



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Komplett systemløsning og design av en webapplikasjon for orkestersceneoppsett ved bruk av React-Node.js og Firebase integrasjon

Full-stack solution and design of a web application for orchestral stage setup using React-Node.js and Firebase integration

Maksim Ohvrill

Andreas Foss Wergeland

Dataingeniør

Fakultet for ingeniør- og naturvitenskap

Atle Birger Geitung

22.05.2023

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Komplett systemløsning og design av en webapplikasjon for orkestersceneoppsett ved bruk av React-Node.js og Firebase integrasjon	<i>Dato:</i> 22.05.2023
<i>Forfatter(e):</i> Maksim Ohvrill & Andreas Foss Wergeland	<i>Antall sider u/vedlegg:</i> 66
	<i>Antall sider vedlegg:</i> 13
<i>Studieretning:</i> Dataingeniør	
<i>Kontaktperson ved studieretning:</i> Atle Birger Geitung	<i>Gradering:</i>
<i>Merknader:</i> Ingen	

<i>Oppdragsgiver:</i> Styreportalen AS	<i>Oppdragsgivers referanse:</i> Epost: post@styreportalen.no
<i>Oppdragsgivers kontaktperson:</i> Stian Sømoe	<i>Telefon:</i> +47 48 43 80 80

<i>Sammendrag:</i> <p>Dette prosjektet hadde som mål å modernisere en eksisterende applikasjon for utforming av sceneoppsett, spesielt rettet mot korps, ved å konstruere en ny løsning. Ved hjelp av utviklingsverktøy slik som React, Node.js og Firebase plattformen, ble den nye applikasjonen designet og utviklet fra bunnen av og deretter overlevert til oppdragsgiver. Basert på brukertesting og oppdragsgivers tilbakemeldinger, ble det konkludert at prosjektet hadde nådd målene, til tross for at enkelte deler av prosjektet ble utelatt.</p> <p>The objective of this project was to modernize an existing application for the design of stage setups, specifically tailored towards orchestras, by constructing a new solution. With the aid of development tools such as React, Node.js, and the Firebase platform, a novel application was designed and developed from the ground up, and then handed over to the client. Based on user testing and feedback from the client, it was concluded that the project had met its objectives, despite certain aspects of the project being omitted.</p>

<i>Stikkord:</i>		
Applikasjonsutvikling	Modernisering	Korps

Forord

Denne rapporten gir en detaljert gjennomgang av bachelorprosjektet Sceneoppsett, gitt av oppdragsgiver Styreportalen AS. Arbeidet med prosjektet og tilhørende rapport er utført av Maksim Ohvrill & Andreas Foss Wergeland.

Vi ønsker å rette en stor takk til Stian Sømoe for hans utmerket veiledning og det gode samarbeidet under hele prosjektperioden. Vår takknemlighet går også ut til utviklerne ved Styreportalen AS for verdifullt innspill knyttet til prosjektets struktur og funksjonalitet.

En spesiell takk rettes til vår veileder, Atle Birger Geitung, for hans omfattende og kvalitetsbevisste veiledning i både skriveprosessen og det ingeniørfaglige arbeidet forbundet med prosjektet og dokumentasjon av den.

Til slutt vil vi gjerne uttrykke vår takknemlighet til våre familie og venner for deres nærvær og støtte gjennom dette prosjektet.

Innholdsfortegnelse

1	Innledning.....	1
1.1	Kontekst.....	1
1.2	Motivasjon.....	2
1.3	Prosjekteier.....	3
1.4	Problembeskrivelse, mål og forskningsspørsmål.....	3
1.5	Oppbygging av rapporten.....	5
2	Prosjektbeskrivelse.....	6
2.1	Praktisk bakgrunn.....	6
2.1.1	Tidligere arbeid.....	6
2.1.2	Initielle krav.....	7
2.1.3	Initiell løsnings-idé.....	9
2.2	Avgrensninger.....	10
2.3	Ressurser.....	11
2.4	Litteratur om problemstillingen.....	12
2.4.1	Oppgradering og modernisering av systemer.....	12
2.4.2	Kostnads- og ressurseffektivisering.....	12
2.4.3	Arbeidsflyt og brukeropplevelse.....	13
2.4.4	Domenekunnskap.....	14
2.4.5	Iterativ arbeidsprosess.....	14
3	Design av prosjektet.....	15
3.1	Forslag til løsning.....	15
3.1.1	Domenemodell.....	15

3.1.2	Rammeverk og bibliotek	17
3.1.3	Lagring av data.....	19
3.2	Valgt løsning.....	20
3.3	Valg av verktøy	21
3.4	Prosjektmetodikk.....	22
3.4.1	Utviklingsmetodikk.....	22
3.4.2	Fremdriftsplan	23
3.4.3	Risikovurdering.....	24
3.5	Evalueringsplan	27
4	Detaljert løsning	28
4.1	Brukstilfeller.....	28
4.2	Brukergrensesnitt.....	30
4.3	Utvikling.....	32
4.3.1	Komponenter.....	32
4.3.2	Database	33
4.3.3	Kanban	34
4.3.4	Git.....	36
4.3.5	Dokumentasjon.....	37
5	Resultater.....	39
5.1	Evalueringsmetode	39
5.1.1	Iterativ tilbakemeldingsprosess	39
5.1.2	Brukerundersøkelser og brukertesting	40
5.2	Evalueringsresultat	43

5.2.1	Brukerundersøkelser og brukertesting	43
5.2.2	Sluttevaluering av prosjektet	45
5.3	Prosjektresultat	46
5.3.1	Design av ferdigstilt applikasjon	46
5.3.2	Funksjonelle og ikke funksjonelle krav	47
5.3.3	Overlevering og dokumentasjon	49
5.4	Prosjektgjennomføring	51
6	Diskusjon	52
6.1	Tekniske valg	52
6.1.1	Valg av verktøy	52
6.1.2	Sikkerhetsproblem	53
6.1.3	Database	53
6.2	Design valg	54
6.2.1	Skalering av applikasjonen	54
6.2.2	Fargevalg	54
6.2.3	Videre design arbeid	54
6.3	Evaluerings	55
6.3.1	Oppdragsgiver	55
6.3.2	Veileder	55
6.3.3	Brukertesting	56
6.4	Forsknings spørsmål	56
6.5	Forbedringspotensialet	58

7	Konklusjon og videre arbeid	59
7.1	Konklusjon.....	59
7.2	Videre arbeid	60
8	Referanser.....	62
9	Illustrasjonslister	65
9.1	Figurliste.....	65
9.2	Tabelliste	66
10	Vedlegg	1
10.1	Visjonsdokument	1
10.2	Prosjekthåndbok	1
10.3	Kravdokument	1
10.4	Systemdokumentasjon	1
10.5	GitHub lagringssted for prosjektet	1
10.6	Videodemonstrasjon av applikasjonen	2
10.7	Sluttevaluering fra oppdragsgiver	3
10.8	Brukerundersøkelsesskjema	4
10.9	Resultat av brukerundersøkelse og -testing	9

1 Innledning

I dette kapittelet vil fokuset være rettet mot det initielle arbeidet og den nødvendige konteksten som utgjør grunnlaget for prosjektet.

1.1 Kontekst

Teknologisk utvikling er en vedvarende prosess, som har drevet innovasjon og fremgang innen ulike sektorer av næringslivet og samfunnet. Denne utviklingen har sørget for at det nå er mulig å utvikle mer kraftfulle og ressurseffektive løsninger enn tidligere. Noe som legger til rette for et bedre tilbud til kunder og brukere. Dermed har behovet for at bedrifter moderniserer og oppgraderer sine systemer og løsninger økt. For å kunne tilby bedre brukeropplevelse, og oppnå mer kostnadseffektive løsninger gjennom anvendelse av nyere teknologi (Seacord, Plakosh og Lewis, 2003).

Det er viktig å merke seg at teknologisk utvikling ikke bare er begrenset til maskinvare, men legger også stor vekt på programvare. Dette er av stor grad, ettersom det kan være både enklere og billigere å bytte ut programvare i forhold til maskinvare. Nyere programvare gir nye, enklere og mer effektive metoder for utvikling som kan øke produktiviteten betydeligere. Dette fører til, at eksisterende maskinvare kan utnyttes på en mer effektiv måte, der høyere ytelse kan oppnås. Dette gjør det mulig å kjøre program og applikasjoner ved hjelp av færre ressurser, noe som igjen kan redusere kostnader for både bruker og bedrift. Det er fordi eldre maskinvare kan fortsatt brukes og bedriften får reduserte driftskostnader (Patterson og Hennessy, 2021).

I et konkurransedyktig marked er det av avgjørende for bedrifter å være i forkant av teknologiske trender og utnytte mulighetene som følger med disse for å gi kunder og brukere et bedre tilbud og holde i tritt med konkurrentene (Christensen, 2016). En velintegret teknologisk infrastruktur reduserer risikoen for driftsavbrudd og fører følgende til reduksjon i kostnader knyttet feilsøking og feilretting. I tillegg, vil et sømløst integrert system tillate enklere skalering og oppdatering i fremtiden, ettersom bedriften må kunne endre og tilpasse produktene sine etter behov (Laudon og Laudon, 2014). Dette vil også redusere tidsforbruket for å implementere nye funksjoner og løsninger, om behovet skulle oppstå. Som sørger for at bedriftens evne til å raskt reagere og tilpasse seg etter kundens behov øker.

Oppsummert er det avgjørende for bedrifter å investere i modernisering av deres løsninger og produkter. For å sikre en sømløs integrasjon og enklere vedlikehold. Dette vil ikke bare resultere i betydelige kostnadsbesparelser for alle parter, men også bidra til å styrke bedriftens konkurranseevne og bærekraft.

1.2 Motivasjon

Motivasjonen bak dette prosjektet ligger i oppdragsgiverens ønske om å oppgradere og utvide deres eksisterende applikasjon, som skal benytte seg av nyere teknologiske løsninger. Det er ønskelig at oppgraderingen skal resultere i en mer brukervennlig og kostnadseffektiv løsning sammenliknet med dagens versjon. Dette vil kunne øke kundetilfredshet gjennom en bedre arbeidsflyt, og generelt sett bedre brukeropplevelse ved bruk av applikasjonen.

I rapporten analyseres og dokumenteres utviklingsprosessen, med visjon om å vurdere hvordan innføring av annet teknologi kan bidra til å forbedre og utvide eksisterende funksjonalitet til applikasjonen. Et vellykket prosjekt for oppdragsgiver er viktig, da det vil gi et bedre tilbud til deres brukere og redusere driftskostnader knyttet til applikasjonen. Videre vil rapporten belyse hvordan overgang til nyere teknologi muliggjør bedre utnyttelse av maskinvare, slik at applikasjonen kan brukes på det aller fleste enheter. Dette understreker viktigheten av dette prosjektet, både for oppdragsgiver og for andre bedrifter som vurderer liknende oppgraderinger for sine systemer.

Sammenfattet ligger prosjektets motivasjon i å undersøke hvordan en oppgradering av en eldre løsning, gjennom en nyere teknologisk tilnærming, kan resultere i en mer sømløs løsning som enkelt kan enkelt integreres i oppdragsgivers systemer. Dette kan potensielt redusere kostnader knyttet til drift av applikasjonen, og gi deres kunder et bedre tilbud. Det er også potensial for at et vellykket og fullført prosjekt, fører til at applikasjonen erstatter dagens løsning og aktivt tas i bruk.

1.3 Prosjekteier

Styreportalen AS er eier og oppdragsgiver for prosjektet. Kildekoden skal ikke offentliggjøres under utviklingsprosessen på noe tidspunkt. Dersom løsningen viser seg å være tilstrekkelig, integreres den som en del av Styreportalens undersystem, Program.no. Dette undersystemet er hovedsakelig utviklet for Norges Musikkorps Forbund (NMF), men er også tilgjengelig for andre organisasjoner som er kunder av Styreportalen. Undersystemet tilbyr en rekke tjenester, inkludert fremstilling av kart over kommende arrangement, verktøy for dommere ved arrangementer, samt applikasjonen Sceneoppsett, som rapporten fokuserer på.

Styreportalen AS ble etablert i 2016 (Proff.no, 2023) og fokuserer på å tilby et utvalg av verktøy og løsninger for å forenkle arbeidsprosesser og drift, primært for korps og musikkorganisasjoner, men også for lag og foreninger. Løsningene de tilbyr kan potensielt bli brukt av flere tusen brukere på dager med mest pågang. Dette er illustrert av Norsk mesterskap (NM) i brass i Bergen, der 6000 unike brukere gjennomførte 30 000 nedlastninger under et enkelt arrangement. Denne statistikken er hentet fra Styreportalens egne data, fremvist under et møte.

Styreportalen har etablert et godt samarbeid med Høgskolen på Vestlandet (HVL) og har tidligere erfaring med veiledning av bacheloroppgaver. Intern veileder fra HVL er også kjent med Styreportalen og deres arbeidsmetoder. Dette samspillet mellom Styreportalen og HVL vil bidra positivt til prosjektets gjennomføring og skal styrke Styreportalen sin posisjon som prosjekteier. Dette samspillet kan være med på å skape et solid fundament for en vellykket gjennomføring og dokumentering av prosjektet.

1.4 Problembeskrivelse, mål og forskningsspørsmål

Styreportalen har som mål å oppgradere sine utdaterte system, for å benytte seg av nyere og bedre tilpasset teknologi. Dette innebærer at et prosjekt ble satt i gang for å oppdatere en av deres eldre løsninger til mer moderne teknologi. Løsningen som er i fokus, er en applikasjon som er en del av en plattform der brukere, hovedsakelig assosiert med korps, kan utforme sceneoppsett for konkurranser. Til konkurransen kan arrangøren finne frem hvert enkelt oppsett og klargjøre scenen for hvert korps.

Dagens løsning fører foreløpig til dårlig brukeropplevelse og økte kostnader. Årsaken til dette er at andre deler av Styreportalens systemer benytter nyere teknologi, noe som medfører ekstra arbeid knyttet til drift og vedlikehold av den utdaterte applikasjonen. Styreportalen erkjenner problematikken og ønsker å adressere dette ved å bygge opp applikasjonen på nytt, samtidig som dagens funksjonalitet beholdes og utvides ytterlig i henhold til kunder og brukeres behov. En mer detaljert beskrivelse av problemet med dagens løsning og visjonen for prosjektet finnes i vedlegg 10.1 (Visjonsdokument).

Formålet med prosjektet er å gjennomføre og dokumentere oppgraderingen av applikasjonen Sceneoppsett, som er presentert i denne rapporten. Målet er å undersøke hvordan oppgradering og utvidelse av eksisterende versjon kan øke kundetilfredshet og redusere drifts og vedlikeholdskostnader ved applikasjonen. Forskningsspørsmålet som rapporten fokuserer på, er derfor følgende:

Hvordan kan oppgradering og utvidelse av Styreportalens eksisterende applikasjon Sceneoppsett forbedre brukeropplevelsen, redusere drifts og vedlikeholdskostnadens samt øke kundetilfredsheten ved å anvende en ny teknologisk tilnærming?

For å gjennomføre prosjektet ble det ikke gitt kildekode til den nåværende versjon av applikasjonen, ettersom det skal benyttes annet type teknologi som gjør det irrelevant for prosjektets gang. Tilgang til selve applikasjonen ble gitt, og tatt som utgangspunkt for utviklingen av både design og funksjoner. Den nye løsningen skal utvikles ved bruk av nyere verktøy og programmeringsspråk, og etter plan ta i bruk nyere løsning for databaselagring av sceneoppsett for brukerne. Teknologi som er nevnt, er fastsatt som et krav fra oppdragsgiver, for å sikre sømløs integrering med deres andre system. Løsningen for dette vil bli beskrevet i detalj senere i rapporten. Ved prosjektets slutt forventes det å oppnå en ferdigstilt løsning, en fullstendig rapport som dokumenterer utviklingsprosessen, samt utviklingserfaring som kan anvendes i arbeidslivet.

1.5 Oppbygging av rapporten

Rapporten er utformet i samsvar med mal for bachelorrappporter, utarbeidet av HVL. Rapportens hovedstruktur vil i stor grad følge det opprinnelige oppsettet, men med noen justeringer tilpasset dette prosjektet. Inkluderte kapitler er som følger:

Tabell 1-1 Oppbygging av rapport

1. Innledning	Denne delen av rapporten beskriver bakgrunnen for prosjektet, motivasjon, hvilke problemer som skal løses, og målene for produktet. Rapportens struktur blir også presentert.
2. Prosjektbeskrivelse	Her beskrives bakgrunnen for prosjektet, arbeid og tidligere løsning, kunnskap og teori som skal brukes i prosjektet. Det nevnes også avgrensninger, prosjekteierens krav og ideer.
3. Design av prosjektet	Dette kapitlet beskriver teknologier og metoder som skal brukes i forhold til det ingeniørfaglige arbeidet. Valg og begrunnelser av løsning, teknologi og utviklingsprosess blir også beskrevet her. Planen for evaluering av resultatene skal også bli presentert her.
4. Detaljert løsning	I dette kapitlet beskrives løsningen og designet av prosjektet i detalj som bygger videre på valgene tatt i forrige kapitel. Her vil rapporten få frem hvordan resultatene skal faktisk oppnås.
5. Resultater	Denne delen beskriver hvordan resultatene er evaluert og hvilke resultater som er oppnådd. Metoder og fremgangsmåter som er brukt i evalueringen blir også presentert.
6. Diskusjon	Her diskuteres årsakene til at resultatene ble som de ble, spesielt er det satt fokus på avvik i forhold til oppsatte plan og mål. Det skal også vurderes hva som kunne vært gjort annerledes og hvordan dette kunne ha påvirket resultatene.
7. Konklusjon og videre arbeid	I dette kapitlet presenteres konklusjoner i forhold til problemstillingene og kravene som ble satt opp i visjonsdokumentet. Anbefalinger for videre arbeid og bruk av resultater som ble oppnådd.
8. Referanser	Inneholder oversikt over referanser som er brukt i rapporten.
9. Illustrasjonslister	Oversikt over alle figurer og tabeller som er brukt i rapporten.
10. Vedlegg	Her kan leseren finne relevante vedlegg i forhold til prosjektet slik som: kravdokument, visjonsdokument, prosjekthåndbok, brukerundersøkelser i forhold til brukerstesting, og andre relevante vedlegg.

