha samme farge, som kan føre til misforståelse på brukerens side. Når det kommer til brukeropplevelsen, er riktig bruk av farger en viktig faktor. Ved å ha atskilt aktivitetstypene med farger vil det bidra til en bedre oppfatning av den visuelle informasjonen for brukeren. På kalenderformen vil aktivitetstypene ligge under hverandre når de har blitt planlagt samtidig. Dette vil også påvirke brukerens visuelle persepsjon negativt. Et eksempel på hvordan en kan løse dette problemet, er å sette aktivitetstypene rett under hverandre i samme kolonne. Dette gjør det tydeligere for brukeren, slik at vedkommende kan kunne skille mellom de forskjellige aktivitetstypene.

7.2.3 Design

Siden vi hadde begrensede rettigheter ved utformingen av løsningen, vurderte vi hvordan vi kunne videreutvikle nettsiden for å tilby brukeren flere interaksjonsmuligheter. Med å implementere en editor inn på "Skjermer"-siden (*Figur 6.4.1*), vil brukeren få muligheten til å velge egne farger og former som passer deres behov. Dette vil gi større variasjon og mer interaksjonsmuligheter mellom bruker og applikasjon, med tanke på visualisering.

7.3 Refleksjon

Prosjektet har vært krevende både i tid og gjennomførelse. Vi har hatt en lærerik prosess hvor vi har fått innsikt i ulike faser av et prosjekt fra start til slutt. Vi har fått innsikt i hvordan vi kan skape et godt samarbeid innad i et team og med oppdragsgiver, samt i selve utviklingsprosessen. Teorien om Scrum og prosjektorganisering har vi også fått mulighet til å praktisere under arbeidet av prosjektet. Vi fikk bruke kunnskapen vi har tilegnet oss under studietiden i dette prosjektet, og er etter arbeidet mer bevisst på hvordan det vil være å jobbe med større utviklingsprosjekter i arbeidslivet. Gruppen har hatt stort læringsutbytte både personlig og faglig. Alle i gruppen kan være enige om at produktivitet, kommunikasjon og oppmøte er viktig. Dette er noe vi også kan ta med oss videre i fremtidig prosjektarbeid.

Ved prosjektarbeid vil det likevel være utfordringer. Gruppen kom sammen frem til at det som tok lengst tid var kartleggingen, mer spesifisert, skisseringen og forståelse av koden. På starten av prosjektet nevnte veileder og teamet i Digimaker at koden kunne være vanskelig å forstå og at oppgaven dermed kunne bli utfordrende å ferdigstille etter alle kravspesifikasjoner. Noe av kodespråket hadde vi erfaring med fra tidligere, men selve oppsettet og prosjektet i sin helhet var allerede stort, der det krevde en del kode kompetanse å forstå den eksisterende løsningen. En annen utfordring var at alle gruppemedlemmene hadde ulike timeplaner og dermed ble mye av kommunikasjonen gjort digitalt. Det krevde ekstra hardt arbeid individuelt der det var viktig med en god plan for arbeidsfordelingen. Vi anser oss på tross av dette fornøyde med samarbeidet innad i gruppen, og med Digimaker.

Kapittel 8: Vedlegg

8.1 Prosjektdagbok

Bachelorprosjekt på OsloMet våren 2022 - Gruppe 62

av: Dani Tran, Kisan Srirangan, Marthe V. Marstrander og Linda Kadrijaj

Denne dagboken inneholder kun gruppemøter - enten i form av planlagte gruppemøter eller av møter med bedriften, veiledere osv. Alle gruppemedlemmene jobbet med prosjektet individuelt også. Vi opprettet også en gruppechat med alle gruppemedlemmer både på Messenger og Discord, så det var også flere samtaler der, enten mellom hver av oss, eller som en gruppe.

Tirsdag 19. oktober 2021 (23:00 - 00:30)

Sted: Discord

Tilstede: Alle

- Vi hadde vårt første møte i kveld, der vi la plan for hvilke bedrifter vi skal kontakte. Det har blitt opprettet en mappe i Google Drive, og en felles søknad har blitt skrevet og godkjent. Planen er å bruke resten av uken til å sende ut søker og finne en veileder. Det skal så skrives en felles statusrapport som skal sendes inn til skolen innen 25. oktober.

Mandag 25. oktober 2021 (22:00 - 23:00)

Sted: Discord

Tilstede: Alle

- I dag gikk vi gjennom tilbakemeldinger og svar fra selskaper og bedrifter vi kontaktet angående bacheloroppgaven vår. Vi har ikke fått inn så mange svar, og vi har fortsatt ikke funnet en oppgave. Imidlertid har vi blitt ferdige med å skrive statusrapporten, og vi har fått sendt den inn.

Torsdag 4. november 2021 (13:00 - 13:40)**Sted:** Digimaker (Helsfyr)**Tilstede:** Alle

- Vi var på vårt første intervju med en potensiell bedrift for bacheloroppgaven. Vi ble presentert med en løsning for Fantasy Fotball som de ønsket at vi skulle videreutvikle. Vi fikk informasjon om teknologiene de bruker, samt om selskapet forøvrig. Vi har fått tilsendt en beskrivelse av oppgaven på mail, samt blitt registrert som administrator for en del av siden slik at vi kan utforske den eksisterende løsningen.

Mandag 8. november 2021 (22:30 - 00:00)**Sted:** Discord**Tilstede:** Alle

- Vi hadde et møte der vi gikk gjennom prosjektbeskrivelsen vi hadde fått fra bedriften. Vi diskuterte ulike måter på å løse oppgaven på.

Mandag 15. november 2021 (22:30 - 23:30)**Sted:** Discord**Tilstede:** Alle

- Vi har skrevet prosjektskissen og sendt den inn. I tillegg har vi planlagt hvordan vi skal jobbe fremover, og hva som må gjøres før neste innlevering, altså "Innlevering av kontrakt". En av oppgavene som må utføres før denne fristen er å utvikle en fremdriftsplan, som skal presenteres for bedriften. Dette skal presenteres 09.12 på OsloMet.

Mandag 6. desember 2021 (16:00 - 17:00)**Sted:** Messenger**Tilstede:** Alle

- I dag begynte vi på presentasjonen vi skal fremføre, for bedriften og andre grupper som skal skrive oppgaven deres hos den samme bedriften, på møtet 09. desember. I presentasjonen skal vi i 15 min presentere om oss selv og selve oppgaven. Vi skal også nevne om det vi anser som de største utfordringene med oppgaven vår.

Onsdag 8. desember 2021 (19:00 - 19:30)**Sted:** Discord**Tilstede:** Dani, Kisan og Marthe (**Forfall:** Linda pga jobb)

- Vi gikk gjennom presentasjonen vår, og snakket om endringer vi kunne gjøre på fremdriftsplanen. I morgen skal vi presentere for bedriften.

Torsdag 9. desember 2021 (09:00 - 12:00)**Sted:** OsloMet - P35 PH461**Tilstede:** Dani, Kisan og Marthe (*Linda ble med på Teams*)

- I dag hadde vi et møte med Digimaker, hvor vi presenterte oppgaven til dem om hvordan vi skal løse den, og hvor de gikk gjennom om bedriften deres og teknologiene vi skal bruke. Vi fikk veiledning angående oppgaven, og fikk oppklart det vi lurte på.

Onsdag 5. januar 2022 (12:00 - 13:15)**Sted:** Discord**Tilstede:** Alle

- I dag ble vi enige om å opprette kontakt med veileder som vi har fått utdelt. Vi har sendt mail og ønsker et møte for å planlegge prosjektarbeidet og hvordan veiledning skal foregå. Vi har fordelt arbeidsoppgaver i forprosjektrapporten og har avtalt ukentlige gruppemøter hvor vi oppdaterer hverandre om hvor langt vi har kommet med de ukentlige målene og i prosjektet generelt. Fast gruppemøte hver onsdag kl 12:00 er satt.

Mandag 10. januar 2022 (10:00 - 10:30)**Sted:** Teams**Tilstede:** Alle

- Vi hadde et møte med Digimaker hvor vi fikk oppklart noen spørsmål som vi hadde fra forprosjektet.

**Tirsdag 11. januar 2022 (12:00 - 12:30)****Sted:** Zoom**Tilstede:** Dani, Kisan og Marthe (**Forfall:** Linda pga jobb)

- I dag hadde vi et møte med veilederen vår, Roza Abolghasemi, på Zoom. Vi introduserte oss til hverandre, og hun forklarte oss generelt om hva hun kunne hjelpe oss med, samt ga hun oss tips om hva vi bør gjøre i løpet av prosjektet.

Onsdag 12. januar 2022 (12:00 - 14:00)**Sted:** Discord**Tilstede:** Alle

- Vi hadde det første ukentlige gruppemøte, hvor vi jobbet med vår egen del av forprosjektet.

Onsdag 19. januar 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Dani og Kisan (**Forfall:** Linda pga jobb og Marthe pga annet skolearbeid)

- Vi fortsatte med å jobbe med forprosjektet. Linda og Marthe kunne ikke delta på møtet, så de jobbet individuelt.

Mandag 24. januar 2022 (12:30 - 13:30)**Sted:** Discord**Tilstede:** Alle

- I dag gikk vi igjennom forprosjektet vårt med hele gruppa. Vi fullførte og leverte det på Canvas. Vi planla også startoppsett for selve prosjektet og programmeringen.

Onsdag 26. januar 2022 (-)**Sted:** -**Tilstede:** -

- Ukens gruppemøte uteble. Vi brukte tiden på å se på videoer angående Angular og Laravel, for å prøve å forstå kodespråkene.