2 Prosjektbeskrivelse

Dette kapitlet fokuserer på den praktiske bakgrunnen for prosjektet, samt begrensningene prosjektet har med hensyn til ressurser. Det dekker også hvilke ressurser prosjektet blir tildelt for utvikling, samt den forskningen som blir utført ved hjelp av relevant litteratur knyttet til problemstillingen.

2.1 Praktisk bakgrunn

I dette kapitlet beskrives det tidligere arbeidet som prosjektet tar utgangspunkt i, samt de initielle kravene og løsningsideen for prosjektet.

2.1.1 Tidligere arbeid

Dette prosjektet tar utgangspunkt i en applikasjon Sceneoppsett som ble utviklet i 2017 ved bruk av Hypertext Preprocessor (PHP) som programmeringsspråk og Cloud Firestore – Firebase for databaselagring med server lokasjon i USA, informasjonen ble gitt i møte med oppdragsgiver. Applikasjonen utgjør per dags dato en del av et større system nevnt tidligere Program.no, som består av to hovedkomponenter. Første komponenten er en kulturkalender der brukere kan publisere arrangementer som konserter og lignende. Den andre komponenten er et administrasjonsverktøy som muliggjør håndtering av festivaler og konkurranser. Innenfor dette administrasjonsverktøyet er applikasjonen Sceneoppsett.

Systemet tilbyr også en rekke funksjoner, inkludert opprettelse av påmeldingsskjemaer, kommunikasjon med påmeldte deltakere via e-post og SMS, oppsett av program, gjennomføring av kollisjonstester for å identifisere potensielle problemer, og publisering av programmet. Videre inneholder systemet en resultattjeneste der dommer eller jurymedlemmer kan registrere resultater. Disse kan dermed sendes til deltagere ved hjelp av e-post eller mobilapplikasjonen Program.no som er også del av dette systemet. Systemet blir brukt av omtrent 100 arrangementer hvert år, fra små lokale arrangementer til nasjonale mesterskap. Denne informasjonen var fremmet under et møte med oppdragsgiver.

Dagens løsning muliggjør konfigurasjon av et sceneoppsett ved å la brukeren dra og plassere instrumenter på en canvas som illustrerer en scene. Brukeren kan kontrollere én dirigent som er utplassert på scenen, legge til et bestemt antall stoler, og legge til en kommentar knyttet til

oppsettet. Etter brukeren er ferdig med å konfigurere oppsettet, kan det lagres til en database og kan sendes videre til et arrangement eller konkurranse. Brukeren kan ha flere oppsett lagret, og kan på et senere tidspunkt hente inn og redigere et gitt oppsett.



Figur 2-1 Design av nåværende applikasjon Sceneoppsett (testside fra Styreportalen)

Utvikling av den nye løsningen tar inspirasjon fra visse aspekter av funksjonalitet og design fra den nåværende løsningen. Oppdragsgiverens eksisterende løsning er fortsatt i bruk og er tilgjengelig for deres kunder. For å støtte utviklingen av den nye løsningen, har det blitt opprettet tilgang til en testside, der det er mulig å teste funksjonalitet på lik linje som en vanlig bruker.

2.1.2 Initielle krav

De initielle kravene for prosjektet er fastsatt som et utgangspunkt for utviklingen. Disse kravene inkluderer både funksjonelle krav, som fokuserer på de spesifikke funksjonene applikasjonen skal introdusere eller forbedre, og ikke-funksjonelle krav, som setter krav til aspekter som brukervennlighet, arkitektur og integritet.

2.1.2.1 Funksjonelle krav

Følgende funksjonelle krav er gitt av oppdragsgiver som et utgangspunkt, med mulighet for at det kan oppstå flere behov under utviklingsperioden. Oppdragsgiver angir også hvilke funksjoner den

nåværende løsningen allerede har. Videre rangeres kravene etter prioritet for implementering, fra lav til middels og høy:

- Lav prioritet: Kravene er ønskelig, men ikke essensielle.
- Middels prioritet: Krav med følgende prioritet er viktig for å oppnå forbedret funksjonalitet, men er ikke kritisk for kjernefunksjonaliteten.
- Høy prioritet: Krav med høy prioritet setter fundament for applikasjonens funksjoner, disse skal prioriteres fremst slik at løsningen skal bli brukbar.

Tabell 2-1 Oversikt over utgangspunkt for funksjonelle krav

Krav	Prioritet	Dagens løsning
Antall stoler	Høy	Ingen
Lagring av sceneoppsett	Høy	Dagens løsning oppfyller dette kravet
Plassering av instrumenter på canvas	Høy	Bruker kan dra og plassere instrumenter på canvas
Kommentar	Høy	Bruker kan legge ved kommentar i inndatafeltet
Dirigentinnstillinger	Høy	Bruker kan vise frem dirigent med notestativ og podium, men uten beskrivelse
Oversikt	Middels	Ingen
Navn på instrument	Middels	Ingen
Oppsettguide	Middels	Ingen
Filopplasting av eget sceneoppsett i form av PDF eller bilde	Middels	Ingen
Ikoner skal være fastbestemt størrelse	Middels	Brukeren kan endre størrelsen på ikoner/instrumenter
Skalering ved endring i skjermstørrelse	Middels	Ingen, for store komponenter blir skjult
Legge til stoler	Middels	Brukeren kan velge antall stoler som blir plassert på scenen automatisk
Beskrivelse av dirigent	Lav	Ingen
Språkstøtte	Lav	Dagens løsning er bare tilgjengelig på norsk
Egendefinerte objekter	Lav	Det er mulig for brukeren å legge inn, men ikke navngi eller tilpasse

Innenfor applikasjonen skal brukeren ha mulighet til å plassere bildelementer av musikkinstrumenter og annet utstyr på canvas, som danner grunnlaget for sceneoppsettet. Instrumenter skal identifiseres ved navn, både i valgmenyen og når det er plasser på canvas ved at bruker holder muspekeren over instrumentet. Det ble fastsatt krav om at det ikke skal være mulig

å endre størrelsen på instrumentene, noe som er et problem med dagens løsning. Videre skal applikasjonen følge et bestemt skaleringsmønster ved endring i vindusstørrelsen, hvor alle komponentene ved applikasjonen skaleres inntil en viss størrelse på vinduet er nådd.

Det ble også stilt spesielle krav i forhold til dirigenten, som skal kunne vises med eller uten notestativ og podium, som må også spesifiseres med tekst både når bruker drar over muspekeren og som data for rapport generering. Krav om egendefinerte objekter for instrumenter som ikke er på listen under den vanlige menyen, ble også fremmet.

Brukeren skal kunne legge ved en kommentar til sceneoppsettet. Det ble diskutert om det skulle være støtte for flere språk, men dette kravet ble ikke ansett som prioritet. Dette skyldes at applikasjonen først og fremst er ment for norske korps under NMF.

2.1.2.2 Ikke-funksjonelle krav

Brukergrensesnittet skal utvikles ved hjelp av et deklarativt brukergrensesnittbibliotek som er et krav gitt av oppdragsgiver som blir beskrevet mer i detalj senere i rapporten. Applikasjonen skal benytte en moderne og kostnadseffektiv løsning for datalagring, ved å integrere en tredjepartsløsning for databaselagring som er fastsatt av oppdragsgiver. Dette blir også beskrevet mer i detalj senere i rapporten. Datalagring skal foregå på en sikker måte. Siden det ikke skal håndteres noe form for brukeropplysninger, trenger heller ikke prosjektet å ta hensyn til regelverk knyttet til dette. Løsningen må kunne håndtere innkommende trafikk som er rettet mot oppdragsgiveres brukerbases. Prosjektet vil prioritere å gi applikasjonen en mer brukervennlig og moderne design i forhold til den eksisterende løsningen.

2.1.3 Initiell løsnings-idé

Designets løsning bygger på tidligere design, med mål om å modernisere applikasjonen og forbedre arbeidsflyten. Målet er å undersøke løsninger med lignende konsepter innen tegning og/eller modellering som tar i bruk komponenter og elementer. Dette innebærer å analysere arbeidsflyten i slike applikasjoner, vurdere plasseringen av ulike verktøy og tilgjengeligheten av funksjoner for brukere.

Ved å kombinere relevant kunnskap fra analysen, vil det gi godt grunnlag for utføring av brukertesting og undersøkelser for å identifisere hvilken løsning som kan fungerer best. Dette kan bidra til å sikre en god brukeropplevelse.

Den nye løsningen for applikasjonen vil videreføre funksjonalitet fra den eksisterende løsningen. I tillegg vil den utvide funksjonalitet i tråd med krav fra oppdragsgiver. Den oppgraderte løsningen vil hovedsakelig innebære forbedring av eksisterende funksjoner, samt legge til nye funksjoner som lar brukeren lage et mer oversiktlig og detaljert sceneoppsett.

Oppdragsgiveren anser den nåværende løsningen som tilfredsstillende for deres brukere, dette ble fortalt under et møte. Imidlertid er det et ønske om ytterligere funksjonalitet, mens verdien for prosjektet ligger hovedsakelig i overgangen og oppgraderingen til annen teknologi. Dette vil videre sikre integrasjonen av løsningen med oppdragsgiverens øvrige systemer, som tidligere nevnt i kapittel 1.4 (problembeskrivelse, mål og forskningsspørsmål).

2.2 Avgrensninger

Under det første møtet med oppdragsgiveren ble det presentert krav som er beskrevet i kapittel 2.1.2 (initielle krav). Blant disse kravene er alle i utgangspunktet gjennomførbare. Prosjektet gir muligheten til å begrense omfanget ved å utelate enkelte krav med lav prioritet, dersom det oppstår problemer eller tidsbegrensninger.

Et eksempel på dette er at dersom det ikke er tilstrekkelig med tid til å integrere databaseløsningen, vil det være tilfredsstillende for oppdragsgiveren at lagringsinformasjonen oppgis i et format som de kan arbeide videre med. Hvis prosjektet ikke blir fullført, vil oppdragsgiveren fortsette med utviklingen av applikasjonen. Applikasjonen skal utvikles innenfor oppdragsgiverens utvalgte tekniske infrastruktur, dette begrenser valget av utviklingsverktøy. Imidlertid ble det ikke satt noen begrensninger for hvilke biblioteker innenfor det tekniske infrastrukturen som kunne brukes eller hvilken tilnærming som skulle benyttes for utviklingen.

2.3 Ressurser

I prosjektet vil det bli benyttet flere ressurser for å oppnå tilstrekkelig kunnskap om problemstillingen, oppdragsgiverens krav og utviklingsprosessen. Google Scholar¹ blir brukt for å finne vitenskapelig informasjon relevant for prosjektet. En betydelig del av introduksjonen til utviklingsverktøy blir hentet fra bøker funnet gjennom Google Scholar. Videre benyttes også dokumentasjonssider for programmeringsspråk, utviklingsbibliotek og rammeverk som en annen viktig ressurs i prosjektet.

Det finnes andre applikasjoner som har mange lignende funksjoner som oppdragsgiveren ønsker, men disse er ikke spesifikt tilpasset deres bruksområde. Likevel kan disse applikasjonene være nyttige ressurser for å hente inspirasjon til applikasjonen i prosjektet. Blant de identifiserte applikasjonene i søket etter lignende løsninger ble disse applikasjonene funnet, Draw.io (JGraph_Ltd, 2023), miro.com (Miro, 2023), Lucidchart (Lucid_Software_Inc, 2023), Microsoft Visio (Microsoft, 2023b). Disse applikasjonene brukes til å lage diagrammer og illustrasjoner i bedriftsmiljøer, privat bruk og skolemiljøer. Det er også mulig at det finnes flere lignende løsninger som ikke er nevnt her, men disse har blitt benyttet som inspirasjonskilder under utviklingsprosessen.

Av disse er Draw.io mest interessant, fordi applikasjonen inneholder mange av de ønskede funksjonene i forhold til oppdragsgiverens krav. Draw.io gir brukere muligheten til å plassere figurer, tegne og legge inn tekst, og det brukes hovedsakelig til modellering av ulike typer diagrammer. Imidlertid, når det gjelder modellering av sceneoppsett for et korps, ville det være upraktisk for brukerne å måtte laste opp egne figurer for instrumenter og deretter modellere oppsettet på denne måten.

Det finnes flere programmer som fungerer på samme måte som Draw.io, og basert på analyse av disse programmene har de vist seg å være effektive i bruk. På grunn av deres vellykkede design vil oppgavens layout bli inspirert av disse programmene for å skape en brukervennlig og oversiktlig opplevelse for brukere.

¹ Nettbasert søkemotor for bøker, artikler og andre akademiske ressurser

Oppdragsgiveren har ikke tilført ytterligere ressurser enn tilgang til testnettside for deres eksisterende applikasjon og tilgang til den nye databaseløsningen for testing av den nye applikasjonen. Utviklingen blir gjennomført ved bruk av eget utstyr.

2.4 Litteratur om problemstillingen

For å få innsikt i hvordan problemstillingen skulle løses og hvilken metodikk som skulle benyttes for utviklingsprosessen og dokumentasjon av den, ble det utført en grundig undersøkelse og analyse av relevant faglitteratur som beskrives i underkapitlene. Litteratur var i hovedsak funnet ved hjelp av Google Scholar og IEEE Xplore².

2.4.1 Oppgradering og modernisering av systemer

For å oppnå en dypere forståelse av tilnærmingen til eldre systemet og få innsikt i deres oppbygning og funksjonalitet gjennom bruk av reverse engineering³ (omvendt utvikling) (*Seacord, Plakosh og Lewis, 2003*) benyttet. Denne boken introduserer til konsepter og praksis bak omvendt utvikling, samt vektlegger modernisering av eldre løsninger og hvordan dette fører til enklere integrering og vedlikehold.

Videre ble kunnskapen om omvendt utvikling supplert av artikkelen (*Perez-Castillo et al., 2011*), som fokuserer på å fremme engasjement hos bedrifter med å oppgradere deres løsninger. Artikkelen presenterer verktøy, prinsipper og metoder for omvendt utvikling, restructuring⁴ (omstrukturering) og forward engineering⁵ (fremadrettet utvikling). Denne informasjonen var spesielt relevant ettersom kildekoden til den eksisterende løsningen ikke ble gitt, men i stedet ble det gitt en testside som utgangspunkt.

2.4.2 Kostnads- og ressurseffektivisering

I situasjoner hvor eldre systemer krever betydelige økonomiske ressurser for vedlikehold, kan det noen ganger vise seg å være mer kostnadseffektivt å oppdatere systemet i stedet for å vedlikeholde

² Nettbasert søkemotor for bøker, artikler og andre akademiske ressurser

³ Prosess for å forstå noe uten å vite hvordan der bygd opp eller fungerer

⁴ Prosess som involverer forbedring av struktur og kvalitet samt innføring av nye krav

⁵ Utvikle et system fra kravspesifikasjon fra f.eks. omstrukturering til et ferdig produkt

det eksisterende over lengre tid. Dette skyldes ofte at organisasjonen ikke lenger har personell som behersker programmeringsspråket det opprinnelige systemet ble utviklet i. Det kan derfor være økonomisk gunstig å konvertere til et nytt programmeringsspråk som det nåværende personellet har kompetanse innen. (Seacord, Plakosh og Lewis, 2003) diskuterer hvordan overgangen til et nyere system kan bidra til å redusere drifts- og vedlikeholdskostnader.

2.4.3 Arbeidsflyt og brukeropplevelse

Designet av applikasjonen er inspirert av prinsippene fremhevet i (Krug, 2014). Mange av disse prinsippene har vært nyttige for å beholde applikasjonens enkelhet og estetikk. Hovedargumentet i boken er at brukere ikke bør overveldes av sideinnholdet, og at et effektivt brukergrensesnitt kan betydelig forbedre arbeidsflyten. Brukervennlighet er essensiell for å sikre at brukere enkelt kan utføre deres oppgaver.

Universal Design er også en sentral overveielse i utformingen av applikasjonen, med (Goldsmith og PRP Architects (Firm), 2000) som gir relevant innsikt i dette konseptet. To tilnærminger til universal design er verdt å merke seg, top-down og bottom-up. Top-down tilnærmingen fokuserer på å utvikle produkter tilgjengelige for personer med nedsatt funksjonsevne, som også er brukervennlige for personer uten nedsatt funksjonsevne. Bottom-up tilnærmingen, derimot, vurderer produkter utformet på en slik måte at produktet til slutt er brukervennlige også for personer med nedsatt funksjonsevne. Et eksempel på dette er automatgir i biler som er en funksjon som forenkler bruken for alle, inkludert personer med nedsatt funksjonsevne.

Til slutt fremmer (Lidwell, Holden og Butler, 2003) en metode for design som tar for seg fem kritiske spørsmål:

1. Hvordan kan designet påvirke oppfattelsen?
2. Hvordan kan brukere lære fra et design?
3. Hvordan kan brukervennligheten av et design forbedres?
4. Hvordan kan et designs appell økes?
5. Hvordan kan bedre designbeslutninger tas?

Denne boken inneholder 125 forskjellige designprinsipper som er relevante for alle typer designarbeid.

2.4.4 Domenekunnskap

Det er essensielt å anskaffe nødvendig forståelse av domeneområdet til produktet som skal utvikles, for å indentifisere områder som må håndteres og hvilke oppgaver som skal løses. Boken (Marinescu og Avram, 2007) introduserer prinsippene bak metode Domain-Driven Design (DDD) også kjent som domenedrevet design, som tilbyr en rekke verktøy for å oppnå bedre forståelse av domenet det jobbes med. For modellering av domene benyttes informasjon fra artikkel (Specification, 2007), som presenterer Unified Modeling Language (UML) med eksempler og brukstilfeller, med særlig vekt på domenemodellering.

2.4.5 Iterativ arbeidsprosess

Prosjektet gjennomføres som iterativ prosess som nevnt i (Larman, 2004). Iterativ arbeidsprosess handler om at utviklingen av applikasjonen gjøres ett steg om gangen. Når et steg er ferdig, kan applikasjonen testes og deretter gå videre til neste steg. Dermed blir applikasjonen kjørbare veldig tidlig i utviklingen, og med videre utvikling kan applikasjonen utvides. Slik vil applikasjonen vokse med tiden.

3 Design av prosjektet

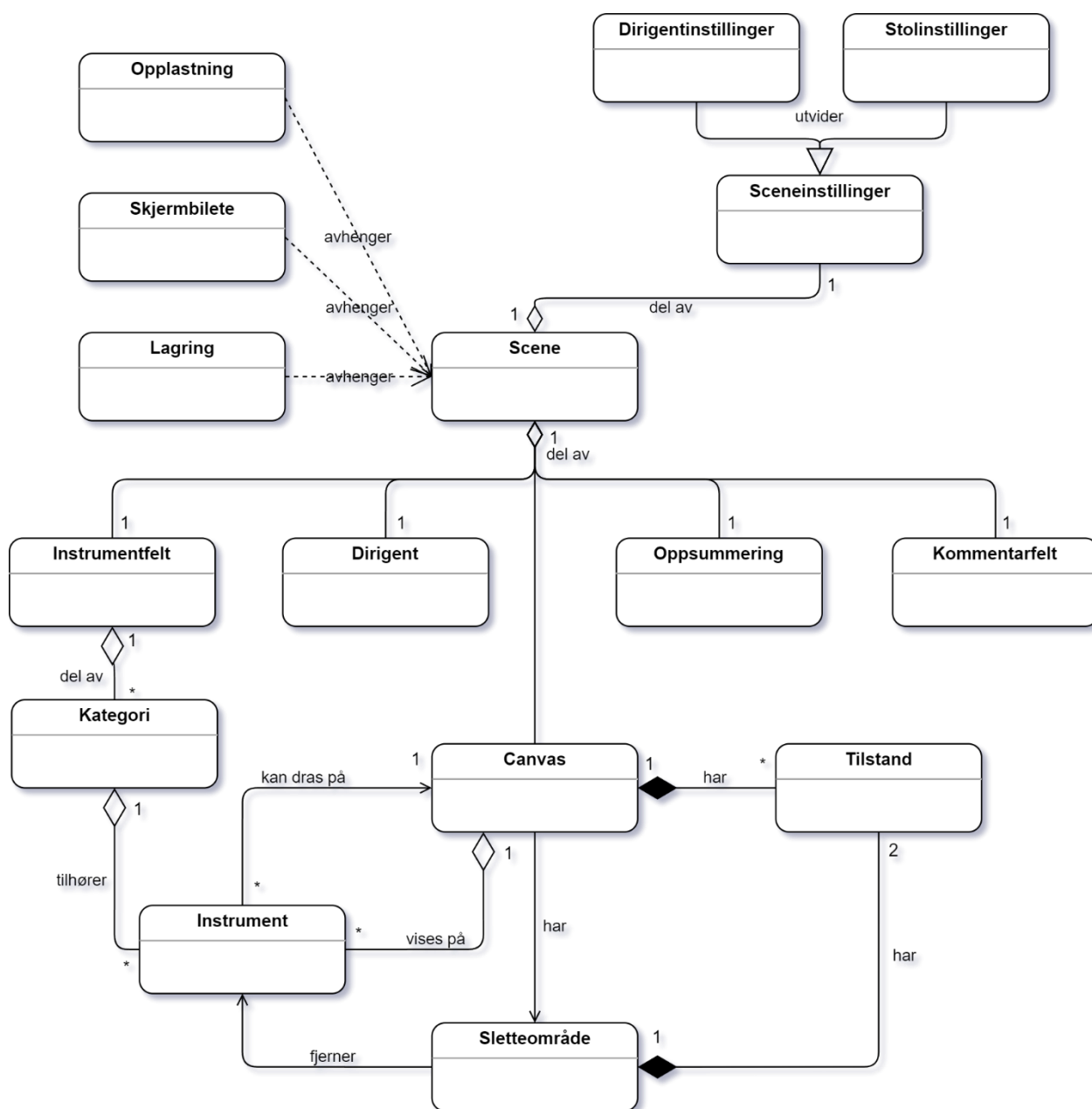
Dette kapitlet tar for seg designet av prosjektet og utforsker ulike tilknyttede designaspekter. Kapitlet gir en oversikt over hvordan prosjektet er satt opp, hvilke alternativer som ble vurdert, og hvilke valg som ble tatt i forhold til teknologi og metoder. Videre diskuteres integrasjon med oppdragsgiverens systemer og betydningen av brukertesting for dette prosjektet.

3.1 Forslag til løsning

Under utviklingsprosessen for applikasjonen Sceneoppsett var det tett samarbeid med oppdragsgiver Styreportalen for å identifisere passende løsningsalternativer. Det valgte alternativet oppfyller utgangspunkt kravene fra oppdragsgiver og vil sikre integrasjon av applikasjonen i oversystemet Program.no. Dette kapitlet presenterer de valgte alternativene, samt andre teknologier og tilnæringer som ble tatt i betraktning.

3.1.1 Domenemodell

I utviklingen av en applikasjon er det viktig å definere domenet tidlig i prosessen, dette er for å sikre forståelse for hvordan de forskjellige komponentene og funksjonene skal samhandle i en applikasjon (Fowler og Parsons, 2011). En av metodene som kan brukes for dette er DDD, nevnt tidligere i kapittel 2.4.4 (domenekunnskap). Som kan illustreres og beskrives nærmere ved bruk av UML notasjon.



Figur 3-1 UML Domenemodell over applikasjon Sceneoppsett (modellert ved hjelp av draw.io)

Denne forenklete modellen gir en kort beskrivelse av domenet til applikasjonen Sceneoppsett, der modellen er utformet i samsvar med UML notasjonen, ved bruk av DDD kjerneprinsipper som i korte trekk går ut på å fokusere på kjerneområdet og domenelogikken som betyr å identifisere viktigste konseptene og prosessene i domenet og sørge for at dette er tatt hånd om. Start med enkle modeller og deretter utvid til mer komplekse modell ettersom forståelsen for domenet øker, slik som i tilfelle representert ovenfor er en forenklet versjon av domenemodellen. Til slutt samarbeid

med domeneeksperter som i dette prosjektets tilfelle er oppdragsgiver for å anskaffe bedre forståelse av domeneområdet (Marinescu og Avram, 2007).

Selv om domenedrevet modellering med UML notasjon er veldig kjent tilnærming, er det også verdt å nevne andre tilnærminger som er blitt tatt opp til vurdering. Blant disse er Entity–relationship (ER) – modellering som går ut på å identifisere entiteter som utgjør et system, deres respektive attributter og relasjoner mellom dem. Entiteter kunne i dette tilfelle vært komponenter til applikasjonen og deres attributter altså deres egenskaper, samt relasjoner altså samhandling mellom dem (Thalheim, 2013).

Et annet alternativ som har vært opp til vurdering er Unified Process (UP), som er heller en iterativ og inkrementell metode for å utvikle et system som tar hensyn til hele livssyklusen til systemet (Eductive, 2023). Dette kunne være det beste alternativet å jobbe med i forhold til DDD, siden UP tar for seg alt fra kravspesifikasjon til testing og implementering av løsningen.

Selv om ER og UP er anerkjente og verdifulle metoder for å modellere domener, kan DDD være en mer egnet tilnærming for dette prosjektet. Dette kan begrunnes med at DDD tilbyr mer effektive verktøy for å beskrive og modellere forbindelser mellom komponenter og deres funksjoner, noe som er særlig viktig for å illustrere systemets virkemåte. Ettersom prosjektet potensielt kan få flere krav, er det også viktig at modellen skal kunne skaleres og tilpasses enkelt. Derfor stiller den valgte tilnærmingen sterkere også på dette området. Videre legger et av tidligere nevnte DDD – prinsipper vekt på at modellen kan gjøres mer kompleks etter hvert, noe som innebærer en iterativ arbeidsprosess og dermed forenkler arbeidet med prosjektet.

3.1.2 Rammeverk og bibliotek

Det eksisterer rammeverk og biblioteker som undersøkes for å identifisere løsninger for utviklingen av den nye applikasjonen. Hvert rammeverk besitter unike karakteristikk og funksjoner. Nedenfor presenteres en generell forklaring av de ulike rammeverkene.

3.1.2.1 React

React (Meta, 2023) er et JavaScript-bibliotek spesifikt utformet for konstruksjon av webapplikasjoner. Det vedlikeholdes av Meta, tidligere kjent som Facebook. React anvender en

sammensetning av JavaScript, HyperText Markup Language (HTML) og Cascading Style Sheets (CSS) for å produsere interaktive og dynamiske webapplikasjoner.

3.1.2.2 Next.js

Next.js (Vercel, 2023) er et rammeverk basert på React biblioteket for webapplikasjoner som fokuserer på klient- og server-rendering, innkapslet ruting, datainnhenting og Application Programming Interface (API) – ruter. Dette gjør Next.js til et passende valg for utviklingen av webapplikasjoner som krever hyppig nettverkskommunikasjon.

3.1.2.3 Node.js

Node.js (OpenJS_Foundation, 2023a) er et rammeverk spesielt utformet for utvikling av baksystemet i webapplikasjoner. Node.js benyttes for å koble webapplikasjoner til databaser, håndtere nettverkstilkoblinger og å være vert for nettsider.

3.1.2.4 Express.js

Express.js (OpenJS_Foundation, 2023b) er et rammeverk basert på Node.js-plattformen som behandler Hypertext Transfer Protocol (HTTP) – forespørsler. Det er et minimalistisk og hurtig verktøy for konstruksjon av webapplikasjoner. Express.js tilbyr en enkel og effektiv metode for å administrere ruting, sikkerhet og håndtering av forespørsler og svar.

3.1.2.5 Diskusjon av alternativene

Riktige valg av rammeverk eller bibliotek har stor innvirkning på utviklingen av en webapplikasjon, da ulike løsninger er mer egnet for forskjellige typer applikasjoner. React er et fleksibelt bibliotek som gir utviklere større frihet sammenlignet med mer restriktive rammeverk. Dette skyldes at rammeverk er ofte mer spesialiserte og rettet mot bestemte bruksområder, som tidligere nevnt. På en annen side kan React brukes til å utvikle nesten alle aspekter av en webapplikasjon, noe som gir utviklere større fleksibilitet og mulighet til å tilpasse løsningen etter behov.

For dette prosjektet har oppdragsgiveren spesifisert at React skal benyttes i utviklingen av den nye løsningen. Dette begrunnes med at resten av deres system allerede er programmert ved hjelp av

React, noe som vil muliggjøre enkel integrasjon av applikasjonen i deres system, samt redusere ressurser allokert til drift og vedlikehold av applikasjonen. Dette ble konkludert i et møte med oppdragsgiveren. Imidlertid er det verdt å undersøke andre alternative verktøy som kan benyttes for gjennomføring av prosjektet.

3.1.3 Lagring av data

Applikasjonen krever en løsning som kombinerer flere teknologier for å lagre informasjon knyttet til brukerens sceneoppsett, med muligheten for å hente inn sceneoppsett på et senere tidspunkt for videre endringer. Følgende alternativer diskuteres.

3.1.3.1 JSON

JSON (JavaScript Object Notation) er et standardformat som brukes for datautveksling. Formatet er tekstbasert og har en syntaks som ligner objekter i JavaScript. Denne likheten muliggjør konvertering mellom JavaScript-objekter til JSON strenger, noe som er spesielt nyttig i arbeid med React. JavaScript-objektene kan deretter brukes i React-applikasjoner for å håndtere og vise data på en enkel og effektiv måte, samt konverteres enkelt til JSON og lagres til database.

3.1.3.2 Dokumentdatabase

En dokumentdatabase er i hovedsak av type NoSQL – database som organiserer og lagrer data i dokumenter. Dette vil si at hvert dokument er en egen entitet som inneholder informasjon i et bestemt format som for eksempel JSON. Denne typen database er velegnet for webapplikasjoner der data kan ha forskjellige attributter og struktur kan variere. Dette gir da fleksibilitet for lagring og handtering av data, og kan tilpasses enklere ved endring i krav og behov.

3.1.3.3 Firebase

Firebase er en plattform som gir utviklere ulike verktøy og tjenester for å bygge og skalere applikasjoner enklere. Den var først grunnlagt som et selvstendig selskap i 2011, og deretter kjøpt opp av Google i 2014, som har videreutviklet Firebase-plattformen (Google, 2023a). Firebase tilbyr to databaseløsninger som er Firebase Realtime og Cloud Firestore der begge løsningene er av type NoSQL – database.

- Firebase Realtime lar utviklere lagre og synkronisere data i sanntid. Her brukes det JSON – struktur for dataorganisering og er egnet for applikasjoner som krever sanntidsfunksjonalitet.
- Cloud Firestore er en mer fleksibel og skalerbar databaseløsning, og lar utviklere lage mer spesifikke spørringer, som betyr at det kan hentes akkurat det applikasjonen har behov for.

Firebase støtter også frakoblet funksjonalitet og integrerer essensielle egenskaper som autentisering, filbehandling og analyseverktøy.

3.1.3.4 Diskusjon av alternativene

I diskusjonen om de ulike alternativene rettes nå vurderingen mot hvordan hver løsning kan oppfylle applikasjonens krav til lagring, håndtering og gjenbruk av data. JSON anerkjennes som et enkelt og effektivt format for datautveksling, spesielt med tanke på applikasjonens bruk av React. I den nåværende løsningen benyttes Firebase Realtime Database, men det er blitt påpekt under et møte med oppdragsgiver at dette ikke er optimalt på grunn av antall spørringer som kreves for datahenting. Oppdragsgiver ønsker derfor at Cloud Firestore skal benyttes som database – løsning for prosjektet. Cloud Firestore muliggjør mer spesifikk datahenting, reduserer antall spørringer og trafikk, noe som resulterer i kostnadsreduksjon knyttet til drift av løsningen. Dette er konklusjonen som er blitt gjort i samråd med oppdragsgiver.

3.2 Valgt løsning

Oppdragsgiver understreker viktigheten av at den foreslåtte løsningen er kompatibel med deres eksisterende system. Fra et funksjonelt perspektiv, er oppgaven relativt åpen siden funksjonalitet kan løses slik utviklerne ønsker, imidlertid spesifiserer oppdragsgiver enkelte funksjonelle krav som må tilfredstilles. Brukergrensesnittet gis betydelig frihet innenfor prosjektets rammer. Applikasjonen skal utvikles ved hjelp av React, Node.js og Cloud Firestore, som velges på grunn av oppdragsgiverens preferanser og deres kunnskap.

For å utarbeide en visuell løsning vil det bli gjennomført brukertester som blir presentert senere i rapporten der ulike demonstrasjoner av applikasjonens utforming presenteres. Disse testene tar hensyn til både funksjonelle og visuelle aspekter. Plasseringen av ulike funksjoner og elementer

inspireres av sammenlignbare applikasjoner som presenteres i kapittel 2.3 (ressurser), og som deler likhetstrekk med den foreslåtte løsningen.

3.3 Valg av verktøy

Prosjektet tar i bruk følgende teknologi og verktøy for utvikling og kommunikasjon gjennom hele prosessen. Sentralt i samarbeidet om kodeutvikling benyttes GitHub, en plattform for lagring og samarbeid om kildekode som muliggjør effektivt samarbeid på det samme prosjektet. For håndtering av datahenting og -opplasting til GitHub, anvendes Git, et versjonskontrollsystem, og GitKraken, et brukervennlig grensesnitt for Git som visualiserer og forenkler bruken av Git-kommandoer via et grafisk brukergrensesnitt.

Kommunikasjon med oppdragsgiver foregår primært gjennom Slack og Whereby for å gjennomføre digitale møter. Internt mellom utviklerne benyttes Messenger og Discord som kommunikasjonsplattformer. Microsoft Teams fungerer som en sentral lagringsplass for alle dokumenter slik som Word, PDF og bilder og beskytter mot skrivekonflikter når flere personer jobber med det samme dokumentene.

Canvas benyttes som et verktøy for kommunikasjon med intern veileder for prosjektet og tilgang til nødvendig kursmaterieell. Microsoft Word er hovedverktøyet for skriving av rapporter og andre støttedokumenter, mens Microsoft Excel brukes til å lage tabeller, matriser og diagram. Draw.io anvendes for konstruksjon av diagrammer og domenemodeller. I utviklingsprosessen benyttes Visual Studio Code (VSCode) som integrert utviklingsmiljø (IDE). Når det gjelder selve utviklingen av applikasjonen, vil React, Node.js og Firebase være de primære utviklingsverktøyene.

3.4 Prosjektmetodikk

For å sikre en best mulig utførelse av prosjektet, blir det tatt i bruk flere metodikker som benyttes for å veilede prosjektet i riktig retning.

3.4.1 Utviklingsmetodikk

For å enkelt tilpasse seg endringer og håndtere problemer underveis i utviklingsarbeidet, følges en iterativ og inkrementell utviklingsmetodikk. Utviklingsprosessen inkluderer samprogrammering og perioder med separat arbeid, der programmeringen utføres i henhold til smidige metoder. Dette legger til rette for iterativ utvikling og kontinuerlig forbedring av programmet.

Ved å dele arbeidet opp i mindre oppgaver som skal fullføres innenfor en bestemt periode eller iterasjon, sikres kontinuerlig tilbakemelding. Dette gjør det mulig å gjennomføre endringer, legge til eller fjerne funksjoner underveis i arbeidet, og oppnå smidig utvikling. Programmet holdes kjørbart og testbart til enhver tid, noe som er en av fordelene ved å bruke React. Løpende tilbakemeldinger bidrar til kontinuerlig forbedring av applikasjonen, basert på innspill som kommer frem under hver iterasjon.