**Onsdag 02. februar 2022 (12:00 - 12:30)****Sted:** Discord**Tilstede:** Alle

- Vi brukte tiden på å sette opp et par spørsmål vi hadde fra møtet vi hadde med Digimaker på OsloMet, og ting vi har lurt på senere, angående oppgaven og siden deres. Vi skal ha et møte med dem på mandag, så vi brukte tiden på å forberede oss til det. Vi starter nå med opplæringen av de ulike programmeringsspråkene i henhold til sprint 2.

Mandag 07. februar 2022 (14:00 - 14:30)**Sted:** Teams**Tilstede:** Alle

- I dag hadde vi det første ukentlige statusmøte med Digimaker i dag, hvor vi fikk en kodegjennomgang og oppdaterte hverandre om hvordan vi tenker å jobbe videre de neste ukene for å få begynt med utviklingen.

Onsdag 09. februar 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- Vi hadde idémyldring og brukte tiden på å lage forskjellige startskisser, som vi kunne vise frem på møtet på mandag.

Mandag 14. februar 2022 (14:00 - 14:30)**Sted:** Teams**Tilstede:** Alle

- I dag hadde vi det ukentlige statusmøte med Digimaker, der vi så på skisser og gikk igjennom adminsiden på Stadiontavle. Vi avtalte videre å ha et fysisk møte hos Digimaker på torsdag denne uken, for en mer grundig opplæring på hvordan vi bruker den eksisterende løsningen.

Onsdag 16. februar 2022 (-)**Sted:** -**Tilstede:** -

- Vi avlyste ukens gruppemøte av den grunn at vi skal møtes fysisk i morgen hos Digimaker.

Torsdag 17. februar 2022 (16:30 - 18:00)**Sted:** Digimakers kontor på Helsfyr**Tilstede:** Dani, Kisan og Marthe (**Forfall:** Linda pga personlige grunner)

- Vi møtte opp på kontoret til Digimaker for å jobbe sammen fysisk. Der fikk vi en grundigere veiledning om den eksisterende løsningen og hvordan vi kunne jobbe videre med den. Gruppa satte også opp en plan for hva vi skulle gjøre i løpet av den neste uka.

Mandag 21. februar 2022 (14:00 - 14:30)**Sted:** Teams**Tilstede:** Dani og Kisan (**Forfall:** Linda og Marthe pga ferie)

- I dag skulle vi ha vår ukentlige møte med Digimaker, men veilederen vår i Digimaker, Christian, kunne ikke delta. Skisser som vi laget på torsdag og senere, ble dermed sendt til han på mail og vi forklarte hvordan vi tenkte vi kunne løse et par områder på den eksisterende løsningen.

Onsdag 23. februar 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- Vi fikk tilbakemelding fra Christian om skissene vi hadde sendt han på mandag. Vi brukte dette møtet til å forbedre på skissene til neste møte vi har med han på mandag.

**Mandag 28. februar 2022 (14:00 - 14:30)****Sted:** Teams**Tilstede:** Alle

- Dagens møte gikk ut på å planlegge skisser. Vi avtalte å møtes på torsdag for å få hjelp til å sette opp prosjektet på maskinene våre.

Onsdag 02. mars 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- Dagens møte gikk ut på å forklare skissene vi hadde laget, og få tilbakemeldinger på de. Kommentarene var gode, og det trengs ikke noen store endringer på skissene. Vi avtalte å møtes på torsdag for å få hjelp til å sette opp prosjektet på maskinene våre.

Torsdag 03. mars 2022 (12:15 - 16:15)**Sted:** Digimakers kontor**Tilstede:** Dani og Kisan (*Linda og Marthe ble med på Messenger*)

- Vi fikk jobbe på kontoret til Digimaker i dag, og fortsatte på å forbedre skissene våres. Vi fikk også noe hjelp til å sette opp prosjektet på maskinene våre.

Onsdag 09. mars 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- Ukens gruppemøte gikk ut på å snakke om skissene, og på å forbedre de.

Torsdag 10. mars 2022 (12:00 - 15:30)**Sted:** Digimakers kontor**Tilstede:** Dani, Kisan og Marthe (*Forfall: Linda pga jobb*)

- Det ukentlige møtet som skulle ha vært på mandag, ble flyttet til torsdag. Vi fikk også ha det på kontoret til Digimaker. Dermed fikk vi snakket med Christian og vi fikk kommentarer direkte om skissene og ellers til oppgaven også. Kisan og Marthe som har Mac fikk satt opp prosjektet helt på maskinene deres.

**Mandag 14. mars 2022 (14:00 - 14:15)****Sted:** Teams**Tilstede:** Alle

- På denne ukas faste møte med Digimaker, viste vi fram en skisse og avtalte å ha et fysisk møte på kontoret kommende torsdag.

Onsdag 16. mars 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Kisan og Marthe (Forfall: Linda pga jobb og Dani pga annet skolearbeid)

- Ukens gruppemøte gikk på å legge plan for å skrive på rapporten.

Torsdag 17. mars 2022 (12:30 - 14:30)**Sted:** Digimakers kontor**Tilstede:** Alle

- I dag jobbet vi alle sammen på kontoret til Digimaker. Dani og Linda fikk satt opp prosjektet på maskinene deres i dag. Vi fortsatte med å skrive videre på rapporten.

Onsdag 23. mars 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- Vi brukte tiden på å skrive videre på rapporten, og forstå helt hvordan siden fungererer.

Torsdag 24. mars 2022 (12:00 - 15:30)**Sted:** Digimakers kontor**Tilstede:** Dani og Kisan (Forfall: Linda og Marthe pga annet skolearbeid)

- Dani og Kisan møtte opp på kontoret og oppdaterte Christian med hvor langt vi har kommet. Mandagsmøtene har blitt flyttet til torsdag, siden det passer bedre for å oppdatere Christian ukentlig om hva vi har gjort, mens på onsdager jobber gruppa sammen videre med prosjektet som vanlig, og på mandager når enn det trengs.

Onsdag 30. mars 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- I dag brukte vi tiden på å se gjennom koden, og se på filstruktur, kodestruktur osv.

Onsdag 06. april 2022 (12:00 - 14:00)**Sted:** Discord**Tilstede:** Alle

- Ukens gruppemøte gikk ut på å kode malene. Vi prøvde oss frem ved å endre på den eksisterende malen som Digimaker hadde kodet fra før av.

Torsdag 07. april 2022 (11:30 - 15:30)**Sted:** Digimakers kontor**Tilstede:** Dani og Kisan (*Linda og Marthe på Messenger*)

- I dag la vi frem et plan på hvordan vi skulle fordele oppgavene mellom oss i løpet av de neste ukene, for å kunne bli ferdig med et ferdig utkast som skal sendes til Hedvig. Samtidig skal vi ha en prototype klar for å vises til Digimaker etter påske.

Onsdag 13. april 2022 (-)**Sted:** -**Tilstede:** -

- Det ukentlige gruppemøtet ble avlyst på grunn av påske.

Onsdag 20. april 2022 (12:00 - 14:00)**Sted:** Discord**Tilstede:** Alle

- Vi jobbet videre både på koden og rapporten, etter hvordan vi hadde fordelt arbeidsoppgavene mellom oss.

Torsdag 21. april 2022 (21:00 - 22:00)**Sted:** Discord**Tilstede:** Dani, Linda og Marthe (**Forfall:** Kisan pga personlige grunner)

- Siden vi er en av de gruppene som har en engelskspråklig veileder, har vi et tilbud om å kontakte Hedvig Bjørge som jobber som rådgiver for skriveverkstedet ved NMBU. Vi klargjorde et førsteutkast som vi kunne sende til Hedvig, for å få hjelp og kommentarer på det språklige og struktur. Førsteutkastet viser innledningen vår og hvordan vi har valgt å strukturere prosjektrapporten vår.

Mandag 25. april 2022 (14:00 - 15:30)**Sted:** Digimakers kontor**Tilstede:** Kisan (*spontant møte*)

- Aksel Susegg som jobber for Digimaker, er den som har jobbet mest med Stadiontavle-løsningen. Så for å få hjelp og veiledning med kode, møtte Kisan han på kontoret til Digimaker i dag.

Onsdag 27. april 2022 (12:00 - 12:30)**Sted:** Discord**Tilstede:** Alle

- Det ukentlige gruppemøtet brukte vi på å fokusere fullt ut på kode. Vi skal bruke tiden i morgen når vi er samlet sammen hos Digimaker til å skrive videre på rapporten.

Torsdag 28. april 2022 (12:00 - 15:30)**Sted:** Digimakers kontor**Tilstede:** Alle

- I dag møttes hele gruppa på kontoret til Digimaker, og jobbet med tilbakemeldingen vi fikk fra Hedvig. Det som ble kommentert på, ble rettet opp og formulert på en bedre måte. Vi satte opp en plan for hva som må gjøres i løpet av de neste ukene opp mot fristen til å levere bacheloroppgaven. Planen er blant annet om å ha møter med Aksel og Roza (veilederen vår), for både hjelp og for videre veiledning. Etterhvert som

vi blir ferdig med å kode, skal vi lage en spørreundersøkelse/brukertesting, og sende denne ut for å få kommentarer på arbeidet vårt.

I neste uke (uke 18) skal Dani og Marthe fokusere på frontend (implementere visninger), mens Kisan fokuserer på backend (database). Dani og Kisan skal skrive om frontend og backend i rapporten. Marthe og Linda skal skrive videre på kravspesifikasjon og om sprintene.