I senere iterasjoner planlegges det å gjennomføre brukertesting og innhente tilbakemeldinger fra oppdragsgiverens kunder for ytterligere forbedring av produktet.

Metoder fra Ekstrem Programmering (XP) (Bell, 2001) benyttes gjennom hele utviklingsperioden av applikasjonen, sammen med Kanban-metodikken for å styre arbeidsflyten. Disse metodene inkluderer:

- Parprogrammering: Utviklerne arbeider i par, der én person koder mens den andre observerer og gir umiddelbar tilbakemelding. Dette fremmer økt kvalitet og samarbeid i utviklingsprosessen.
- Kontinuerlig integrasjon: Endringer og ny funksjonalitet blir kontinuerlig integrert i koden, noe som bidrar til å identifisere og løse problemer tidlig i utviklingsprosessen. Kanban-metodikken gjør det mulig å visuelt styre og overvåke arbeidsflyten for kontinuerlig integrasjon av oppgaver.

- Inkrementerende design: Utviklingen av systemet skjer gradvis, med fokus på å levere fungerende deler av applikasjonen i hver iterasjon. Dette muliggjør tilpasning til endringer, prioritering av funksjonalitet og enklere gjennomføring av brukertesting. Kanban er også en nyttig metode for planlegging av inkrementerende design.
- Slack: Som en del av XP-filosofien inkluderes oppgaver som kan droppes om nødvendig, slik at det er fleksibilitet til å tilpasse seg endringer og prioritere viktig funksjonalitet. Kanban-metodikken er også nyttig i denne sammenhengen, da enkelte oppgaver kan tas ut av planen eller utsettes etter behov, for eksempel ved tidsbegrensninger.

Utviklingsprosessen er også planlagt over en lengre periode, noe som reduserer hastverk og muliggjør en mer fornuftig og målrettet tilnærming til programmeringen.

Selv om oppdragsgiver ikke stiller krav til dokumentasjon, dokumenteres utviklingsprosessen gjennom denne rapporten. Funksjoner, metoder, klasser og andre kodekomponenter skal bli dokumentert ved bruk av JSDoc⁶. Dokumentasjonen bidrar til å gjøre koden mer forståelig og lesbar. Siden oppdragsgiver sannsynligvis vil fortsette utviklingen av det endelige produktet, vil tilstrekkelig dokumentasjon i koden forenkle deres arbeid med å videreføre prosjektet.

3.4.2 Fremdriftsplan

Prosjektets framdriftsplan er satt opp i form av et Gantt-diagram, som fungerer som en retningslinje for prosjektets fremgang. Imidlertid vil det ikke være et absolutt bindende verktøy, da det kan være nødvendig å tilpasse planen underveis. For å ha kontroll over prosjektets tidsbruk, vil antall arbeidstimer bli loggført. Diagrammet tar utgangspunkt i utvalgte innleveringer i bachelor faget og store milepæler ved programmering.

I begynnelsesfasen vil mye tid bli investert i å lære utviklingsrammeverket React. Når tilstrekkelig kunnskap er oppnådd, kan smidig utvikling benyttes for å utforske rammeverket ytterligere.

⁶ Dokumentasjonverktøy for dokumentering av Javascript kildekode

For å etablere et solid grunnlag for oppgaven vil ideer til klassesdiagrammer bli benyttet.

Innleveringer & frister	Start	Lengde	Januar				Februar				Mars					April				Mai		
			UKE 2	UKE 3	UKE 4	UKE 5	UKE 6	UKE 7	UKE 8	UKE 9	UKE 10	UKE 11	UKE 12	UKE 13	UKE 14	UKE 15	UKE 16	UKE 17	UKE 18	UKE 19	UKE 20	UKE 21
Oppstartsmøte & møte med veileder	10.01.2023	2 uker																				
Prosjekthåndbok iterasjon 1 og Visjonsdokument iterasjon 1	23.01.2023	2 uker																				
Midtveisrapport	06.03.2023	4 uker																				
Utkast av rapporten	24.04.2023	10 uker																				
Endelig rapport	22.05.2023	15 uker																				
Koding prosjekt																						
Oppsett av prosjekt	23.01.2023	2 uker																				
Grunnverket av webapplikasjonen	30.01.2023	4 uker																				
Implementering av funksjoner	27.02.2023	8 uker																				
Brukertesting	14.04.2023	1 uke																				
CSS & annet design	25.04.2023	2 uker																				
Møter																						
Hver fredag med oppdragsgiver (ble ikke møter på rød)	Fredag 10:00	15 min																				
Møter med veileder	En gang i månaden	30 min																				

Figur 3-2 Gantt-diagram over fremdriftsplan laget i Microsoft Excel

Gantt-diagrammet er delt opp i tre deler: Innlevering og frister, Koding prosjekt og til slutt Møter. Planlagt startdato og lengde er oppgitt for hvert punkt. Enkelte punkter jobbes med kontinuerlig gjennom hele prosjektet som rapportskrivning. Grønne ruter i Møter er planlagte møter som gikk som planlagt, mens røde ruter er møter som ble avlyst.

3.4.3 Risikovurdering

Risikovurdering er en systematisk prosess som først identifiserer, deretter analyserer og evaluerer potensielle risikoer som kan oppstå i forbindelse med et prosjekt. Hensikten med risikovurdering er å belyse og analysere eventuelle risikoer, samt å lage en plan for hvordan de kan håndteres. Prosessen bidrar til å prioritere ressurser og tiltak for å minimere negative konsekvenser som risikoer kan medføre.

Tabell 3-1 Risikovurdering illustrert ved hjelp av Microsoft Excel

	Hendelse / Risiko	Årsak	Sannsynlighet	Konsekvens	Risiko- produkt	Tiltak
1	Applikasjonen blir ikke tatt i bruk.	Ineffektiv og lite brukervennlig	Høy (4)	Høy (4)	16	Gjennomføre brukertesting samt holde dialog med oppdragsgiver
2	Applikasjonen blir ikke ferdig før frist	Mangel på tid	Middels (3)	Høy (4)	12	Riktig tidsestimert
3	Oppdragsgiver blir ikke tilfredsstillt med resultat	Dårlig kommunikasjon	Middels (3)	Høy (4)	12	Holde kontinuerlig kommunikasjon med oppdragsgiver.
4	Dårlig sikkerhet	Personvern og sikkerhet blir for dårlig håndtert	Lav (2)	Svær Høy (5)	10	Kontinuerlig holde sikkerhet i bakhodet under utvikling.
5	Kompleksitet	Teknologien vi trenger for å lage en god applikasjon er for vanskelig	Høy (4)	Høy (4)	16	Bruke alle mulige hjelpemidler for å forstå teknologien best mulig.
6	Benyttet API, bibliotek og verktøy ikke blir vedlikeholdt	Nye oppdateringer av nettleseren kanskje ikke støttet API/biblioteker vi har brukt	Middels (3)	Svær Høy (5)	20	Sjekk hvor mye biblioteket er brukt av andre og når den kom ut. Utviklere kan bytte eller oppgradere bibliotek/API fortløpende.
7	Kompatibilitetsproblemer	At enkelte maskiner, skjermer eller nettlelere ikke er støttet av applikasjonen	Lav (2)	Høy (4)	8	Teste grundig og utvikle med hensyn på forskjellige nettlelere og skjermer.

Sannsynlighet	Svært Høy (5)	5	10	15	20	25
	Høy (4)	4	8	12	16	20
	Middels (3)	3	6	9	12	15
	Lav (2)	2	4	6	8	10
	Svært Lav (1)	1	2	3	4	5
		Svært Lav (1)	Lav (2)	Middels (3)	Høy (4)	Svært Høy (5)
Konsekvens						

Figur 3-3 Risikomatrix for utregning av risikovurdering utformet i Microsoft Excel

Risikoer vurderes basert på et risikoprodukt, beregnet ved å multiplisere sannsynlighet og konsekvens: $\text{Sannsynlighet} * \text{Konsekvens} = \text{Risikoprodukt}$. Det er imidlertid viktig å opprettholde fokus på risiko gjennom hele prosjektperioden, da uforutsette risikoer kan oppstå eller tidligere identifiserte risikoer kan være feilvurdert. Den iterative utviklingsmetodikken som er valgt, er derfor nyttig for å kunne justere seg etter eventuelle endringer i prosjektets risikovurdering.

Etter identifiseringen av de mest relevante risikoene for prosjektet, planlegges prosessen med hensyn til risikoanalysen. Dette gjøres for å minimere sannsynligheten for negativ innvirkning fra de identifiserte hendelsene. Det eksisterer en risiko for at applikasjonen ikke blir fullført før overlevering til oppdragsgiver. I tilfelle dette skulle inntreffe, har oppdragsgiver muligheten til å videreføre arbeidet med applikasjonen.

I løpet av prosjektets fremdrift kan det oppstå nye risikovurderinger som krever oppmerksomhet. Disse blir dokumentert ved å legge til punkter i risikovurderingstabellen i vedlegg 10.2 (prosjekthåndbok) for å sikre at alle identifiserte problemer blir tatt hensyn til. Deretter blir en risikovurdering gjennomført for hvert problem for å bestemme problemets betydning og risikonivå.

En spesifikk utfordring i utviklingen av applikasjonen er implementeringen av dra-og-slipp-funksjonaliteten. Manglende hensiktsmessig implementering av denne funksjonaliteten kan ha en negativ innvirkning på applikasjonens funksjonalitet.

3.5 Evalueringsplan

For å evaluere applikasjonen opprettholdes kontinuerlig kommunikasjon med oppdragsgiver. Målet er å ha applikasjonen kjørbart til enhver tid, slik at oppdragsgiver kan gi løpende tilbakemeldinger under jevnlig møter. Når applikasjonen når en tilstrekkelig utviklingsfase, skal den testes gjennom brukerundersøkelser og brukertesting. Tilbakemeldingene fra disse testene skal så analyseres og brukes til å utbedre eventuelle mangler, feil eller forbedringer.

Oppdragsgiver har kontakt med flere brukere som har erfaring fra den tidligere løsningen og som kan delta i testingen av den nye applikasjonen. Siden den primære målgruppen for applikasjonen er oppdragsgiverens kunder, kan dette gi relevant tilbakemelding som er nyttig for videreutviklingen av applikasjonen. Planen er å gjennomføre flere runder med brukertesting, men dette avhenger i stor grad av tidsrammen og hvor mye tid som er igjen når fristen nærmer seg.

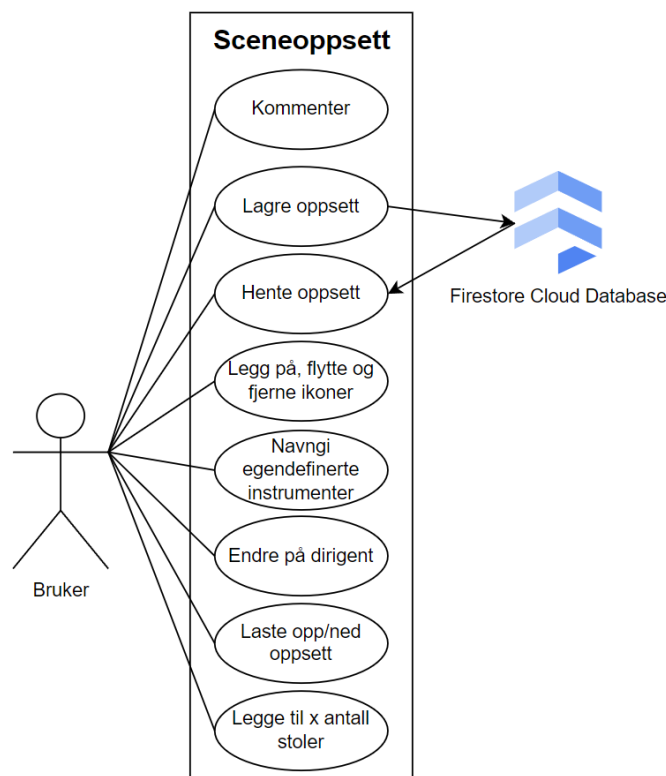
Uavhengig av utfallet vil oppdragsgiver gi tilbakemelding på det endelige resultatet for å kunne evaluere arbeidet på best mulig måte. Ved å involvere oppdragsgiver i evalueringen, kan det bli gitt konstruktiv tilbakemelding om hva som kan være videre arbeid for applikasjonen, samt en konklusjon på de oppnådde resultatene.

4 Detaljert løsning

I dette kapittelet vil det bli gått nærmere inn på hvordan prosjektet blir drevet med fokus på utviklingen av design og funksjonalitet for applikasjonen Sceneoppsett først introdusert i kapittel 2.1 (praktisk bakgrunn). Prosjektet legger stor vekt på implementasjonen av funksjonene i applikasjonen, samt designet av applikasjonen og plasseringen av de ulike komponentene. De ulike funksjonene applikasjonen tilbyr vil bli introdusert og illustrert med brukstilfeller. I vedlegg 10.3 (kravdokument) for prosjektet finnes brukstilfellediagram med respektive brukstilfellebeskrivelser, en domenemodell, samt skisser av brukergrensesnittet og applikasjonen før brukertesting.

4.1 Brukstilfeller

Brukstilfeller illustrerer funksjonene som applikasjonen implementerer og gir dermed en oversikt over hvilke funksjoner en bruker har tilgjengelig. Dette inkluderer også hvordan disse funksjonene samhandler med databaseløsningen.



Figur 4-1 Brukstilfellediagram

For å bruke applikasjonen må brukeren allerede være registrert i systemet og være innlogget. Denne funksjonaliteten blir ikke diskutert siden dette er utenfor applikasjonens domene. Applikasjonen består av ulike komponenter, inkludert en scene, et instrumentpanel, innstillinger for stoler og dirigent, et kommentarfelt, samt funksjoner for lagring og skjermbilde som er del av lagringsprosessen.

Brukeren kan dra instrumentene fra instrumentpanelet og plassere dem på hovedscenen. Det skal også være mulig å angi antall stoler som automatisk skal plasseres på scenen. I innstillingene skal brukeren kunne velge om de ønsker en dirigent med eller uten podium og notestativ, samt om de vil ha en brassband- eller janitsjarguide (Wensberg, 2023). Når brukeren er ferdig med å redigere oppsettet, kan de lagre det ved å trykke på "send inn" – knappen. Oppsettet vil da bli sendt til databasen som en JSON-streng. Brukeren kan fortsatt hente opp og redigere oppsettet før konkurransen etter at det er lagret.

Når oppsettet skal lagres, skal et bilde av sceneoppsettet, kommentaren og hvert instruments posisjon på scenen lagres i databasen. En liste i applikasjonen oppdateres etter hvilke instrumenter som er plassert på scenen. Listen skal hjelpe brukeren med å sjekke at riktig antall instrumenter er plassert på scenen samt holde oversikt over alt som er utplassert på scenen.

For å slette elementer på scenen, skal brukeren kunne trykke og holde på et instrument. Et rødt felt vil da skal kunne vises øverst i scenekomponenten. Deretter skal instrumentet kunne dras opp til dette feltet og slippe det for å fjerne det fra scenen.

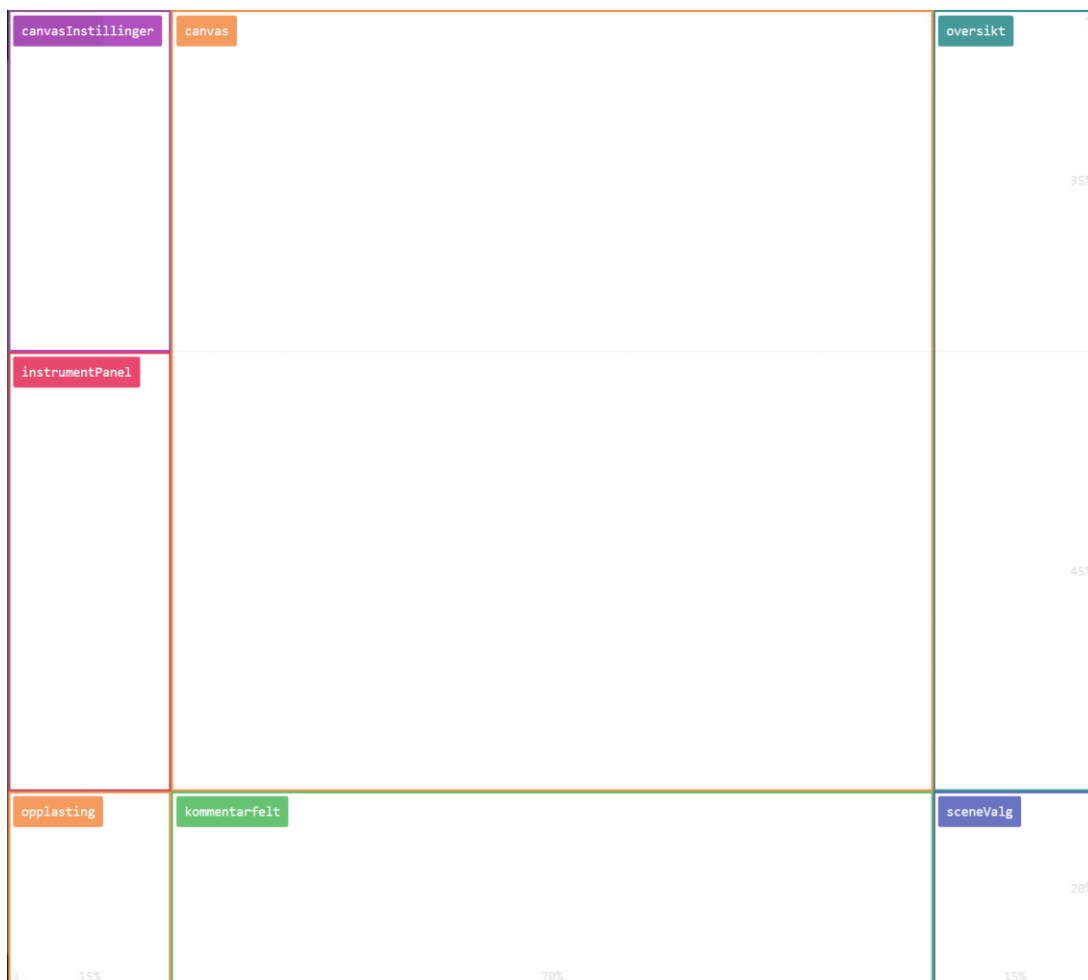
Applikasjonen utvikles som dynamisk skalerbar, noe som betyr at den tilpasser seg forskjellige skjermstørrelser både ved oppstart og under bruk. Dette er spesielt viktig ettersom et oppsett kan opprettes på én datamaskin og deretter hentes på en annen datamaskin med en annen skjermopløsning, samt at brukeren kan ha behov for å endre nettleservinduets størrelse under bruk. På denne måten kan brukere lage oppsett for konkurranser som deretter kan hentes av arrangøren på en annen enhet med en annen skjermopløsning. Når instrumenter er plassert på scenen, skal de tilpasse seg endringer i vindusstørrelsen når nettleservinduet skaleres opp eller ned.

Ved hjelp av Firebase-plattformen unngås det overbelastning av applikasjonen, da Firebase skalere ressursene etter antall brukere og belastning. Dersom mange brukere trenger å bruke

applikasjonen, vil Firebase-plattformen tilpasse seg etter dette behovet. Når Firebase skalerer, vil også prisen for å bruke Firebase skalere i samsvar med ressursbruken (Google, 2023b).

4.2 Brukergrensesnitt

Oppdragsgiver gir frihet til å utvikle et nytt brukergrensesnitt der det er ønskelig å modernisere designet og sikre en bedre arbeidsflyt for brukeren. Fokuset i de fleste kravene er på funksjonalitet, ikke spesifikke brukergrensesnittspesifikasjoner. Det nye brukergrensesnittet henter inspirasjon fra andre applikasjoner omtalt i kapittel 2.3 (ressurser), spesielt med tanke på komponentplassering i nettleservinduet. Analyser av hvordan andre applikasjoner plasserer sine komponenter blir sammenlignet med designet fra den tidligere løsningen, som er presentert i kapittel 2.1.1 (tidligere arbeid). Denne sammenligningen danner grunnlaget for utviklingen av det nye brukergrensesnittet.



Figur 4-2 Komponent plassering ift. nettleservinduet

I den presenterte figuren kan plasseringen av de ulike komponentene i forhold til nettleservinduet observeres. Figuren fungerer som en grov skisse og viser kun plasseringen av hovedkomponentene innenfor den overordnede komponenten, scenen. Scenen fungerer som foreldrekomponent for alle de synlige komponentene på figuren.

Designet er gjort med hensyn på brukerne og for å skape en god opplevelse for dem. Hva brukeren skal gjøre og hvordan brukeren skal oppnå resultatet skal være lett å forstå. Komponentene er plassert med omtanke for arbeidsflyten. Canvas området tar mest plass i applikasjonen ettersom det er det viktigste området og hvor mesteparten av tiden vil bli brukt.

For at applikasjonen skal være intuitivt og lett å bruke må brukergrensesnittet være gjort omtenkstomt. For å få dette til blir det tatt i bruk metoder som KISS (The_Interaction_Design_Foundation, 2023), content-first design (Hampton, 2017) og Universell Utforming (Goldsmith og PRP Architects (Firm), 2000).

For å gjøre applikasjonen mest mulig universell, skal alle instrumentene ha et "alt"-attributt som beskriver hva bildet er dersom det skjer en feil og bildet ikke laster skikkelig. Det er også slik at musen er over instrumenter på scenen kommer navet på instrumentet opp. Det er også tilrettelagt ved at et panel på høyresiden holder oversikt over hvilke ikoner og hvor mange som er på scenen.

Ved valg av farger benyttes 10, 30, 60-regelen (Christina, 2022). Fargepaletten deles inn i tre deler: 10, 30 og 60 prosent. Den 60 prosent delen representerer primærfargen som brukes som bakgrunnsfarge og utgjør basen i fargepaletten. Sekundærfargen, som utgjør 30 prosent, brukes til overskrifter eller bannere på siden. De resterende 10 prosentene er dedikert til viktige elementer på siden, som knapper, viktige punkter eller varsler. Bruk av denne regelen bidrar til å opprettholde en enkel og balansert fargestil på siden.

For å kontrollere at fargevalgene har riktig kontrast, benyttes verktøyet Accessibility Insights for Web. Dette er et tilleggsverktøy for nettlesere som dobbeltsjekker om retningslinjene i forhold til Web Content Accessibility Guidelines (WCAG) 2.1 Level AA overholdes (Microsoft, 2023c).

4.3 Utvikling

Applikasjonen vil bestå av flere komponenter som implementerer tidligere nevnte funksjonaliteter. Denne delen introduserer en detaljert tilnærming til utviklingen av disse komponentene, samt hvordan lagringsprosessen i databasen er planlagt. Verktøyene som benyttes i utviklingsprosessen blir beskrevet, og prosjektets inkrementelle og iterative arbeidsprosess blir ytterligere forklart.

4.3.1 Komponenter

Foreldrekomponenten scene inneholder alle påfølgende komponenter, i tillegg til en valgkomponent som skal implementere funksjoner som lagring av oppsettet og funksjonen for å ta et skjermbilde av oppsettet. I dette underkapittelet vil det bli gitt en nærmere gjennomgang av hvordan underkomponentene skal implementeres og samhandle med hverandre.

4.3.1.1 Canvas

Canvasområdet er den mest sentrale komponenten i applikasjonen, hvor brukeren skal kunne plassere og organisere instrumenter etter ønsket oppsett. Innenfor canvasområdet kan instrumenter flyttes individuelt eller i grupper ved å markere et område og flytte alle instrumentene innenfor det valgte området. Canvasområdet kan også styres av komponent for canvasinnstillinger som skal inneholde funksjon som vil forenkle prosessen med å legge til stoler med å la bruker velge ulike typer stoler, antall stoler, antall rader og eventuell bueformet formasjon.

Hvis brukeren ønsker hjelp med å sette opp et janitsjar- eller brassoppsett, skal dette kunne gjøres ved hjelp av guideverktøy fra komponent for canvasinnstillinger. Når en av disse oppsettene er valgt, skal det vises et omriss av det valgte oppsettet på canvasområdet. Brukeren skal kunne deretter plassere instrumentene de ønsker å inkludere i oppsettet på scenen.

En annen vesentlig funksjon er oversiktskomponenten, som kontinuerlig skal gi brukeren en oppdatering over alle elementer plassert på canvasområdet. Dette skal oppnås ved at oversiktskomponenten overvåker tilstanden til samtlige utplasserte elementer på canvasområdet.

4.3.1.2 Instrumenter og instrumentpanel

Oppdragsgiver angir hvilke instrumenter som skal inkluderes i oppsettet, og alle valgte instrumenter er vanligvis brukt i typiske korpsoppsett. Brukerne skal ha muligheten til å benytte egendefinerte instrumenter om de ønsker å spesifisere et instrument som ikke finnes på listen. Dette skal gjøres ved at applikasjonen tilbyr noen generiske figurer, for eksempel sirkler og firkanter, som brukerne kan endre navnet på ved å dobbeltklikke på instrumentet.

Ikonene for instrumentene som oppdragsgiver har gitt, er uten opphavsrett og kan fritt benyttes. Oppdragsgiver har uttalt at ikonene ikke nødvendigvis må være de som er gitt, men at andre ikoner kan brukes hvis de er mer hensiktsmessige. Gjennomgang av ikoner har dermed lav prioritet i prosjektet, men gir nødvendig frihet ved behov.

Instrumentpanelet skal vise alle instrumenter som brukeren kan benytte, inkludert egendefinerte instrumenter. Ettersom det ikke er mulig å vise alle instrumentene samtidig, ønsker oppdragsgiver at de skal kunne grupperes. Disse gruppene skal være skjult som standard, og brukeren skal deretter kunne vise de gruppene de ønsker å bruke instrumenter fra, og skjule dem igjen ved behov. Dette skal bidra til økt oversikt og bedre utnyttelse av plassen.

Brukerne kan dra instrumenter fra instrumentpanelet og plassere dem på canvas. Når brukeren slipper instrumentet på canvas, blir posisjonen lagret i forhold til canvassområdet. Når instrumenter flyttes rundt på canvas, blir posisjonen oppdatert når instrumentet slippes.

4.3.1.3 Kommentarfelt

Brukeren skal ha muligheten til å legge igjen en kommentar ved oppsettet, som kan være nyttig ved generering av en sceneoppsettrapport for arrangøren dersom det er spesielle hensyn som må tas eller noe brukeren ønsker å bemerke til neste gang oppsettet skal redigeres. Kommentarfeltet vil være tomt som standard, og det er ikke nødvendig å fylle det ut for å kunne lagre oppsettet.

4.3.2 Database

Løsningen for lagring, presentert i kapittel 3.1.3 (lagring av data), innebærer bruk av JSON – format for å representere dataene som skal lagres i databasen. Når applikasjonen er i bruk, blir midlertidige data, som inkluderer canvasområdets størrelse, dirigent, guide, kommentarfelt og

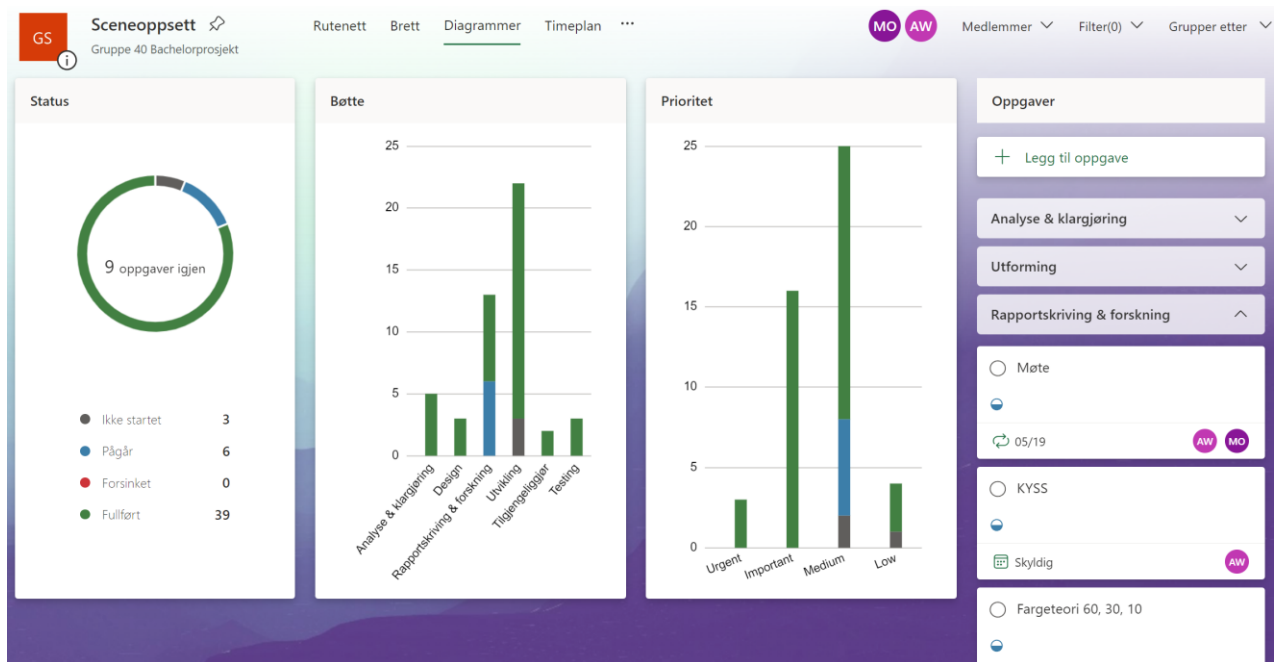
instrumentposisjoner på canvas, konvertert til JSON-format ved hjelp av en funksjon før lagring. Størrelsen på det nåværende canvasområdet skal kun brukes for å konvertere koordinater fra piksler til prosent, og derfor ikke lagres i databasen. I tillegg tas et skjermbilde av canvasområdet. Dette sikrer at all nødvendig informasjon lagres for senere henting av oppsettet, enten for å endre det eller generere en fullstendig sceneoppsettrapport for arrangøren.

For å sikre kompatibilitet med alle skjermstørrelser skal posisjonen angis til hvert instrument som en prosentandel av canvasområdet, i stedet for piksler, når dataene skal lagres i databasen. Ved henting av oppsettet fra databasen og endring av skjermstørrelse, skal instrumentenes nye posisjoner beregnes i pikselkoordinater ved hjelp av den oppdaterte størrelsen på canvasområdet og prosentverdiene for instrumentene hentet fra databasen. Funksjonen skal dokumenteres i koden i henhold til tilnærmingen beskrevet i kapittel 3.4.1 (utviklingsmetodikk).

4.3.3 Kanban

Kanbanmetodikken, først introdusert i kapittel 3.4.1 (utviklingsmetodikk), benyttes for både utvikling og dokumentasjon av prosjektet gjennom oppgaver knyttet til rapportskrivningen. Microsoft Planner (Microsoft, 2023d) er verktøyet som benyttes for å implementere Kanbanmetodikken gjennom å lage en oppgavetavle. Dette gjøres ved å opprette og kategorisere oppgaver, ofte kalt "buckets", som visualiseres på et tavlelignende grensesnitt. Disse oppgavene kan flyttes mellom ulike stadier som "ikke startet", "pågår", "forsinket" og "fullført", noe som gir en visuell representasjon av arbeidsflyten.

Denne tilnærmingen fører til en smidig og iterativ arbeidsprosess med jevnlig møter som sikrer kontinuerlig tilbakemelding. Oppgavene kan fordeles mellom utviklere og enkelt administreres ved å tildele, utsette og legge til nye eller slette oppgaver gjennom hele prosjektperioden. Tavlen tar utgangspunkt i den initielle fremdriftsplanen for prosjektet introdusert i kapittel 3.4.2 (fremdriftsplan) som også finnes i vedlegg 10.2 (prosjekthåndbok).

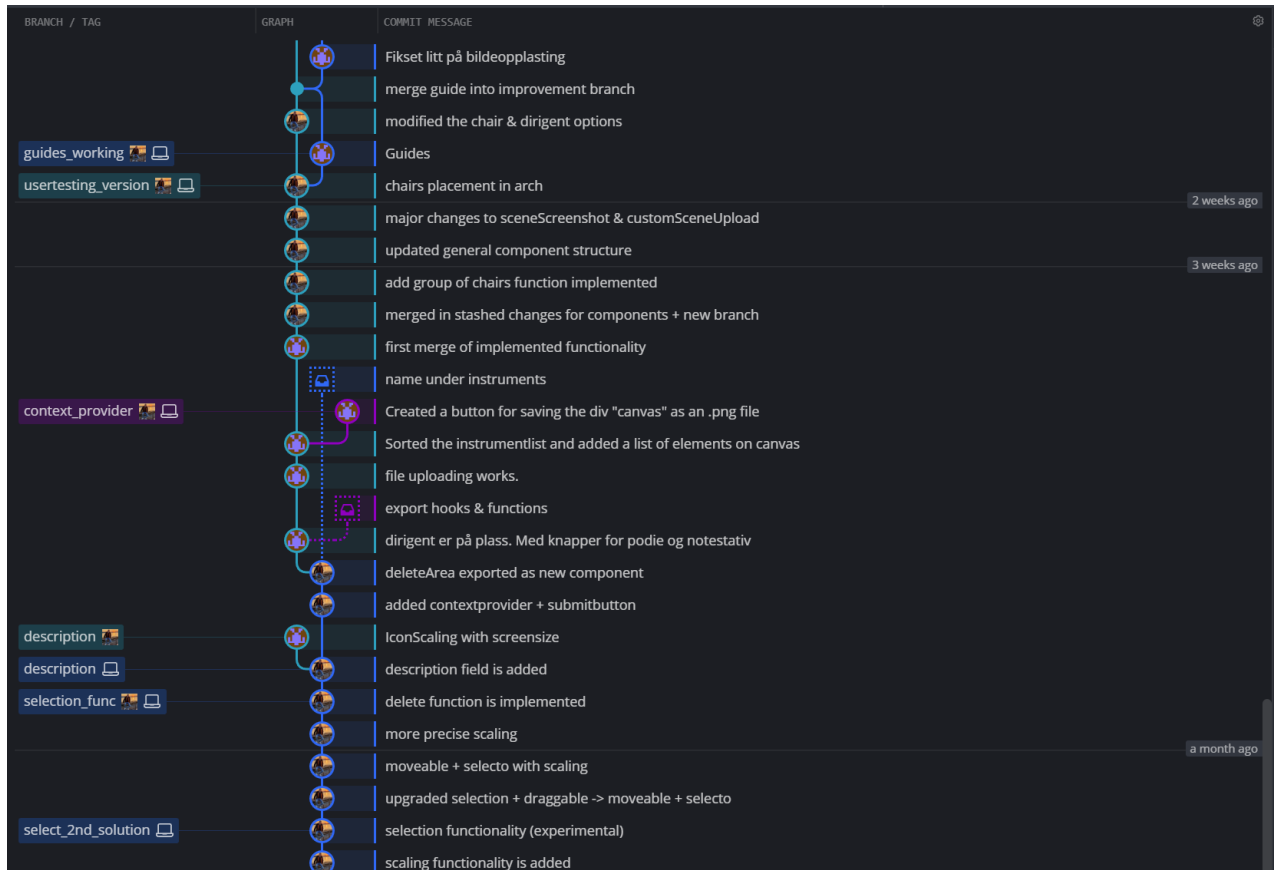


Figur 4-3 Kanban statistikk over prosjektet generert i Microsoft Planner

Illustrert i figuren, er oppgavene delt inn i kategorier som skiller forarbeid, utvikling, dokumentasjon og testing. Oppgavene har ulike prioriteringer, og noen er satt opp som gjentakende oppgaver over en periode, for eksempel møter med oppdragsgiver og veileder. Figuren illustrerer en oppsummering av planen, og gir en oversikt over status for oppgavene, antall oppgaver i hver kategori og prioritet, der fargene representerer statusen på de enkelte oppgavene.