I uke 19 er planen å bli ferdig med all kode i løpet av uka. Dani og Kisan fortsetter videre på skrivingen om front- og backend i rapporten, og skriver også om testdokumentasjon og brukerveiledning. Linda og Marthe lager spørreundersøkelsen, og sender denne ut til medstudenter.

Mandag 02. mai 2022 (09:00 - 12:00)

Sted: OsloMet, Messenger

Tilstede: Alle

- I dag skulle vi egentlig ha hatt et møte med Aksel, men siden han ikke kunne allikevel, valgte gruppa å jobbe sammen på OsloMet. Vi fulgte planen vår for denne uka, og vi er godt på vei med oppgavene.

Onsdag 04. mai 2022 (12:00 - 14:00)

Sted: Discord

Tilstede: Alle

- Vi jobbet videre på de oppgavene vi skulle jobbe med. Både koden og rapporten blir jobbet med parallelt.

Torsdag 05. mai 2022 (09:00 - 12:00)

Sted: OsloMet

Tilstede: Alle

- Gruppa kom frem til en enighet om å jobbe på OsloMet i dag for å komme i mål med oppgavene vi har for denne uka. Planen er å ha et siste statusmøte med Christian hos Digimaker neste torsdag. Hvis vi lurer på noe, kan vi kontakte Aksel på Teams og evt.

møte han mandag på Digimakers kontor. Vi tenker også å kontakte Roza for et møte neste fredag angående struktur på rapporten og tips for sluttspurten av prosjektet.

Mandag 09. mai 2022 (09:00 - 12:00)

Sted: Discord

Tilstede: Alle

- Vi jobbet videre på koden, og nærmer oss det vi ønsker av layout.

Onsdag 11. mai 2022 (12:00 - 14:00)

Sted: Discord

Tilstede: Dani, Linda og Marthe (**Forfall:** Kisan pga sykdom)

- Ukens gruppemøte gikk ut på å jobbe videre med koden og rapporten.

Torsdag 12. mai 2022 (09:00 - 12:00)

Sted: Discord

Tilstede: Dani, Linda og Marthe (**Forfall:** Kisan pga sykdom)

- Siden Kisan er syk, og vi ikke har blitt ferdig med koden vår, har vi valgt å flytte dagens møte med Digimaker til mandag. Vi trenger mer tid, pga. et par bugs.

Fredag 13. mai 2022 (09:00 - 12:00)

Sted: Discord

Tilstede: Alle

- Kisan er tilbake, og i dag jobbet vi videre både på koden og rapporten. Begge arbeidene gikk parallelt.

Mandag 16. mai 2022 (14:30 - 15:30)

Sted: Digimakers kontor

Tilstede: Alle

- I dag hadde vi vår siste statusmøte med Digimaker. Først brukte vi tiden på å forberede oss og finpusse litt på koden. Vi viste senere fram sluttproduktet vårt - altså det vi hadde fått til av kode, og oppgaven. Christian og Chen (utviklingssjef) var tilstede under møtet.

**Onsdag 18. mai 2022 (09:00 - 18:00)****Sted:** OsloMet, Messenger**Tilstede:** Alle

- Gruppa møttes i dag på OsloMet for å sitte og skrive på rapporten sammen. Vi har fordelt forskjellige deler mellom hverandre, og skriver på de delene vi skal skrive på.

Torsdag 19. mai 2022 (09:00 - 15:00)**Sted:** OsloMet, Messenger**Tilstede:** Alle

- I dag fortsatte videre på skrivingen av rapporten, akkurat som i går.

Fredag 20. mai 2022 (09:00 - 15:00)**Sted:** OsloMet**Tilstede:** Alle

- I dag fikk vi ha et møte med den interne veilederen vår, Roza, og på møtet gikk vi gjennom strukturen vår hittil på rapporten. Vi fikk kommentarer og tips fra henne, som vi skal ta i bruk videre i skrivingen. Videre skal vi i dag og ut helgen fortsette med skrivingen av rapporten, men individuelt. Målet er å skrive ferdig så mange deler som mulig til søndag kveld.

Mandag 23. mai 2022 (09:00 - 21:00)**Sted:** Messenger**Tilstede:** Alle

- Vi fortsatte på skrivinga, og hadde statusmøter kl. 09, 15, og 21. Planen er å skrive ferdig alle deler til i morgen kveld, og så skal vi ha et møte for å gå gjennom, finpusse og levere rapporten.

Tirsdag 24. mai 2022 (21:00 - 06:00)**Sted:** Discord, Messenger**Tilstede:** Alle

- Nå er vi ferdige med rapporten, og skal levere inn på Inspera.

8.2 Oppgavebeskrivelse fra Digimaker

Teknologi

Laravel

Angular

MySQL

Bakgrunn

I vår løsning, stadiowntavle.no, er det tilrettelagt for visning av aktiviteter knyttet mot idrettsanlegg (baner, garderober og andre lokaliteter). Vi ønsker å kunne implementere nye visninger med minimalt behov for programmering. Per i dag har vi tabellbasert visning, uten særlig mye tilpasningsmuligheter når det gjelder farger og layout.

Ønsker om funksjonalitet

Vise maler i administrasjonsgrensesnittet

- Valg av parametere
 - o ClubID er automatisk
 - o Aktivitetstyper
 - o Dag
 - o + andre aktuelle
 - o Farger (CSS?)
- Presentasjon av visning
- Kopiering av URL, lagre visning for bruk i editor

Valg av visning i editor

Etablering av nye maler (HTML + koder for aktiviteter) – dette er ikke aktuelt for brukere/kunder – kun for vår egen del.

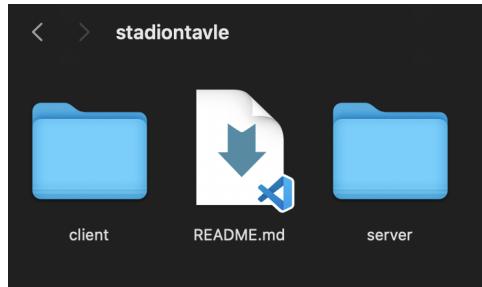
Styring av farger etc. (CSS?)

Kan også være aktuelt å se på generell bruk av editoren

8.3 Installasjonsveiledning

Denne delen er tatt rett fra [README.me-filen](#) fra GitHub-repositoryen til Digimaker for Stadiontavle, og har blitt oversatt til norsk.

README.me-filen kan du også finne i prosjekt zip mappen, som vist på bildet under:



Oppsett for Windows

Nyttige programmer

- xampp 7.4.x / php 7.4.x
- Git Bash
- Github Desktop

Oppsett av xampp

1. Plasser repositoryen inn i **xampp/htdocs/** mappen.

Denne PCen > Windows (C) > xampp > htdocs				
	Navn	Endringsdato	Type	Størrelse
⚡	dashboard	13.02.2022 22:53	Filmappe	
⚡	img	13.02.2022 22:53	Filmappe	
⚡	stadiontavle	20.05.2022 19:15	Filmappe	
⚡	webalizer	13.02.2022 22:53	Filmappe	
⚡	xampp	13.02.2022 22:53	Filmappe	
⚡	applications	27.08.2019 16:02	Chrome HTML Do...	4 kB
⚡	bitnami	27.08.2019 16:02	CSS-dokument (Ca...	1 kB
⚡	favicon	16.07.2015 17:32	Ikon	31 kB
⚡	index	16.07.2015 17:32	PHP Source File	1 kB

2. Finn og rediger xampp/apache/conf/httpd.conf.

	Navn	Endringsdato	Type	Størrelse
extra	13.02.2022 22:57	Filmappe		
original	13.02.2022 22:53	Filmappe		
ssl.crt	13.02.2022 22:53	Filmappe		
ssl.csr	13.02.2022 22:53	Filmappe		
ssl.key	13.02.2022 22:53	Filmappe		
charset.conv	23.12.2021 12:04	CONV-fil	2 kB	
httpd.conf	13.02.2022 23:11	CONF-fil	22 kB	
magic	23.12.2021 12:04	Fil	14 kB	
mime.types	21.01.2022 17:46	TYPES-fil	60 kB	
openssl.cnf	14.12.2021 17:45	CNF-fil	11 kB	

3. Erstatt linjene:

DocumentRoot "C:/xampp/htdocs"

<Directory "C:/xampp/htdocs">

med:

DocumentRoot "C:/xampp/htdocs/stadiontavle/client/src"

<Directory "C:/xampp/htdocs/stadiontavle/client/src">

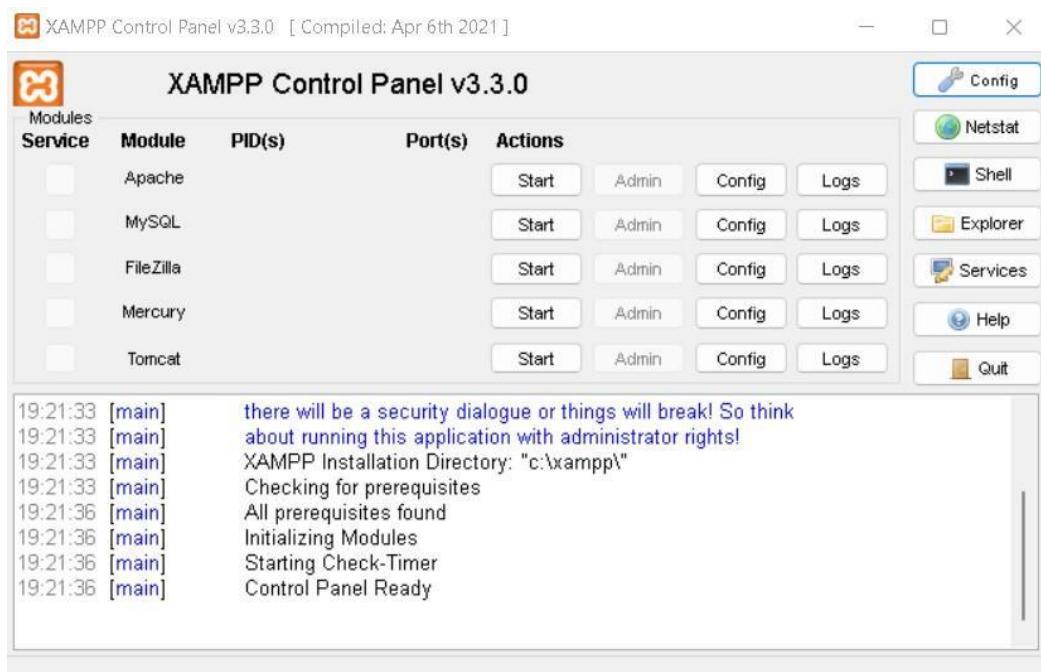
```

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this
directory, but
# symbolic links and aliases may be used to point to other
locations.
#
DocumentRoot "C:/xampp/htdocs/stadiontavle/client/src"
<Directory "C:/xampp/htdocs/stadiontavle/client/src">
#
# Possible values for the Options directive are "None",
"All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksIfOwnerMatch
ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* ---.
"options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important.
Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#

```

4. Start oppxampp ved å kjøre **xampp-control.exe**.

Denne PCen > Windows (C:) > xampp				
	Navn	Endringsdato	Type	Størrelse
⚡	properties	13.02.2022 22:57	Konfigurasjonsinnst...	1 KB
⚡	readme_de	21.01.2022 17:47	Tekstdokument	8 kB
⚡	readme_en	21.01.2022 17:47	Tekstdokument	8 kB
⚡	service	30.03.2013 13:29	Program	60 kB
⚡	setup_xampp	30.03.2013 13:29	Satsvis Windows-fil	2 kB
⚡	test_php	18.12.2019 18:25	Satsvis Windows-fil	2 kB
⚡	uninstall.dat	13.02.2022 22:57	DAT-fil	250 kB
⚡	uninstall	13.02.2022 22:57	Program	12 228 kB
⚡	xampp_shell	13.02.2022 22:53	Satsvis Windows-fil	2 kB
⚡	xampp_start	30.03.2013 13:29	Program	116 kB
⚡	xampp_stop	30.03.2013 13:29	Program	116 kB
⚡	xampp-control	06.04.2021 13:38	Program	3 290 kB



Oppsett av klient

- **Node**

For å starte klienten trenger du **NodeJS 8.9.x** som kan lastes ned [her](#). Last ned **.msi-filen** for enkel oppsett. Åpne terminal og sjekk installering med:

```
$ npm -v
```

5.6.0

Hvis du ikke finner den, prøv å legge til *C:\Program Files\nodejs* til **Path** i **systemvariabler**.

```
C:\Users\dani2>npm -v  
5.6.0
```

- **Angular**

For å laste ned Angular, må vi først få **Angular CLI 1.7.4**. Denne kan lastes ned enten globalt eller lokalt. Åpne en terminal og naviger til **klientmappen**:

Globalt:

```
$ npm install -g @angular/cli@1.7.4
```

Lokalt:

```
$ npm install @angular/cli@1.7.4
```

Sjekk installering med *ng -v*. Nå kan du installere avhengighetene med kommandoen:

```
$ npm install
```

Igjen kan du sjekke med *ng -v*, og du bør også se at **Angular 5.2.7** og **TypeScript 2.5.3** er installert.

```
C:\Users\dani2\AppData\Roai > + <

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os": "darwin", "arch": "any"} (current: {"os": "win32", "arch": "x64"})
  added 1422 packages in 134.939s

C:\xampp\htdocs\stadiontavle\client>ng -v

Angular CLI: 1.7.4
Node: 8.9.4
OS: win32 x64
Angular: 5.2.7
... common, compiler, compiler-cli, core, forms, http
... platform-browser, platform-browser-dynamic, router

@angular/animations: 5.2.8
@angular/cdk: 5.2.4
@angular/cli: 1.7.4
@angular/language-service: 9.1.2
@angular/material: 5.2.4
@angular-devkit/build-optimizer: 0.3.2
@angular-devkit/core: 0.3.2
@angular-devkit/schematics: 0.3.2
@ngtools/json-schema: 1.2.0
@ngtools/webpack: 1.10.2
@schematics/angular: 0.3.2
@schematics/package-update: 0.3.2
typescript: 2.5.3
webpack: 3.11.0

C:\xampp\htdocs\stadiontavle\client>
```

● Start

Du skal nå kunne starte klienten med kommandoen

```
$ npm start
```

Åpne nettleseren din og naviger til **localhost:4200**.

```
C:\xampp\htdocs\stadiontavle\client>npm start
> fantasy-football@0.0.0 start C:\xampp\htdocs\stadiontavle\client
> ng serve

** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

Oppsett av server

- **php**

Å starte serveren krever **php 7.4.x** og bør allerede være installert hvis du lastet ned **xampp**. Du kan sjekke om du har riktig php-versjon ved å kjøre:

```
$ php -v
```

PHP 7.4.26

Hvis du ikke finner den, prøv å legge til *C:\full\path\to\xampp\php* til **Path** i **systemvariabler**.

```
C:\xampp\htdocs\stadiontavle\server>php -v
PHP 7.4.27 (cli) (built: Dec 14 2021 19:52:13) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
```

- **Composer**

For å installere serveravhengighetene trenger du **Composer** som kan lastes ned [her](#).

Etter installasjonen, naviger til servermappen og kjør kommandoen:

```
$ composer install
```

Bekreft at riktig **Laravel**-versjon ble installert ved å kjøre:

```
$ php artisan -v
```

Laravel Framework 5.6.40

```
C:\xampp\htdocs\stadiontavle\server>php artisan -v
Laravel Framework 5.6.40
```

- Start

Før du starter serveren, finn **.env-filen** og rediger linjene med **DB-spesifikasjonene** dine. Hvis du bruker **xampp** og ikke har endret noen konfigurasjoner, skal linjene se slik ut:

DB_CONNECTION=mysql

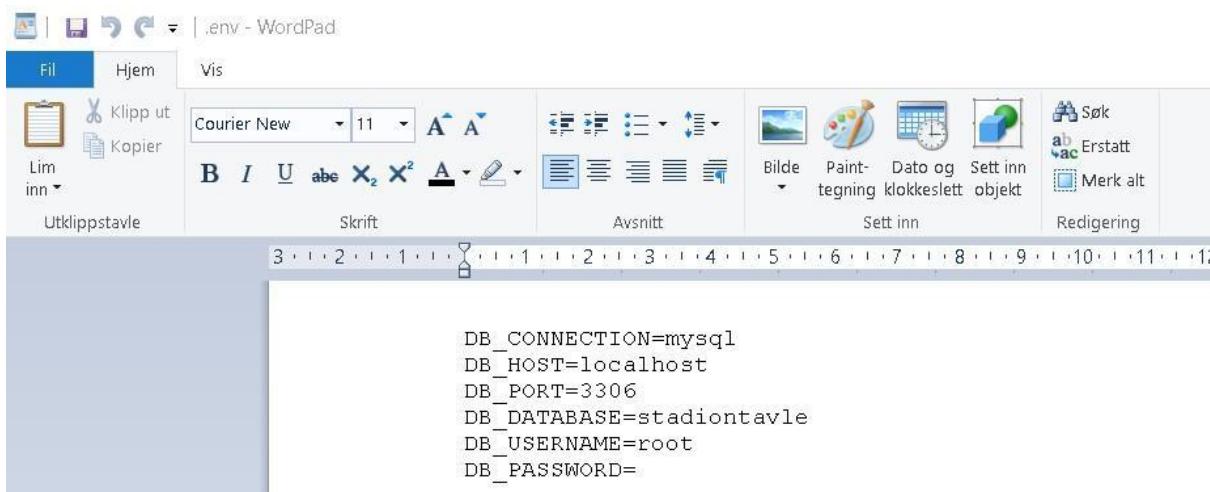
DB_HOST=localhost

DB_PORT=3306

DB_DATABASE=stadiontavle

DB_USERNAME=root

DB_PASSWORD=



Du kan nå starte serveren ved å kjøre kommandoen:

`$ php artisan serve`

Merk at endringer i **.env-filen** krever at serveren starter på nytt.

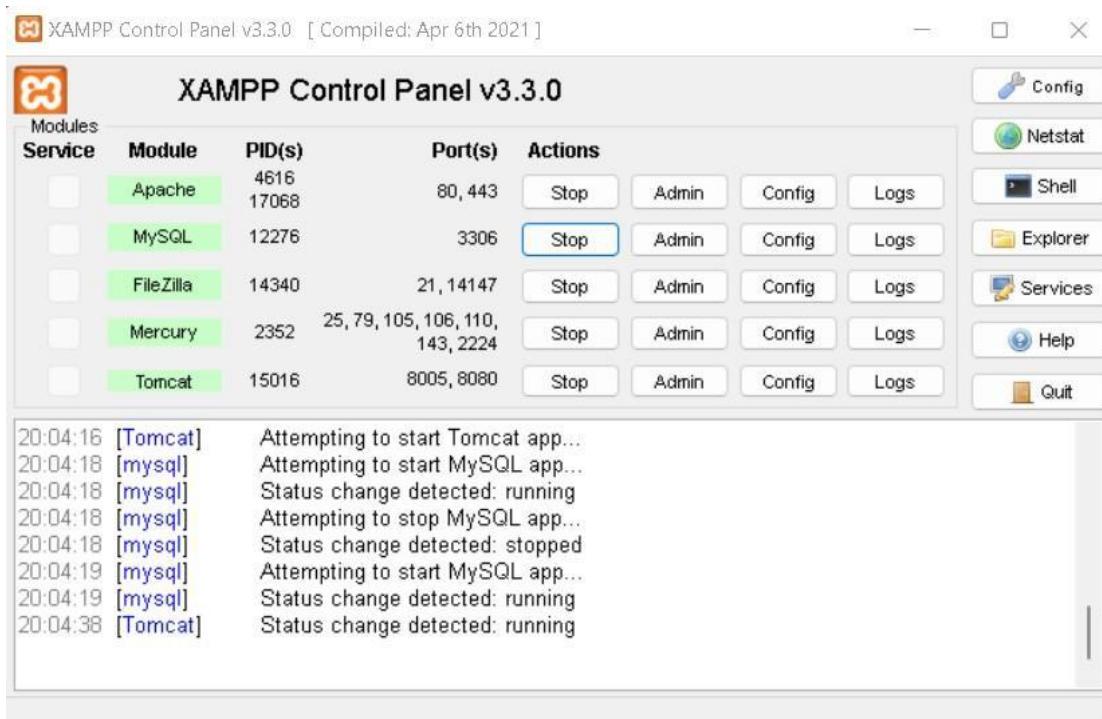
```
C:\xampp\htdocs\stadiontavle\server>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
[Fri May 20 20:10:03 2022] PHP 7.4.27 Development Server (http://127.0.0.1:8000) started
```

Oppsett av database

Start databaseserveren og lag en ny database. Importer **.sql-filen** fra servermappen.

Hvis du bruker **xampp**, åpner du kontrollpanelet og klikker på **Start** på MySQL-linjen.

Åpne nettleseren din og nавигer til localhost/phpmyadmin/



På bildet ovenfor kan du se hvordanxampp ser ut, når man har MySQL, Apache, og de andre modulene på.

Import til gjeldende tjener

File to import:

Filen kan bli pakket (gzip, bzip2, zip) eller utpakket.
En komprimert fils navn må ende med **[format].[compression]**. Eksempel: **.sql.zip**

Bla gjennom datamaskinen: (Maksimum størrelse: 40MiB)

You may also drag and drop a file on any page.

Filens tegnsett:

Klikk på "importer", og deretter velger du "Velg fil"-knappen for å importere sql-filen.

phpMyAdmin

Tjener: 127.0.0.1 » Database: stadiontavle

Struktur SQL Søk Spørring Eksporter

✓ Importen er fullført, 549 spørninger utført. (04042022stadiodb_backup.sql)

Importen er fullført.

phpMyAdmin

Tjener: 127.0.0.1 » Database: stadiontavle

Struktur SQL Søk Spørring Eksporter Importer Operasjoner Privilegier Rutiner Hendelser

Filters

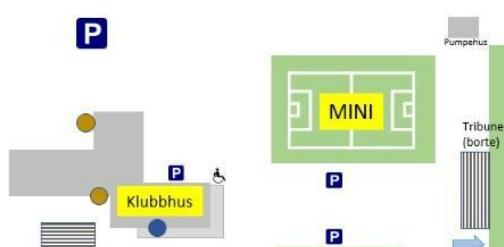
Containing the word:

Tabell	Handling	Rader	Type	Sammenligning	Størrelse	Overheng
articles	Se på Struktur Søk Sett inn Tøm Slett	29	InnoDB	utf8mb4_unicode_ci	176,0 KiB	-
chips	Se på Struktur Søk Sett inn Tøm Slett	471	InnoDB	utf8mb4_general_ci	80,0 KiB	-
clubs	Se på Struktur Søk Sett inn Tøm Slett	10	InnoDB	utf8_general_ci	16,0 KiB	-
invite_participants	Se på Struktur Søk Sett inn Tøm Slett	8	InnoDB	utf8_general_ci	16,0 KiB	-
leagues	Se på Struktur Søk Sett inn Tøm Slett	28	InnoDB	utf8mb4_unicode_ci	16,0 KiB	-
league_points_matrix	Se på Struktur Søk Sett inn Tøm Slett	528	InnoDB	utf8_general_ci	48,0 KiB	-
live_squads	Se på Struktur Søk Sett inn Tøm Slett	5.602	InnoDB	utf8mb4_unicode_ci	1,5 MiB	-
live_squad_rounds	Se på Struktur Søk Sett inn Tøm Slett	50.552	InnoDB	utf8mb4_unicode_ci	8,0 MiB	-
log	Se på Struktur Søk Sett inn Tøm Slett	10.671	InnoDB	utf8_general_ci	1,5 MiB	-
matches	Se på Struktur Søk Sett inn Tøm Slett	2.159	InnoDB	utf8mb4_unicode_ci	384,0 KiB	-

Her er hele databasen for Stadiontavle.

digimakeras/stadiontavle localhost localhost:4200

Nå har du fått opp
Stadiontavle på din
lokale maskin.



Oppsett for Mac OSx

Nyttige programmer

- php 7.4.x
- Github Desktop

Oppsett av klient

● Node

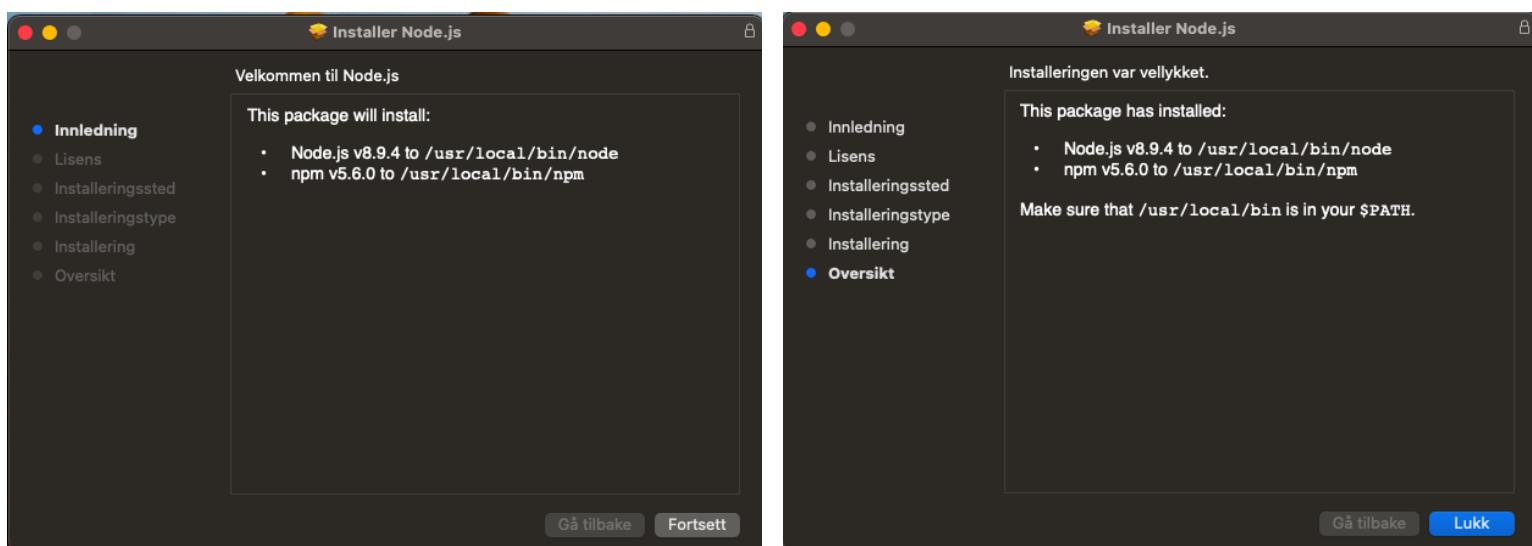
For å starte klienten trenger du **NodeJS 8.9.x** som kan lastes ned [her](#). Åpne terminal og sjekk installering med:

```
$ npm -v
```

5.6.0

Hvis du ikke finner den, kan du prøve å redigere `~/.bashrc`-filen og legge til:

```
PATH=/usr/bin/node:$PATH
```



```
[Laxsans-Air:~ kisan$ npm -v
5.6.0
```

- **Angular**

For å laste ned Angular, må vi først få **Angular CLI 1.7.4**. Denne kan lastes ned enten globalt eller lokalt. Åpne en terminal og naviger til **klientmappen**:

Globalt:

```
$ npm install -g @angular/cli@1.7.4
```

Lokalt:

```
$ npm install @angular/cli@1.7.4
```

Sjekk installering med *ng -v*. Nå kan du installere avhengighetene med kommandoen:

```
$ npm install
```

Igjen kan du sjekke med *ng -v*, og du bør også se at **Angular 5.2.7** og **TypeScript 2.5.3** er installert.