4.3.4 Git

Samarbeid om koden til prosjektet realiseres gjennom bruk av Git versjonskontrollsystem, GitHub oppbevaringssted for prosjektfiler og GitKraken, et grafisk brukergrensesnitt for å benytte Git, som er introdusert i kapittel 3.3 (valg av verktøy).



Figur 4-4 Utklipp over commits til prosjektet fra GitKraken

Samarbeidet om kode skjer ved at deltakere av prosjektet får tilgang til et lagringssted, kjent som en "repository", lagret i GitHub. Dette lagringsstedet kloneres ned til den lokale maskinen gjennom en kloningsfunksjon tilgjengelig via Git, ved hjelp av GitKraken-grensesnittet. Prosjektoppgaver, som implementering av funksjoner, fordeles deretter. For hver oppgave opprettes "branches" (grener), som illustrert i figuren over. Dette muliggjør isolering av arbeidet for å unngå innvirkning på hovedkoden.

Når nødvendige endringer er gjennomført, lagres disse, eller "committes", og "pushes" deretter til GitHub ved hjelp av Gitfunksjoner tilgjengelig via GitKraken. Under lagringsprosessen er det mulig å legge igjen kommentarer eller en beskrivelse av hva som er gjort, som også kan

observeres i figuren. Funksjonene "merges", eller flettes til slutt sammen, til hovedgrenen – hovedkoden, slik at arbeidet utført av deltakerne samles i et felles resultat. For å holde den lokale koden oppdatert, lastes endringer fra GitHub ned kontinuerlig ved hjelp av "pull" – funksjonen fra Git. Dette muliggjør en iterativ arbeidsprosess med kontinuerlig integrasjon av endringer til prosjektet og en strukturert tilnærming til utviklingsprosessen, som effektiviseres ved hjelp av de nevnte verktøyene.

4.3.5 Dokumentasjon

Dokumentasjon for dette prosjektet utføres hovedsakelig gjennom denne rapporten, men suppleres også med detaljert dokumentasjon av kode som diskutert tidligere i kapittel 3.4.1 (utviklingsmetodikk), benyttes JSDoc for å forklare funksjonaliteten til hver enkelt funksjon og komponent, inkludert formålet, funksjonsmåte og returnerte resultater, som illustrert i figuren nedenfor.

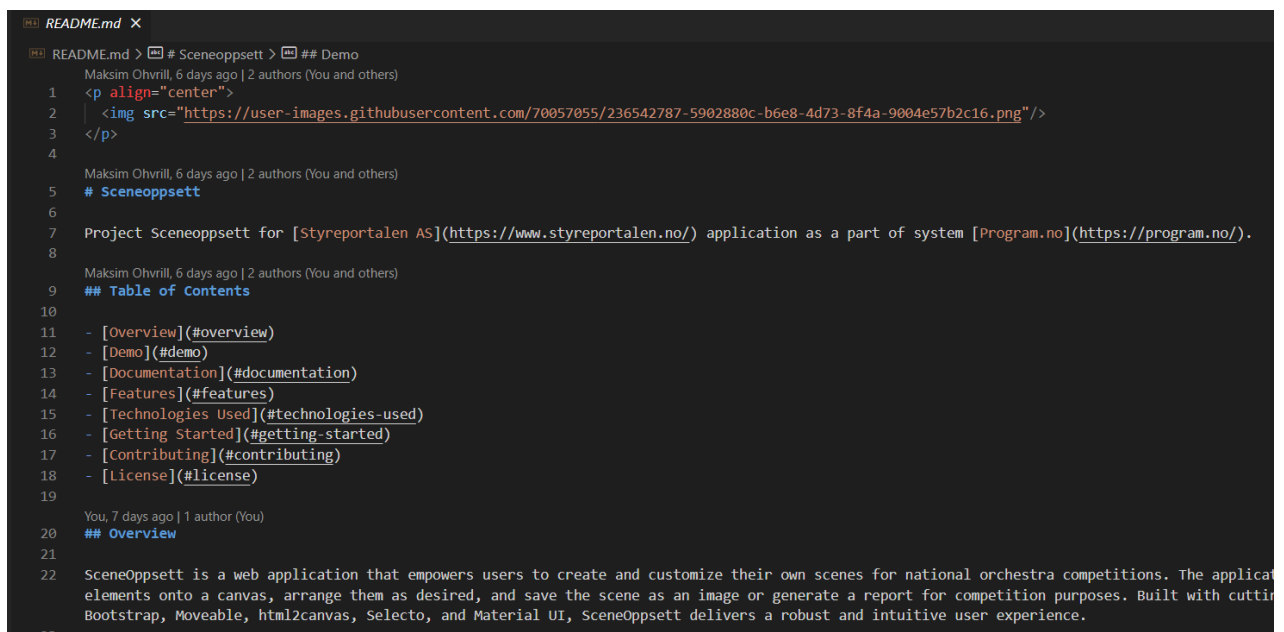
```
5  /**
6  * CustomSceneUpload is a functional React component that allows users to upload
7  * custom images for their own scenes. The component accepts files in png, jpg, or pdf format.
8  *
9  * @param {object} props - The properties passed to this component.
10 * @param {React.RefObject} props.sceneRef - A React ref object that refers to the scene element.
11 *
12 * @returns {JSX.Element} A form element with the class 'formContainer',
13 * that contains inputs for file selection, and buttons for deleting and uploading the selected file.
14 *
15 * @example
16 * // To include the CustomSceneUpload component in a parent component:
17 * <CustomSceneUpload sceneRef={sceneRef} />
18 */
19 function CustomSceneUpload(props) {
20   const [file, setFile] = useState();
21
22   function handleChange(event) {
23     setFile(event.target.files[0]);
24   }
25
26   function handleDelete() {
27     const sceneDiv = props.sceneRef.current;
28     if (sceneDiv.lastChild.className === "uploadedImage") {
29       sceneDiv.removeChild(sceneDiv.lastChild);
30       sceneDiv.children[0].style.display = "block";
31       sceneDiv.children[1].style.display = "block";
32     }
33   }
34   if (!file) {
35     setFile(null);
36   }
37 }
```

Figur 4-5 Eksempel på dokumentering av funksjon i kode v.h.a. JSDoc

Figuren over illustrerer et utdrag fra koden, merket med JSDoc kommentarer, og gir en mal for hvordan alle prosjektets funksjoner og komponenter skal dokumenteres. Slik dokumentasjon

danner grunnlaget for å generere en nettside for kodedokumentasjon, som kan brukes av utviklere hos oppdragsgiver for videre arbeid med prosjektet. Som det kan observeres i figuren, er dokumentasjonen skrevet på engelsk. Dette er et bevisst valg for å maksimere tilgjengeligheten av koden, gitt at det ikke kan forutsies hvilken bakgrunn fremtidige utviklere av prosjektet vil ha. I tillegg tillater engelsk enklere og mer presis beskrivelse av tekniske begreper.

Dokumentasjonen av prosjektet utvides ytterligere med en README-fil i GitHub lagringssted. Denne filen er ment å inneholde en detaljert beskrivelse av applikasjonens funksjonalitet, støttet av relevante illustrasjoner, samt en oversikt over de teknologiene som er anvendt. I tillegg vil filen inneholde lenker til brukertesting, applikasjonen under brukertesting, den ferdigstilte applikasjonen, og annen relevant informasjon. README-filen tjener det formål å sørge for at prosjektinformasjonen er lett tilgjengelig både for oppdragsgiver og for prosjektvurdering.



```
README.md X
README.md > # Sceneoppsett > ## Demo
Maksim Ohvrill, 6 days ago | 2 authors (You and others)
1 <p align="center">
2 
3 </p>
4
Maksim Ohvrill, 6 days ago | 2 authors (You and others)
5 # Sceneoppsett
6
7 Project Sceneoppsett for [Styreportalen AS](https://www.styreportalen.no/) application as a part of system [Program.no](https://program.no/).
8
Maksim Ohvrill, 6 days ago | 2 authors (You and others)
9 ## Table of Contents
10
11 - [Overview](#overview)
12 - [Demo](#demo)
13 - [Documentation](#documentation)
14 - [Features](#features)
15 - [Technologies Used](#technologies-used)
16 - [Getting Started](#getting-started)
17 - [Contributing](#contributing)
18 - [License](#license)
19
You, 7 days ago | 1 author (You)
20 ## Overview
21
22 SceneOppsett is a web application that empowers users to create and customize their own scenes for national orchestra competitions. The applica
elements onto a canvas, arrange them as desired, and save the scene as an image or generate a report for competition purposes. Built with cuttin
Bootstrap, Moveable, html2canvas, Selecto, and Material UI, SceneOppsett delivers a robust and intuitive user experience.
23
```

Figur 4-6 Utklipp fra README dokumentet for prosjektet

Dette er et enkelt utklipp av utgangspunkt for dokumentet som gir en hint over hvordan struktur av dokumentet skal være samt hva det skal inneholde.

5 Resultater

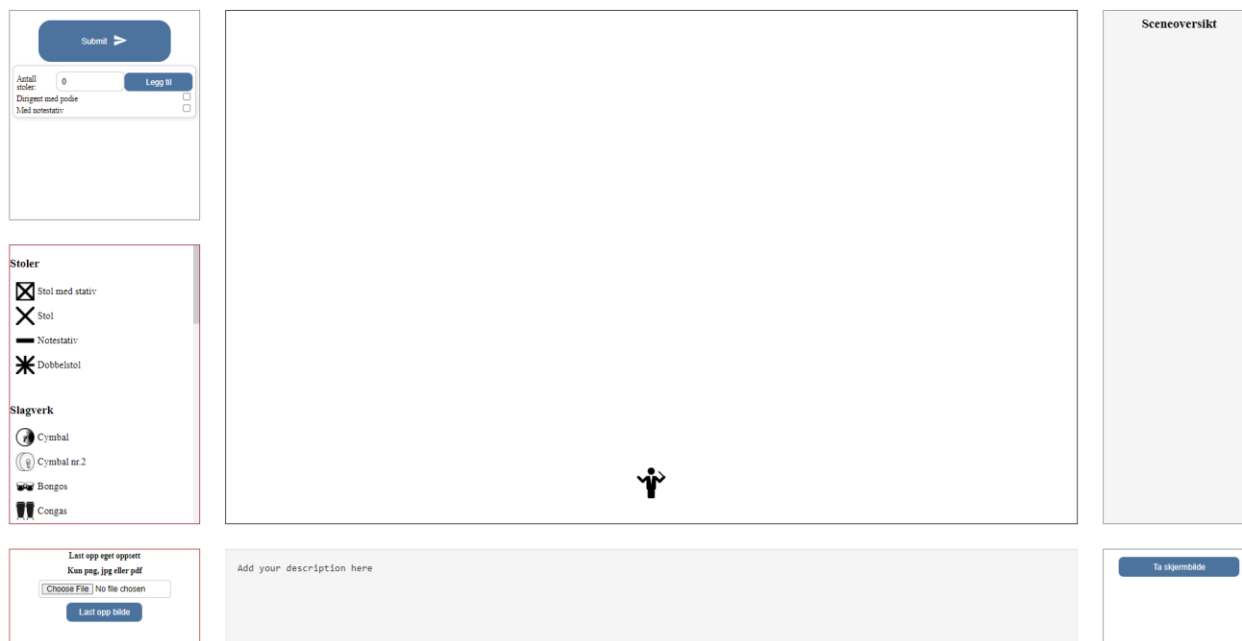
Dette kapitlet presenterer resultatene oppnådd gjennom utviklings- og testfasene av prosjektet. Det tar for seg evalueringemetodene som ble benyttet både under og etter prosjektets gjennomføring. Videre gir det en detaljert beskrivelse av oppnådde resultater gjennom prosjektets løp, og endelige produktets form. Det trekker også sammenligninger mellom den faktiske gjennomføringen av prosjektet og den opprinnelige planen for gjennomføringen.

5.1 Evalueringemetode

Gjennom prosjektets løp ble evalueringer utført på grunnlag av metodene diskutert i kapittel 3.5 (evalueringsplan). Disse evalueringene tillot en kontinuerlig mottakelse av tilbakemeldinger under utviklingsfasen, noe som ga mulighet for å justere funksjonalitet, rette opp feil og håndtere problemer i sanntid. Dette styrket den inkrementelle og iterative utviklingsprosessen, samtidig som det ga rom for nødvendige tilpasninger i forhold til prosjektets tidsrammer og frister. Evalueringen ble ytterligere utvidet ved innføringen av brukerundersøkelser, som inkluderte brukertesting da applikasjonen nådde et mer avansert utviklingsstadium. Prosessen bak brukerundersøkelsene vil bli ytterligere beskrevet under.

5.1.1 Iterativ tilbakemeldingsprosess

I løpet av utviklingsfasen ble det avholdt ukentlige møter med oppdragsgiver for å verifisere fremdriften og planlegge videre arbeid. Dette ga muligheter for løpende tilbakemeldinger om prosjektets retning og for å få innsikt fra oppdragsgivers utviklere ved eventuelle utfordringer. Gjennom bruk av React, var applikasjonen kontinuerlig i en kjørbær tilstand, noe som tillot konstant testing av implementerte funksjoner og sikret en iterativ tilbakemeldingsprosess.



Figur 5-1 Utforming av applikasjon under testing

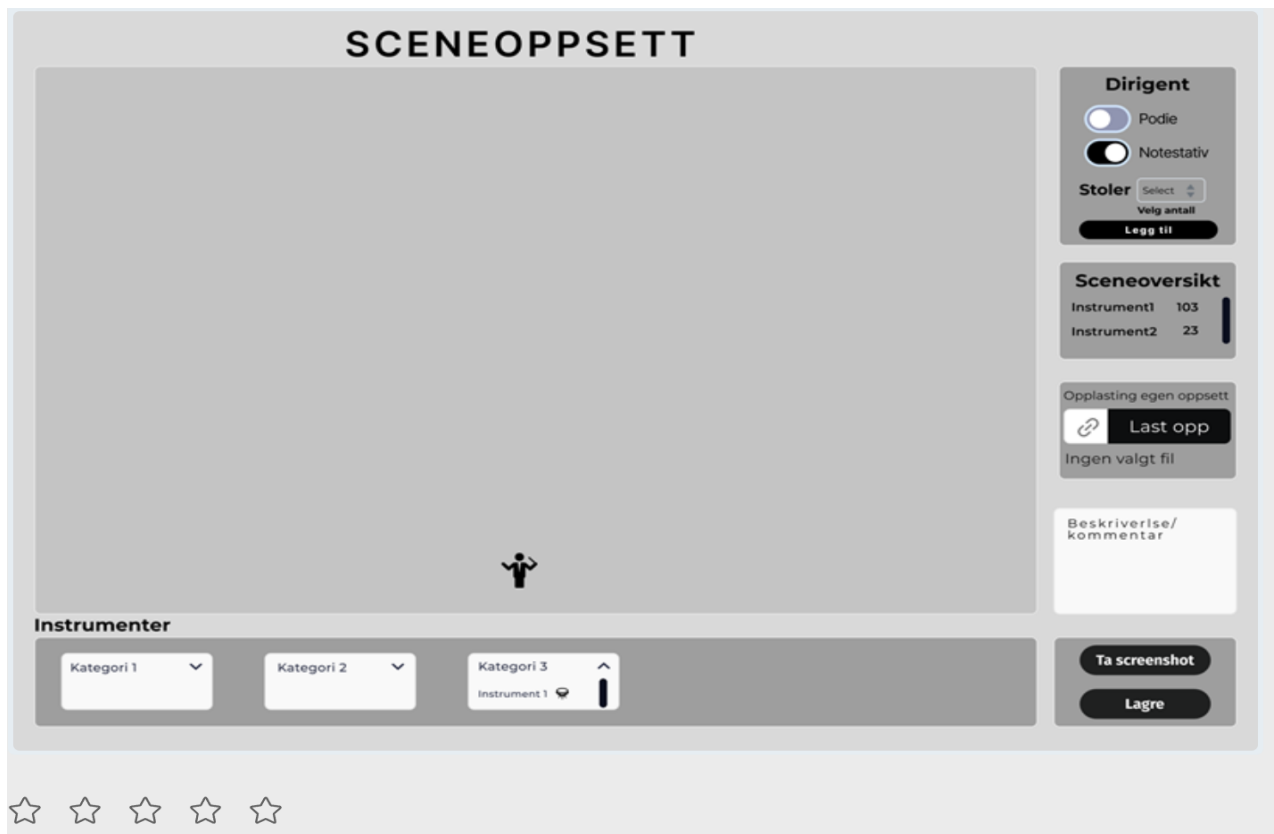
Figuren ovenfor er illustrasjon av applikasjonens utseende, tilgjengelig for oppdragsgiver og senere deltakere i brukerundersøkelse og -testing. Nye funksjoner ble umiddelbart reflektert i applikasjonen, opprettholdt i en konstant kjørbar tilstand.

Videre ble det utført pilottester med studenter fra samme årskull for å samle tilbakemeldinger. Disse testene var avgjørende for å få innsikt i brukernes perspektiv og respons på applikasjonen. Ved utvikling av en applikasjon er det en risiko for tunnelsyn, ettersom utvikleren har inngående kjennskap til applikasjonens funksjon og bruk. Pilottestene bidro til å avdekke aspekter som kan være utfordrende å oppdage i utviklingsfasen.

5.1.2 Brukerundersøkelser og brukertesting

I løpet av utviklingsprosessen ble det gjennomført brukertesting, som ble diskutert tidligere i kapittel 3.5 (evalueringsplan), for å evaluere både funksjonalitet og brukergrensesnittet av applikasjonen. Verktøyet Microsoft Forms (Microsoft, 2023e) ble benyttet for å lage et vedlegg 10.8 (brukerundersøkesskjema) og distribuere det til potensielle testdeltakere. Skjemaet ble sendt til oppdragsgiver, som deretter formidlet det videre. Microsoft Forms samlet inn svar og genererte diagrammer for rangeringsspørsmål, noe som forenklet analysen av brukertilbakemeldinger og bidro til forbedringer og videre utvikling av prosjektet.