```
kisan@Laxsans-Air client % ng -v
Angular CLI: 1.7.4
Node: 8.9.4
OS: darwin x64
Angular: 5.2.7
... common, compiler, compiler-cli, core, forms, http
... platform-browser, platform-browser-dynamic, router

@angular/animations: 5.2.8
@angular/cdk: 5.2.4
@angular/cli: 1.7.4
@angular/language-service: 9.1.2
@angular/material: 5.2.4
@angular-devkit/build-optimizer: 0.3.2
@angular-devkit/core: 0.3.2
@angular-devkit/schematics: 0.3.2
@ngtools/json-schema: 1.2.0
@ngtools/webpack: 1.10.2
@schematics/angular: 0.3.2
@schematics/package-update: 0.3.2
typescript: 2.5.3
webpack: 3.11.0
kisan@Laxsans-Air client %
```

- **Start**

Du skal nå kunne starte klienten med kommandoen

```
$ npm start
```

Åpne nettleseren din og naviger til **localhost:4200**.

Oppsett av server

- **php**

Å starte serveren krever **php 7.4.x**. Du kan laste ned riktig versjon ved å skrive

```
$ brew install php@7.4
```

i terminalen.

Du kan sjekke om du har riktig php-versjon ved å kjøre:

```
$ php -v
```

PHP 7.4.26

```
kisan@Laxsans-Air ~ % php -v
PHP 7.4.29 (cli) (built: Apr 14 2022 11:48:33) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.29, Copyright (c), by Zend Technologies
```

- **Composer**

For å installere serveravhengighetene trenger du **Composer** som kan lastes ned [her](#).

Etter installasjonen, naviger til servermappen og kjør kommandoen:

```
$ composer install
```

```
kisan@Laxsans-Air server % php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
kisan@Laxsans-Air server % php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01
d4ba75c0ee0459970c2200a51f492d557530c71c15d8dba01eae') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"

Installer verified
kisan@Laxsans-Air server % php composer-setup.php

All settings correct for using Composer
Downloading...

Composer (version 2.3.5) successfully installed to: /Users/kisan/Documents/GitHub/stadiontavle/server/composer.phar
Use it: php composer.phar

kisan@Laxsans-Air server % php -r "unlink('composer-setup.php');"
kisan@Laxsans-Air server % sudo mv composer.phar /usr/local/bin/composer
```



```
[kisan@Laxsans-Air server % composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Package operations: 96 installs, 0 updates, 0 removals
```

Bekreft at riktig **Laravel**-versjon ble installert ved å kjøre:

```
$ php artisan -v
Laravel Framework 5.6.40
```

```
[kisan@Laxsans-Air server % php artisan -v
Laravel Framework 5.6.40
```

- **Start**

Før du starter serveren, finn **.env-filen** og rediger linjene med **DB-spesifikasjonene** dine. Linjene skal se slik ut:

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=stadiontavle
DB_USERNAME=root
DB_PASSWORD=
```

Du kan nå starte serveren ved å kjøre kommandoen:

```
$ php artisan serve
```

Merk at endringer i **.env-filen** krever at serveren starter på nytt.

Oppsett av database

Start databaseserveren i terminal og lag en ny database. Importer **.sql-filen** fra servermappen. Deretter kan du se SQL-dataene i terminal.

Se delkapittel [4.5.1 Filstruktur og database](#), for gjennomgang av oppsett av database.

8.4 Feilmelding

```
** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
Date: 2022-05-20T17:52:27.325Z
Hash: 187298663ea9f7af0d5f
Time: 54972ms
chunk {inline} inline.bundle.js (inline) 3.85 kB [entry] [rendered]
chunk {main} main.bundle.js (main) 2.39 MB [initial] [rendered]
chunk {polyfills} polyfills.bundle.js (polyfills) 556 kB [initial] [rendered]
chunk {scripts} scripts.bundle.js (scripts) 622 kB [initial] [rendered]
chunk {styles} styles.bundle.js (styles) 909 kB [initial] [rendered]
chunk {vendor} vendor.bundle.js (vendor) 26.1 MB [initial] [rendered]

ERROR in multi ./src/styles.scss ./node_modules/dragula/dist/dragula.css ./node_modules/ngx-toastr/toastr.css ./node_modules/froala-editor/css/froala_editor.pkgd.min.css ./node_modules/froala-editor/css/froala_style.min.css ./node_modules/font-awesome/css/font-awesome.css ./node_modules/ngx-bootstrap/datepicker/bs-datepicker.css ./node_modules/kamp-visning-component.css
Module not found: Error: Can't resolve 'C:\xampp\htdocs\stadiontavle\client\node_modules\kamp-visning-component.css' in 'C:\xampp\htdocs\stadiontavle\client'

webpack: Failed to compile.
```

Når man skal kjøre client, kan det hende at man får opp en feilmelding som vist på skjermbildet ovenfor.

Navn	Endringsdato	Type	Størrelse
node_modules	20.05.2022 19:48	Filmappe	
src	20.05.2022 19:15	Filmappe	
.angular-cli	20.05.2022 19:15	JSON File	2 kB
.editorconfig	20.05.2022 19:14	Editor Config Sour...	1 kB
.gitignore	20.05.2022 19:14	Git Ignore Source ...	1 kB
favicon	20.05.2022 19:14	Ikon	6 kB
favicon-16x16	20.05.2022 19:14	PNG-fil	1 kB
favicon-32x32	20.05.2022 19:14	PNG-fil	3 kB
kamp-visning.component	20.05.2022 19:15	CSS-dokument (Ca...	2 kB

En quick-fix vi har for denne feilmeldingen, er å endre navnet på "kamp-visning.component.css"-filen til "kamp-visning-component.css". Etter det skal man putte filen inn i node_modules-mappen. Hvis ikke dette fungerer, kan man slette node_modules-mappen, og prøve å ta *npm install* i client-mappen i terminal.

```
C:\xampp\htdocs\stadiontavle\client>npm start
> fantasy-football@0.0.0 start C:\xampp\htdocs\stadiontavle\client
> ng serve

** NG Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
Date: 2022-05-20T18:09:17.955Z
Hash: 1d3850b50e56d18119b9
Time: 57293ms
chunk {inline} inline.bundle.js (inline) 3.85 kB [entry] [rendered]
chunk {main} main.bundle.js (main) 2.39 MB [initial] [rendered]
chunk {polyfills} polyfills.bundle.js (polyfills) 556 kB [initial] [rendered]
chunk {scripts} scripts.bundle.js (scripts) 622 kB [initial] [rendered]
chunk {styles} styles.bundle.js (styles) 924 kB [initial] [rendered]
chunk {vendor} vendor.bundle.js (vendor) 26.1 MB [initial] [rendered]

webpack: Compiled successfully.
```

8.5 Fremdriftsplan

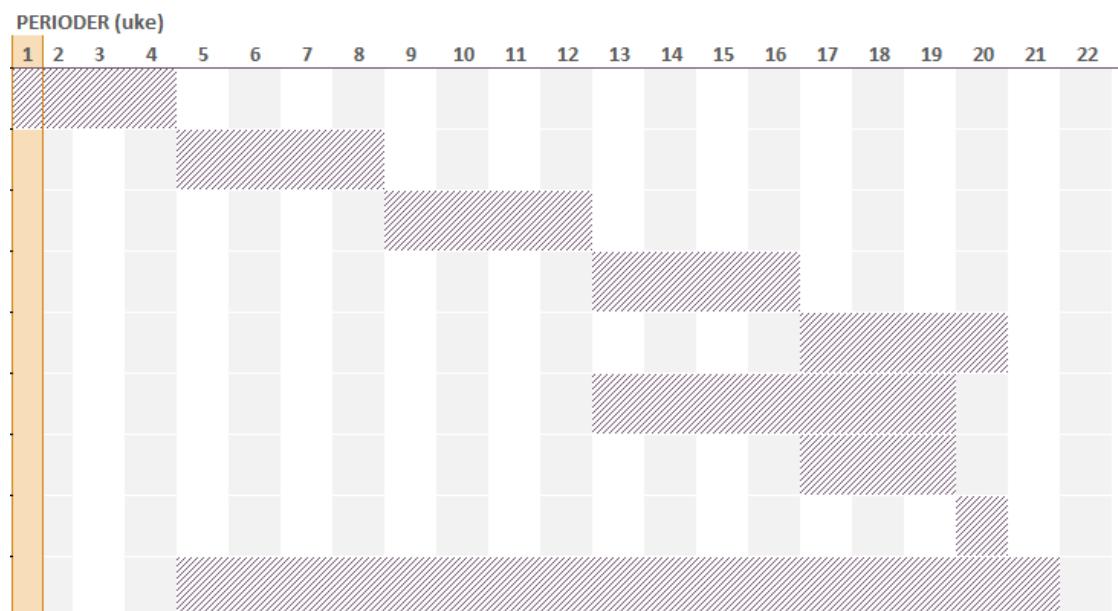
Fremdriftsplan for Sandefjord – Presentasjon av aktiviteter

Periodeutheving:

AKTIVITET	PLANLAGT START	PLANLAGT VARIGHET	FAKTIK START	FAKTIK VARIGHET	PROSENT FULLFØRT
Sprint 1	1	4			
Sprint 2	5	4			
Sprint 3	9	4			
Sprint 4	13	4			
Sprint 5	17	4			
Enhetstesting	13	7			
Systemtest	17	3			
Akseptansetest	20	1			
Sluttrapport	5	17			

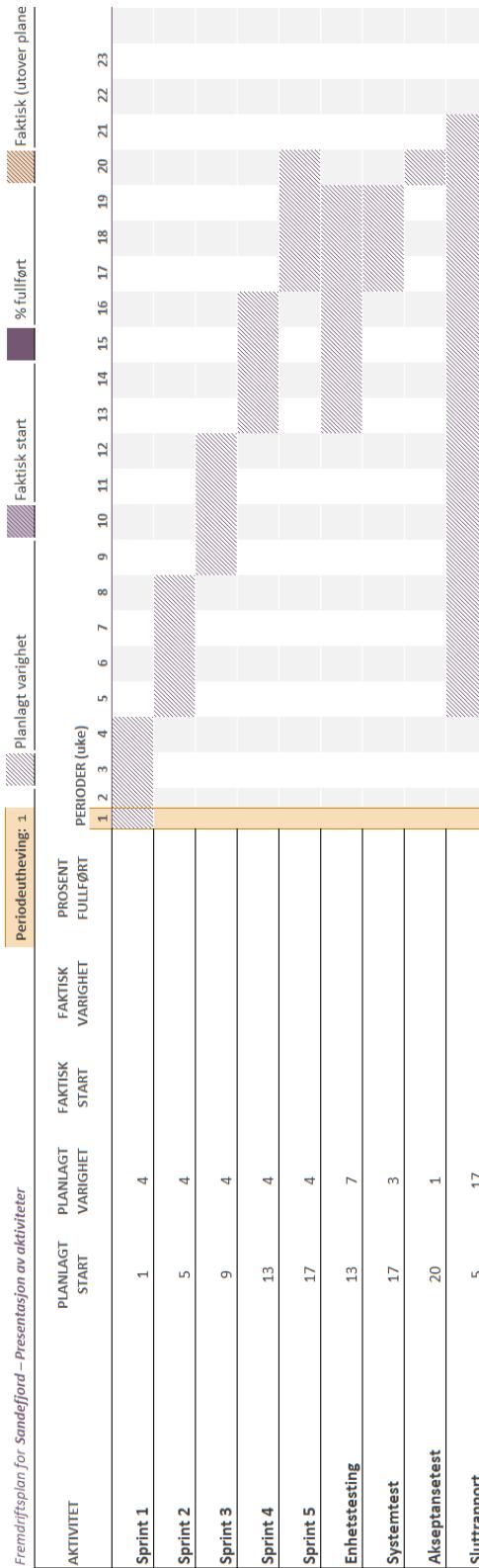
8.5.1 Første del av Gantt - diagrammet (venstre)

: 1 Planlagt varighet Faktisk start % fullført Faktisk (utc)



8.5.2 Andre del av Gantt - diagrammet (høyre)

Prosjektplanlegging



8.5.3 Fullstendig Gantt - diagram

8.6 Kontrakter

AVTALE OM PROSJEKTOPPGAVE

Den 9. desember 2021 ble følgende avtale inngått mellom OsloMet – storbyuniversitetet, med registrert forretningsadresse Pilestredet 46, 0167 OSLO og organisasjonsnr. 997 058 925 (heretter **OsloMet**),

Digimaker as, med registrert forretningsadresse Grenseveien 91 og organisasjonsnr. 913 212 878 (heretter "samarbeidsparten")

Og følgende studenter (heretter "Studentene")

- Dani Tran, med adresse Høybråtenveien 1B, 1055 OSLO og studentnummer s346177
- Kisan Srirangan, med adresse Fjellstuveien 35, 0982 OSLO og studentnummer s345873
- Linda Kadrijaj, med adresse Gjerdrums vei 10B, 0484 OSLO og studentnummer s333751
- Marthe V. Marstrander, med adresse Robert Levins gate 5, 0150 OSLO og studentnummer s333930

1. BAKGRUNN

- 1.1. Studentene er studenter ved OsloMet og skal i forbindelse med sine studieløp forfatte og innlevere en prosjektoppgave innenfor IT (heretter "Prosjektoppgaven"). Prosjektoppgavens arbeidstittel er "Sandefjord – presentasjon av aktiviteter".
- 1.2. Samarbeidsparten driver næringsvirksomhet innenfor Programmeringstjenester og ønsker å samarbeide med OsloMet om tilretteleggingen av Studentenes arbeid med Prosjektoppgaven.

2. VEILEDNING

- 2.1. Studentene skal ha to veiledere, én fra OsloMet og én fra **samarbeidsparten**.
- 2.2. Digimaker vil følge opp og veilede ved behov. Prosjektet vil bli gjennomført etter en smidig arbeidsmetodikk, og studentene vil dermed blir fulgt opp og gitt veiledning regelmessig (normalt 1 gang per uke, men mer ved behov).

3. FINANSIERING MV.

- 3.1. Med mindre annet er avtalt, bærer partene sine egne kostnader i forbindelse med Prosjektoppgaven.
- 3.2. OsloMet skal stille ordinære studiefasiliteter, herunder tilgang til egnede lokaler og datanettverk, til rådighet for Studentene i tråd med OsloMets til enhver tid gjeldende retningslinjer.
- 3.3. Samarbeidsparten skal for egen regning stille til rådighet nødvendige lisenser og utstyr samt lokaler til rådighet i den grad det er nødvendig.

- 3.4.** Samarbeidsparten kan etter avtale dekke Studentenes utgifter til reiser og materiell.

4. LAGRING AV PROSJEKTRELATERTE DATA

- 4.1.** Partene skal hver for seg påse at data av betydning for Prosjektoppgaven lagres på forsvarlig vis.

- 4.2.** Partene er enige om at elektroniske primærdata skal lagres i **samarbeidspartens** systemer.

5. RETTIGHETER TIL RESULTATER MV.

- 5.1.** Studentene har opphavsrett til Prosjektoppgaven. Det samme gjelder alle resultater av arbeidet med Prosjektoppgaven, såfremt disse resultatene er frembrakt av Studentene alene. Rettighetene til alle resultater, metoder, dokumenter, dataprogrammer, prototyper og annet materiale og andre produkter som utarbeides eller utvikles av Studentene sammen med **samarbeidsparten** og OsloMet i forbindelse med arbeidet med Prosjektoppgaven, tilfaller Studentene, samarbeidsparten og OsloMet i fellesskap, med mindre annet er avtalt.

- 5.2.** Samarbeidsparten kan fritt og uten vederlag benytte Prosjektoppgaven og resultatene av arbeidet med Prosjektoppgaven i egen virksomhet.

- 5.3.** OsloMet eier originaleksemplarene av Prosjektoppgaven med vedlegg. OsloMet kan fritt og uten vederlag benytte Prosjektoppgaven og resultatene av arbeidet med Prosjektoppgaven i sin egen virksomhet, herunder i forsknings- og undervisningsøyemed. For publisering av Prosjektoppgaven gjelder bestemmelsene i punkt 7.1-7.3.

- 5.4.** Bestemmelsene ovenfor gjelder bare så langt de er forenlig med ufravikelige regler etter norsk lov, herunder åndsverklovens regler om ideelle rettigheter.

- 5.5.** Hver part har ansvar for å klarere tredjemens eventuelle rettigheter til materiale som bringes frem av vedkommende part som bakgrunnsmateriale for Prosjektoppgaven.

- 5.6.** Studentene har plikt til å melde fra til **samarbeidsparten** og OsloMet om mulig patenterte oppfinnelser som blir frembrakt i forbindelse med arbeidet med Prosjektoppgaven. Hvis en slik oppfinnelse er frembrakt på grunnlag av faglig bistand fra veilederne eller øvrige innsatsfaktorer fra de øvrige partene, skal partenes respektive andeler i rettighetene gjenspeile deres innsats.

6. KONFIDENSIALITET

- 6.1.** **Samarbeidsparten** skal bevare taushet om all informasjon mottatt fra Studentene, og skal ikke formidle slik informasjon til ansatte og andre representanter, personer og enheter enn dem som behøver å kjenne til informasjonen for at rettigheter og plikter etter Avtalen skal kunne utøves. Dette gjelder likevel ikke informasjon som

- (i) er alminnelig kjent
- (ii) en part kan dokumentere at parten kjente til før den ble mottatt fra den annen part
- (iii) en part kan dokumentere at er utviklet av ansatte eller andre som ikke hadde kjennskap til informasjonen mottatt fra den annen part
- (iv) en part kan dokumentere at er blitt gjort tilgjengelig for parten fra en tredjepart som hadde en rett til å frigi denne informasjonen.

Studentene skal bevare taushet om informasjon mottatt fra samarbeidsparten, der samarbeidsparten har informert Studentene om at informasjonen er konfidensiell.

7. PUBLISERING MV.

- 7.1.** Alle forskningsresultater fra arbeidet med Prosjektoppgaven skal være gjenstand for publisering uten restriksjoner, i tråd med universitets- og høgskoleloven § 1-5 (6).
- 7.2.** Publisering kan skje i OsloMets institusjonelle arkiv på internett eller i hvilken som helst publikasjon valgt av Studentene.
- 7.3.** Hvis det er saklig grunn til det, kan partene avtale at offentliggjøring og publisering av hele eller deler av Prosjektoppgaven eller resultatene av arbeidet med Prosjektoppgaven skal utsettes i inntil tre år. Partene skal likevel i størst mulig utstrekning legge til rette for at Studentene kan bruke Prosjektoppgaven i jobbsøknader eller i et senere doktorgradsarbeid.

8. AVTALENS VARIGHET

- 8.1.** Avtalen varer fra Avtaledato, og løper inntil Prosjektoppgaven er ferdig utarbeidet, innlevert og sensurert.
- 8.2.** Hvis Studentene avslutter arbeidet før Prosjektoppgaven er innlevert, kan hver part si opp Avtalen med virkning fra utløpet av inneværende studiesemester.
- 8.3.** Avtalen pkt. 5-7 gjelder videre selv om Avtalen for øvrig er oppfylt eller sagt opp.

9. TVISTELØSNING

- 9.1.** Avtalen er underlagt og skal fortolkes i samsvar med norsk rett.
- 9.2.** Eventuelle tvister mellom partene skal søkes løst ved forhandlinger. Hvis forhandlingene ikke fører frem, kan hver part bringe tvisten inn for de alminnelige domstoler.
- 9.3.** Oslo er avtalt verneting.