Brukertesting for dette prosjektet ble strukturert i to deler. Den første delen fokuserte på designelementene i applikasjonen, hvor brukerne ble presentert for flere designalternativer og ble bedt om å kommentere på disse. Målet var å evaluere plasseringen av ulike komponenter på skjermen for å identifisere det mest hensiktsmessige brukergrensesnittet og arbeidsflyten. Deltagere var bedt om å sette spesielt fokus på arbeidsflyten under evalueringen. Følgende eksempel er en av alternativene som var presentert for deltagere av brukerundersøkelsen og -testing, resterende alternativer er vedlagt under prosjektets vedlegg 10.3 (kravdokument).



Figur 5-2 Eksempel på designalternativ (utformet i Figma)

Etter rangering av designalternativer var deltagere stilt følgende spørsmål:

1. Hvilket oppsett likte du best? Hvorfor? (Skriv nummeret på det oppsettet du likte best)
2. Har du ytterligere ideer eller betraktninger angående plassering av komponentene som ikke er representert i alternativene ovenfor?
3. Er det et bestemt fargevalg og dimensjon for tekst, komponenter og sceneområdet som kunne bidra til en mer behagelig og effektiv brukeropplevelse med applikasjonen? Ville det være ønskelig å kunne tilpasse disse aspektene etter individuelle preferanser?

I den andre delen fikk deltagere tilgang til en demoversjon av applikasjonen illustrert i (figur 5-1) som var det faktiske brukertesting hvor de kunne teste nåværende funksjoner og deretter gi tilbakemelding på deres opplevelse. Målet med denne delen var å undersøke om det var enkelt eller utfordrende for deltagere å forstå hvordan de skulle samhandle med applikasjonen.

Testingen ble gjennomført som umoderert fjernstyrt testing, der deltagerne individuelt gjennomførte testingen uten direkte veiledning. Testingen ble utført over nett og deltagerne hadde frihet til å gjennomføre testen når det passet for dem. Applikasjonen som ble benyttet til brukertesting var i nærheten av å være funksjonelt sett komplett (Justinmind, 2019).

Hoveddeltakerne i brukerundersøkelse og -testing var oppdragsgivers kunder, hovedsakelig NMF. Som primære brukere av applikasjonen, ga de verdifulle tilbakemeldinger. Dette ga innsikt i hvordan brukerne ønsket at applikasjonen skulle se ut.

I sin helhet bidro brukertestene til å gi innsikt i brukernes forventninger og preferanser, bidra til å bestemme applikasjonens utseende og funksjonalitet, og avdekke eventuelle feil som krevde korrigering.

Spørsmålene som ble stilt til demoversjonen av applikasjonen inkluderte blant annet vurdert hvor intuitiv applikasjonen var i bruk, hvor enkelt det var å finne funksjoner, og eventuelle forbedringer deltakerne ønsket å se i den fremtidige versjonen av applikasjonen. Til demoversjonen ble disse syv spørsmålene stilt:

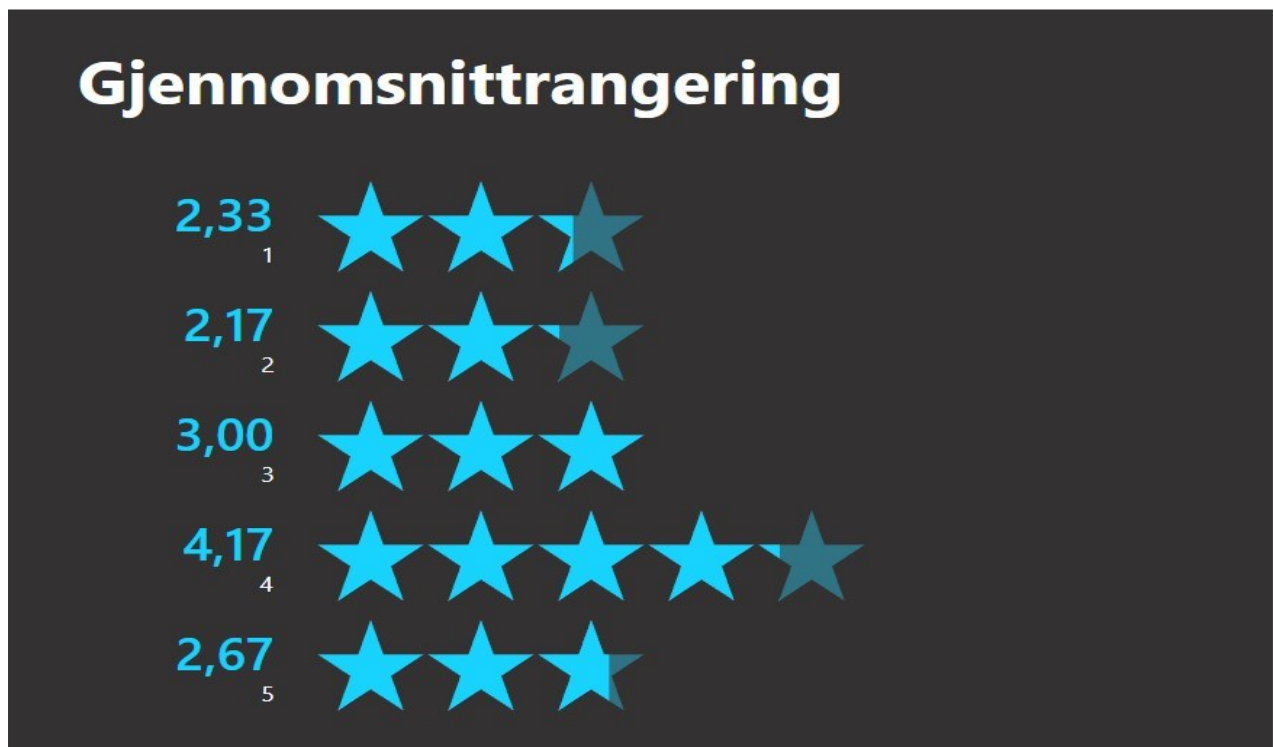
1. Hva var din første tanke da du åpnet appen?
2. Følte du at appen var intuitiv å bruke?
3. Var det noen funksjoner som du ikke kunne finne eller var vanskelig å finne?
4. Hva synes du om funksjonene i applikasjonen?
5. Fikk du noen feilmeldinger eller problemer mens du brukte appen?
6. Var det enkelt å fullføre handlingene du ønsket å gjøre i applikasjonen?
7. Er det noen funksjoner eller endinger du ville ønske var inkludert i appen for å forbedre opplevelsen?

5.2 Evalueringsresultat

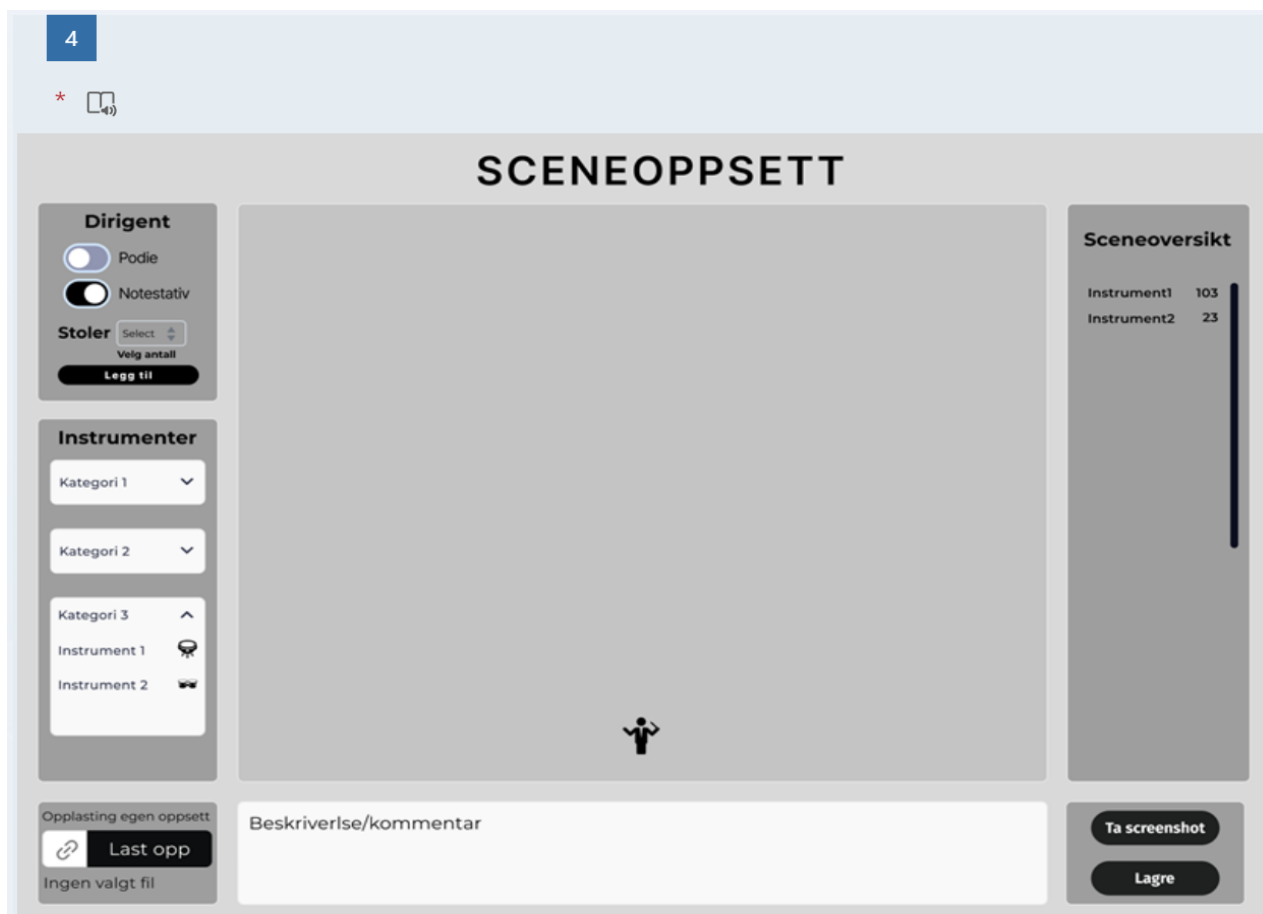
Fra evalueringen gjennomført under prosjektets gang, ble det samlet inn tilbakemeldinger fra brukerundersøkelse og -testing både fra deltagere og oppdragsgiver, disse blir beskrevet videre i underkapittel.

5.2.1 Brukerundersøkelser og brukertesting

På grunn av begrensninger i utviklingstid, ble det bare mulig å gjennomføre en runde med umoderert fjernstyrt brukertesting. Denne tilnærmingen viste seg å være den mest realistiske med tanke på tidsrammen. Resultatene fra brukertesting ble deretter integrert i den endelige fasen av utviklingsprosessen før overlevering. Totalt ble det samlet inn seks tilbakemeldinger i løpet av en uke. De mest bemerkelsesverdige funnene inkluderte at designalternativ to mottok den mest negative responsen, mens alternativ fire ble best mottatt. Dette kan observeres i følgende oppsummering fra responsene.



Figur 5-3 Oppsummering av respons fra brukerundersøkelse



Figur 5-4 Oppsett nr.4 fra brukerundersøkelse

Det oppsettet som fikk mest positiv respons i undersøkelsen var kjennetegnet av en mer naturlig lesbarhet, med instrumentpanelet plassert på venstre side, hvor instrumenter kan dras fra. I tillegg ble innstillingene for canvasområdet også plassert på venstre side.

Responsen på testdelen av undersøkelsen var variert. Selv om det ble presisert at fokuset ikke var på designet, men funksjonaliteten var likevel det noen deltakere som kommenterte på design og fargevalg.

Mesteparten av responsen indikerte at det var enkelt å finne funksjonene og opprette et sceneoppsett. Noen tilbakemeldinger avdekket feil, som for eksempel at for mange stoler overskred grensene for canvasområdet. Disse tilbakemeldingene ble tatt i betraktning i utbedringen av den endelige applikasjonen før overlevering, som blir presentert i prosjektresultatet. Fullstendig resultat av brukerundersøkelse og -testing finnes under vedlegg 10.9 (resultat av brukerundersøkelse og -testing).

5.2.2 Sluttevaluering av prosjektet

Brukerundersøkelse og -testing, ble ytterlig supplert med en sluttevaluering fra oppdragsgiver, som tidligere diskutert i kapittel 3.5 (evalueringsplan). Oppdragsgivers tilbakemelding antyder tilfredshet med prosjektets resultater, noe som betraktes som en indikasjon på at prosjektet generelt sett nådde sine mål.

Det ble likevel identifisert enkelte områder for forbedring. En av disse er kontrasten i applikasjonen, der en økning kan forbedre lesbarheten. Det ble også foreslått å utvide funksjonaliteten for valg av instrumenter på canvasområdet.

En potensiell utfordring er at når stoler legges til canvas, kan disse ende opp stablet oppå hverandre hvis brukeren gjentatte ganger benytter funksjonen uten å flytte tidligere plasserte stoler. Dette ble anerkjent som et område for potensiell forbedring.

På den positive siden ble det gitt ros til oversiktsfunksjonen som viser hvilke elementer som er til stede på canvasområdet til enhver tid. Imidlertid ble det foreslått en justering av prioritering når det gjelder rekkefølgen på elementene i denne listen, hvor oppdragsgiver vil ha stoler først.

Selv med erkjennelsen av nevnte forbedringsområder, ansees prosjektet som en stødig plattform for videre utvikling. I rapportens etterfølgende deler vil det bli utført en mer dyptgående drøfting av disse punktene. Dette vil være basert primært på prosjektets resultater og oppdragsgivers sluttevaluering.

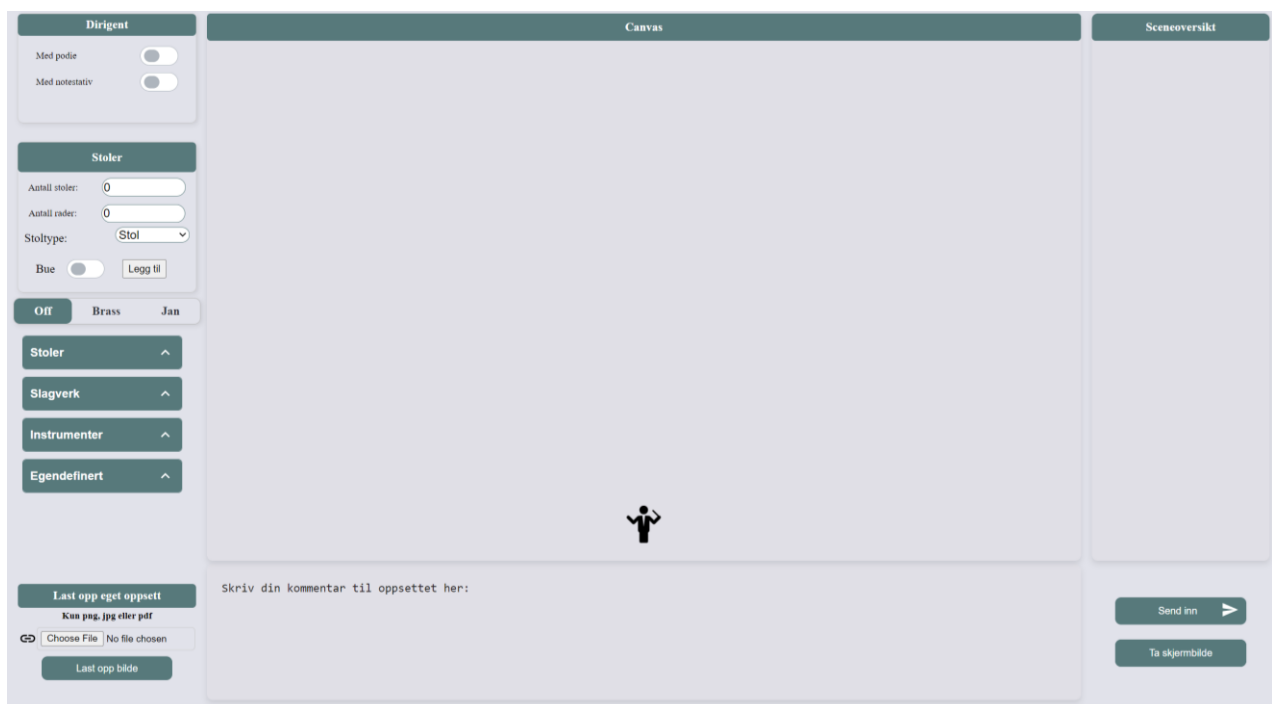
En fullstendig evaluering fra oppdragsgiver er inkludert under sluttevaluering fra oppdragsgiver vedlegg 10.7 (sluttevaluering fra oppdragsgiver).

5.3 Prosjektresultat

Prosjektets endelige resultat er en komplett webapplikasjon for utforming av sceneoppsett, lagret på GitHub lagringssted, som finnes under Vedlegg 10.5 (GitHub-lagringssted for prosjektet). I denne delen av rapporten vil det bli gitt en oppsummering av både de oppfylte funksjonelle og ikke-funksjonelle kravene, samt en gjennomgang av hvordan det endelige designet ble formet. Videre vil det bli beskrevet prosedyrene for overlevering og dokumentasjon av applikasjonen.

5.3.1 Design av ferdigstilt applikasjon

Det endelige designet av applikasjonen ble utformet i samsvar med tilbakemeldingene mottatt under brukerundersøkelsen og testrunden som ble gjennomført. Plasseringen av komponentene samsvarer med det designet som mottok de mest positive tilbakemeldingene fra undersøkelsen, og at fargevalget har blitt ytterligere forbedret. Imidlertid ble den lysgrå bakgrunnen, til tross for tidligere kommentarer, opprettholdt for å sikre et mer behagelig fargevalg for øyet. På tross av dette, sørger kontrasten for at det ikke er noen indikasjoner på lav kontrast ved bruk av verktøyet Accessibility Insights for Web, som først ble nevnt i kapittel 4.2 (brukergrensesnitt).






