Avtalen er undertegnet i 3 eksemplarer, hvorav partene beholder ett eksemplar hver.



Avtalen er inngått på den dato som fremgår innledningsvis i Avtalen.

For OsloMet – storbyuniversitetet

Laurence Habib
Instituttleder,
institutt for
informasjonsteknologi

For samarbeidsparten

Student

Student

Student

Student

Begrepsliste og språk

Frontend	Frontend er det visuelle med en applikasjon.
Backend	Backend, serversiden, fokuserer på den siden av nettstedet brukeren ikke kan se. Altså det som gjør applikasjonen interaktivt.
API/REST-API	Bruk for å samhandle innad i systemet.
Database	Databasen inneholder samling av all relatert data av en applikasjonen.
Cloud	Skytjenester, dataprosessering og datalagring av programvare og informasjon knyttet opp mot internett.
UX	User Experience. Totalopplevelsen en bruker har ved bruk av et produkt, en tjeneste eller et system
Brukergrensesnitt	Samspill mellom menneske og maskin, en form for interaksjon der vi ser fremsiden av en løsning.
CMS	Et publiseringssystem for lagring, håndtering og organisering av innhold på tvers av internett.
CMF	Er et rammeverk som evaluerer og forbedrer en bedrifts muligheter innen informasjonsteknologi. Den er designet for å hente ut større forretningsverdi
eZ Publish	PHP content management system som er utviklet av Ibexa. (Åpen kildekode)

Repository	Repository, også kalt repo, er en katalog eller lagringsplass til et prosjekt. Det kan være lagret på din maskin eller på GitHub.
Branch	Branch lar deg utvikle nye funksjoner, fiks feil eller trygt eksperimentere nye ideer i et avgrenset område i ditt eget repository.
Wordpress	Publiseringssystem for nettsteder skrevet i PHP. WordPress brukes for å administrere innhold som oppdateres ofte f.eks. blogger.
Kjerneprosesser	Kerneprosesser må til for å skape gode kunde verdier. Vanligvis 4 prosesser: Teknologi, struktur, informasjon, mennesker.
Scrum	Smidig rammeverk ofte tilknyttet programvareutvikling, der folk kan adressere komplekse adaptive problemer, samtidig som de produserer produkter av høyest mulig verdi på en produktiv og kreativ måte. Se kap 2.3
Smidig	Med en smidig metode, gjøres planleggingen litt etter litt, og en ny fase i prosessen avgjøres ut i fra hva man oppnådde i forrige fase.
Fossefall	Med en fossefall-modell planlegges hele prosjektet på forhånd, og du vil dele opp til forskjellige oppgaver, hvor hovedoppgaven innen hver aktivitet skal være ferdig og godkjent før en går videre til neste aktivitet.
Inkrementell	Et prosjekt øker i små trinn. Det blir for hver aktivitet/ oppgave lagt til nye funksjoner som gjør at prosjektet gradvis når sin helhet.
Scrum - master	Scrum-master er ansvarlig for at teamet jobber etter de smidige prinsippene innen Scrum. Mellomleddet til produkteier og utviklingsteamet.

CodeCoverage	CodeCoverage er en programvaretesting som bestemmer antall linjer med kode som er vellykket validert under en testprosedyre, som igjen hjelper til med å analysere hvor omfattende programvaren er verifisert ¹⁷ .
--------------	---

¹⁷ (Codegrip, 2020)

Kapittel 9: Kildehenvisning

- [1] Lindsjørn, Y. (2021a). *Introduksjon til Systemutvikling*
- [2] Ramsøy, C. (2021,04.22). *Hva er viktig å huske på når du skal visualisere data?*
<https://www.visma.no/blogg/hva-er-viktig-a-huske-pa-nar-du-skal-visualisere-data>
- [3] Sandnes, F. E. (2018). *Universell utforming av IKT-systemer* (2. utg.) Universitetsforlaget.
- [4] Krattebøl, T. (2020a). *Enhetstesting*
- [6] Scrum (2022). *What is Scrum?*. www.scrum.org
- [7] Lindsjørn, Y. (2021b). *Systemutviklingsprosessen*
- [8] Jain, S. (2019,10.23). *Guidelines & Best Practices for Design RESTful API.*
<https://bytenbit.com/best-guidelines-design-restful-api/>
- [9] Luzniak, K. (2021,12.23). *What is angular used for and when should you use it instead of other frontend technology?*
<https://neoteric.eu/blog/what-is-angular-used-for-and-when-should-you-use-it/>
- [10] Cantero, P. (2019,11.04). *jquery-dateFormat*.
<https://github.com/phstc/jquery-dateFormat>
- [11] Krattebøl, T. (2020a). *Enhetstesting*
- [12] Sommerville, I. (2011). *Software engineering* (9. utg.). New York: Addison-Wesley.
- [13] Krattebøl, T. (2020d). *Testingsteknikker*.
- [14] Angular. (2022,02.28). *Find out how much code you're testing*.
<https://angular.io/guide/testing-code-coverage>
- [15] Krattebøl, T. (2020c). *Systemtesting*.
- [16] Krattebøl, T. (2020d). *Testingsteknikker*.
- [17] Codegrip. (2020,01.13). *Everything you need to know about code coverage*.
<https://www.codegrip.tech/productivity/everything-you-need-to-know-about-code-coverage/>
- [https://www.uia.no/nyskaping/nyheter/hvordan-starte-paa-en-kravspesifikasjon-for-din-it-løsning \(2.1 Innledning\)](https://www.uia.no/nyskaping/nyheter/hvordan-starte-paa-en-kravspesifikasjon-for-din-it-løsning_(2.1_Innledning))

Skissene under [4.2.1 Overordnet arkitektur](#) og [4.2.2 Kommunikasjon](#), ble laget med <https://www.diagrams.net/> (tidligere draw.io).

Skissene under [2.2.2 Analyse og funksjonalitet/design](#), ble laget med Adobe Photoshop 2022.

Bilder:

- Forside
 - OsloMets logo:
<https://ansatt.oslomet.no/documents/585743/81729617/OsloMet+logo+for+nett/755f5bf0-59e1-925a-8aac-e3d8e06f25b4?t=1528291190034> Hentet 07.02.22
 - Stadiontavles logo:
<http://stadiontavle.no/responsivefilemanager/uploads/12/Logo/Stadiontavle%20lightgreen.png> Hentet 07.02.22
 - Digimakers logo:
<https://digimaker.no/static/images/logo.png> Hentet 07.02.22
- [1.5 Presentasjon av veiledere](#)
 - Roza:
https://media-exp1.licdn.com/dms/image/C4D03AQGYfX4_nWASUg/profile-displayphoto-shrink_800_800/0/1647350412341?e=1658361600&v=beta&t=di5cDzXNvxO0rlwlzabjsP0OES2KQLsiZI4znn4ktFl Hentet 01.03.22
 - Christian:
https://media-exp1.licdn.com/dms/image/C5603AQFhe5QJ_Emd8Q/profile-displayphoto-shrink_800_800/0/1516788274413?e=1658361600&v=beta&t=EBwZVifVwifQ5eU-mlzJS7k084GJe-SvKtwaa5f2VMU Hentet 01.03.22
- [3.4 Planning Poker](#)
 - Eksempel på Planning Poker:
<http://www.agilebuddha.com/agile/how-to-do-story-point-sizing-with-planning-poker/> Hentet 17.03.22



- 3.6 Plattform og verktøy

- 3.6.1 Discord:

<https://www.freepnglogos.com/uploads/discord-logo-png/concours-discord-cartes-voeux-fortnite-france-6.png> Hentet 25.04.22

- 3.6.2 Microsoft Teams:

<https://logos-world.net/wp-content/uploads/2021/04/Microsoft-Teams-Logo.png> Hentet 25.04.22

- 3.6.3 GitHub:

<https://logos-world.net/wp-content/uploads/2020/11/GitHub-Logo.png>
Hentet 25.04.22

- 3.6.4 Google Drive:

https://upload.wikimedia.org/wikipedia/commons/6/6a/Google_Drive_text_logo_grey.png Hentet 25.04.22

- 4.3 Teknologier og verktøy

- 4.3.1 Angular:

https://upload.wikimedia.org/wikipedia/commons/thumb/c/cf/Angular_full_color_logo.svg/2048px-Angular_full_color_logo.svg.png Hentet 11.05.22

- 4.3.2 Laravel:

<https://upload.wikimedia.org/wikipedia/commons/thumb/9/9a/Laravel.svg/1200px-Laravel.svg.png> Hentet 11.05.22

- 4.3.3 MySQL:

<https://png.monster/wp-content/uploads/2022/02/png.monster-711.png>
Hentet 11.05.22

- 4.3.4 Visual Studio Code:

https://upload.wikimedia.org/wikipedia/commons/thumb/9/9a/Visual_Studio_Code_1.35_icon.svg/2048px-Visual_Studio_Code_1.35_icon.svg.png

Hentet 11.05.22

- 4.3.5 PhpStorm:

<https://toppng.com/uploads/preview/phpstorm-logo-11609371602rxtjthlyxc.png> Hentet 11.05.22



- 4.3.5 Node JS:

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Node.js_logo.svg/1280px-Node.js_logo.svg.png Hentet 11.05.22

- 4.3.6 Composer:

<https://upload.wikimedia.org/wikipedia/commons/2/26/Logo-composer-transparent.png?20150215175858> Hentet 11.05.22

- 4.3.7 phpMyAdmin:

https://upload.wikimedia.org/wikipedia/commons/4/4f/PhpMyAdmin_logo.svg Hentet 11.05.22