Figur 5-5 Design av ferdigstilt applikasjon

Det endelige designet av applikasjonen har generelt mottatt positive tilbakemeldinger under møte med oppdragsgiver, spesielt med tanke på komponentplassering og arbeidsflyten i applikasjonen. Videre kommer det frem at funksjoner anses som lett tilgjengelig, og designet oppfattes som moderne og tilfredsstillende, noe som igjen blir bekreftet i den endelige tilbakemeldingen fra oppdragsgiver. Selv om det fortsatt er rom for forbedring, anses det overordnede designet og arbeidsflyten som mer enn tilstrekkelig i forhold til forventet resultat.

5.3.2 Funksjonelle og ikke funksjonelle krav

Gitt prosjektets prioriteringer og tidsbegrensninger, ble ikke en database implementert, selv om applikasjonen er utformet på en måte slik at implementeringen databasen kan gjøre ved senere anledning. Oppdragsgiver har uttrykt at en database ikke er en høy prioritet på dette stadiet og at tilfredsstillelse ville oppnås om applikasjonen produserte en klar JSON-streng for oppsettet, slik at en database lett kan legges til i fremtiden. Dette førte til at det ble tatt skritt mot å legge til en lagringsfunksjon for sceneoppsett, men denne ble ikke fullført. Imidlertid ble det implementert en funksjon som konverterer sceneoppsettdata til en JSON-streng og omvendt. Utfra dette kan det konkluderes med at store deler av denne funksjonen er allerede på plass og gir et godt grunnlag for videre utvikling. Språkstøtte ble vurdert som et krav med lav prioritet fra prosjektets start, og det var derfor naturlig å utsette dette kravet når tiden ble begrenset.

Tabell 5-1 Oversikt over implementering av funksjonelle krav

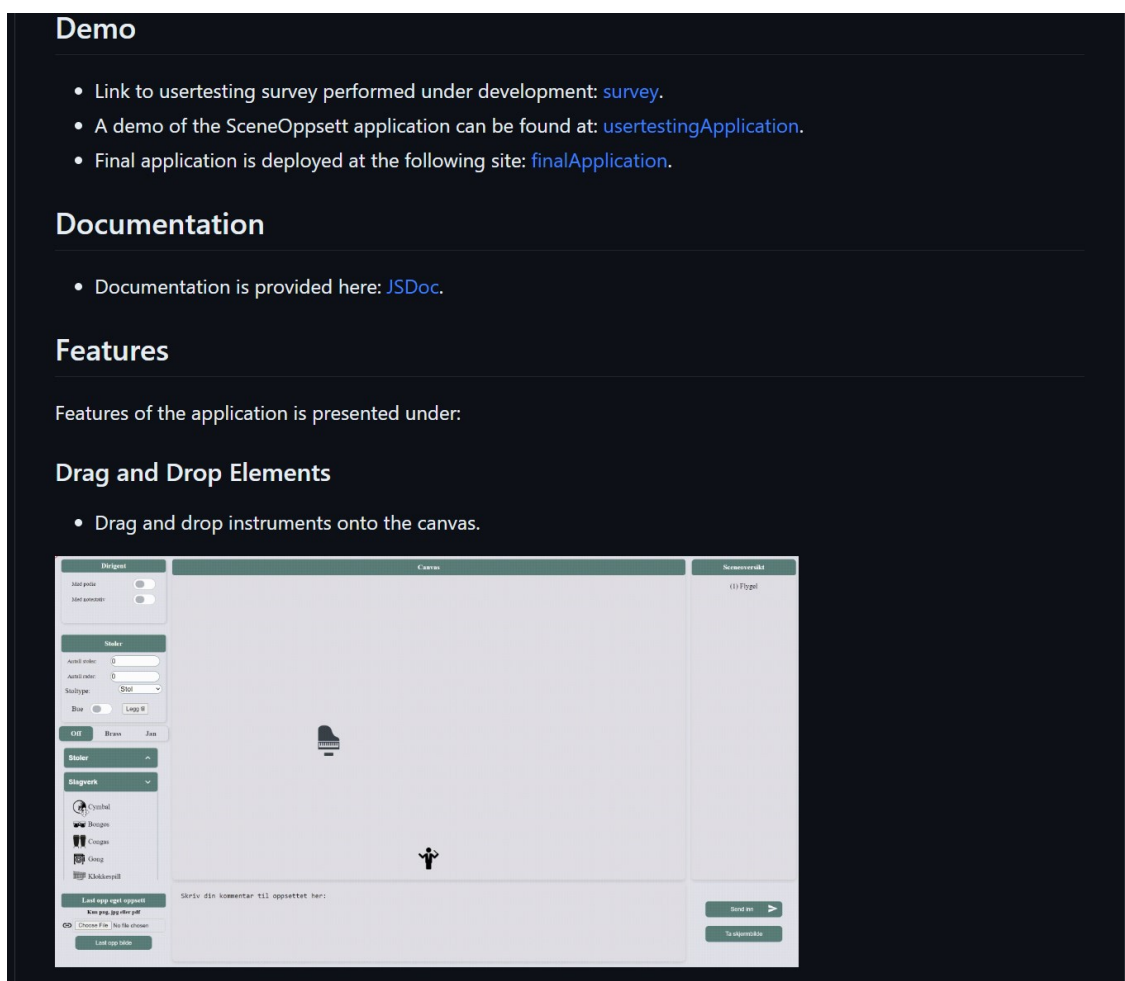
Krav	Utsatt/påbegynt	Ikke fullført/delvis/fullført		
Antall stoler				
Lagring av sceneoppsett				
Plassering av instrumenter på canvas				
Kommentar				
Dirigentinnstillinger				
Oversikt				
Navn på instrument				
Oppsettguide				
Filopplasting av eget sceneoppsett i form av PDF eller bilde				
Ikoner skal være fastbestemt størrelse				
Skalering ved endring i skjermstørrelse				
Legge til stoler				
Beskrivelse av dirigent				
Språkstøtte				
Egendefinerte objekter				
Fullført: 	Påbegynt: 	Delvis: 	Utsatt: 	Ikke fullført: 

Tabellen ovenfor illustrerer at majoriteten av de initialt fastsatte krav ble igangsatt i løpet av utviklingsfasen, og kun to av de opprinnelige kravene forble ikke-implementert. I løpet av prosjektets forløp ble det ikke innført ytterligere krav. Imidlertid, ble det identifisert flere potensielle løsninger for noen av de eksisterende kravene. Disse alternativene ble grundig diskutert for å avgjøre hvilken løsning som ville være mest passende. Et eksempel på dette er funksjonen for å legge til stoler, som ble utvidet for å inkludere evnen til å arrangere stoler i buer.

I forhold til de ikke-funksjonelle kravene ble størstedelen også oppfylt i løpet av prosjektet. Teknologivalget for utvikling ble fastsatt av oppdragsgiver og ble anvendt konsekvent gjennom hele prosjektets løp. Til tross for dette, ble ikke en komplett databaseløsning implementert, noe som betyr at dette spesifikke kravet ikke ble fullt oppfylt. Samtidig ble kravet om modernisering av designet oppfylt. Applikasjonen fremviser et mer moderne design og tilbyr en forbedret arbeidsflyt i forhold til den tidligere løsningen, noe som ble bekreftet i de avsluttende møtene med oppdragsgiver.

5.3.3 Overlevering og dokumentasjon

Under et møte med oppdragsgiver ble det avtalt tidspunkt for overlevering av applikasjonen. Overleveringen skjer ved å legge til oppdragsgiver som samarbeidspartner på GitHub-lagringssted. Dermed får oppdragsgiver full tilgang til all kildekode, samt dokumentasjon i form av JSDoc-dokumentasjon, tidligere diskutert i kapittel 3.4.1 (utviklingsmetodikk). JSDoc-dokumentasjonen er også satt opp som en nettside hvor all dokumentasjonen som er utført i koden ligger, samt en enkelt navigering gjennom den. Videre beskrivelse av dokumentasjonen finnes i vedlegg 10.4 (systemdokumentasjon). En videodemonstrasjon av applikasjonen ligger i vedlegg 10.6 (videodemonstrasjon av applikasjonen).



Figur 5-6 Utklipp fra README til GitHub lagringssted

I illustrasjonen ovenfor er det mulig å observere hvordan dokumentasjonen er utformet på GitHub-lagringssted. Det er lagt ved lenker til både brukerundersøkelsen og applikasjonen som

ligger ute - både versjonen som ble brukt for testing og det ferdige produktet. Hver funksjon er demonstrert ved hjelp av Graphics Interchange Format⁷ (GIF) og en beskrivelse av funksjonen.

Sceneoppsett documentation

Members

(constant) `DataContextProvider`

`DataContextProvider` is a React component that leverages the Context API to provide data to all the components in its subtree (children). The provided data includes an array of instruments, a description string, and a `globalSceneSize` object.

The component uses the `useState` hook to manage the state of these values. The state and its updater function are grouped into a values object, which is passed to the `DataContext.Provider` as a value prop.

All children components wrapped inside `DataContextProvider` will have access to these state values and updater functions, allowing them to consume and modify this shared state if necessary.

Source: [DataContextProvider.js, line 33](#)

Example

```
// Wrap your component tree with DataContextProvider to provide the context values to all components in the tree.
<DataContextProvider>
  <YourComponent />
</DataContextProvider>
```

(constant) `ImageDataRenderList` :Array.<{id: number, image: string, name: string, group: string, width: number, height: number}>

`ImageDataRenderList` is a collection of objects, where each object represents an image asset. Each object in the array contains information about an image such as its source path (image), its name (name), its group (group), and its dimensions (width and height). This data is used to render images with specified attributes in the application.

- Home
- Global
- App
- CanvasSummary
- Category
- ChairOptions
- CustomSceneUpload
- DataContextProvider
- DeleteArea
- Description
- Dirigent
- DirigentOptions
- Guide
- GuideOptions
- Helpers
- ImageDataRenderList
- InstrumentList
- Scene

Figur 5-7 Utklipp fra JSDoc applikasjons dokumentasjon

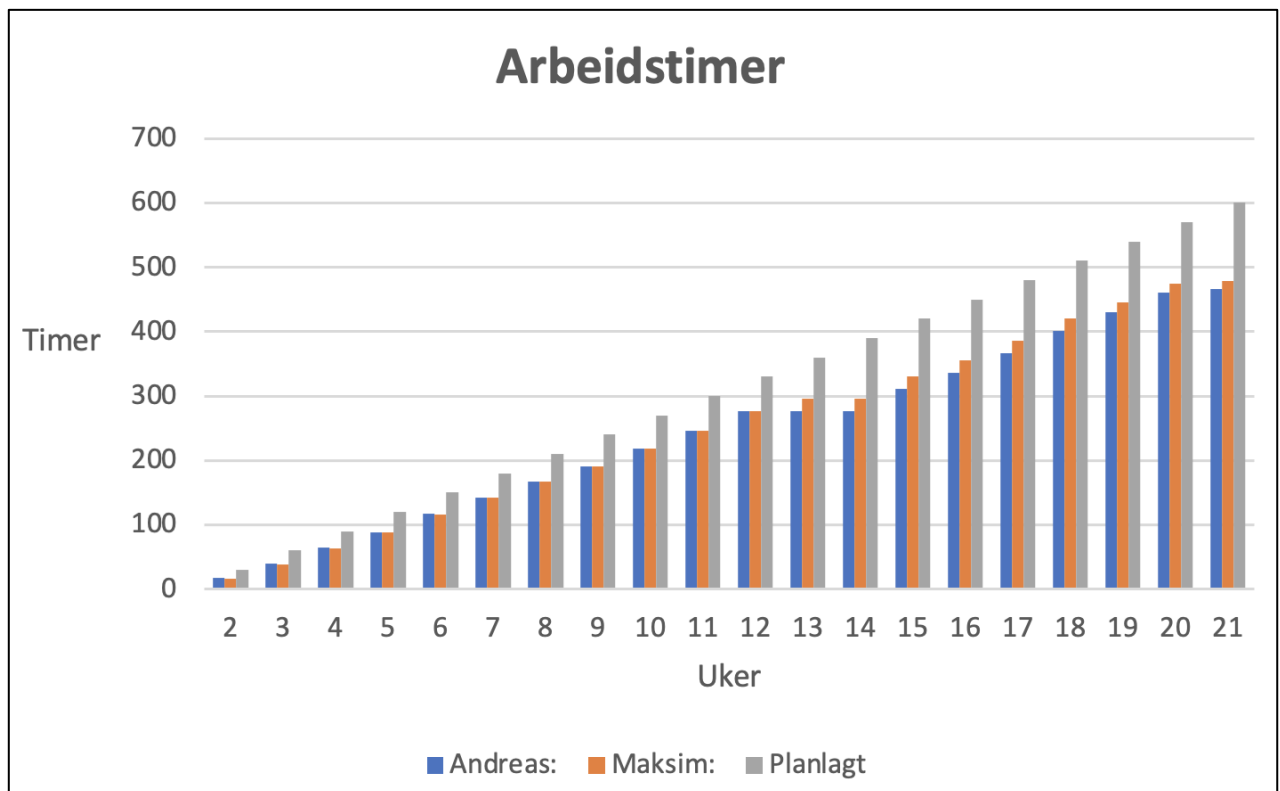
Følgende utklipp er fra JSDoc-dokumentasjonen av applikasjonen Sceneoppsett, referert til under README-dokumentet som er tilgjengelig via GitHub-lagringssted. Nettsiden til JSDoc-dokumentasjonen er gjort tilgjengelig via (Microsoft, 2023a), en funksjon i GitHub som lar brukere gjøre tilgjengelige sider som er lagret i et prosjektarkiv. For prosjektevaluering er det gjort tilgjengelig en midlertidig lenke under vedlegg 10.5 (GitHub lagringssted for prosjektet) som gir lesetilgang til dette prosjektarkivet.

⁷ Grafisk filformat med bevegelige bilder

5.4 Prosjektgjennomføring

Prosjektets plan ble presentert i kapittel 3.4.2 (fremdriftsplan), hvor Gantt-diagrammet ble benyttet som verktøy til prosjektplanlegging. Imidlertid ble det i løpet av prosjektets fremdrift erstattet med bruk av Kanban, som er beskrevet i kapittel 4.3.3 (kanban), for å opprettholde oversikt over arbeidsoppgaver. Kanban ble valgt på grunn av dens evne til å lette utviklingen og dens høye grad av dynamikk. Ved å benytte Kanban kunne oppgaver enkelt tildeles, og statusen til hver oppgave kunne enkelt kontrolleres. Dette ga en mer fleksibel tilnærming til prosjektgjennomføringen, der oppgaver tildeles og håndteres på en mer adaptiv måte underveis i prosjektet.

Under er et diagram som illustrerer sammenhengen mellom planlagte og faktiske arbeidstimer. Den planlagte arbeidstiden var satt i øverste laget med 600 timer per person. Basert på grafen fremstår det som de planlagte timene i stor grad ble fulgt med jevn arbeidsinnsats gjennom prosjektet. Unntaket er uke 13 og 14, der færre timer ble registrert på grunn av påskeferie.



Figur 5-8 Arbeidstimer i forhold til planlagte timer

6 Diskusjon

Dette kapitlet analyserer prosjektets resultat og identifiserer årsaker til avvik fra planene og målene. Valg og tilnærminger gjennom prosjektet påvirket det endelige resultatet, det inkluderer både gode og dårlige valg. Fokuset er på å vurdere konsekvensene av disse valgene og deres innvirkning på resultatene, samtidig som alternative valg som kunne føre til forbedringer utforskes. Oppdragsgivers forventinger i forhold til oppfyllelse blir vurdert. Kapitlet identifiserer prosjektets styrker og svakheter, og analyserer samfunnsmessige og økonomiske påvirkninger. Til slutt blir det identifisert deler som kunne vært forbedret i prosjektet.

6.1 Tekniske valg

Det ble gjort en rekke tekniske valg under utviklingen. Disse bli diskutert i følgende underkapittel.

6.1.1 Valg av verktøy

Et kritisk valg som ble tatt gjaldt implementeringen av dra-og-slipp-funksjonaliteten. På grunn av denne funksjonalitetens sentrale betydning i applikasjonen, var det nødvendig å implementere den tidlig i prosjektet. Dette krevde en tidsinvestering for å identifisere det mest passende verktøyet innenfor React-biblioteket. En rekke tester ble utført for å vurdere potensielle verktøy som kunne oppnå den ønskede funksjonaliteten.

Opprinnelig ble det valgt en kombinasjon av React DnD og React Draggable. Dette viste seg å være et dårlig valg ettersom det ble besluttet å bytte til en annen kombinasjon av verktøy, nemlig React DnD sammen med Moveable, da disse ble ansett som bedre egnet for denne funksjonaliteten.

For å implementere funksjonaliteten for valg og flytting av flere instrumenter på canvas, ble verktøyet Selecto benyttet. For å kunne lage skjermbilder av sceneoppsettet, ble verktøyet html2canvas benyttet.

Alle de nevnte verktøyene ble installert ved hjelp av Node Package Manager (NPM). NPM er en programvarepakkehåndterer for JavaScript, og brukes til å administrere avhengigheter og distribuere pakker i Node.js, som er brukt i denne React applikasjonen.

6.1.2 Sikkerhetsproblem

En betydelig, men nødvendig endring ble foretatt for å takle utfordringene knyttet til bruk av "innerHTML()" og "querySelector()" i React for endring av ulike elementer. Begge disse metodene utgjorde potensielle sikkerhetsrisikoer. Bruk av "innerHTML()" kunne gi opphav til XSS (Cross-Site Scripting) sårbarheter, ettersom den tillater dynamisk oppdatering av nettshedsinnhold og muliggjør innføring av skript i inndatafelt. "querySelector()" kunne potensielt velge nettsheds elementer ved hjelp av CSS-selektorer, og uten tilstrekkelig filtrering av inndata kan dette resultere i uønsket kodeutførelse. I tillegg kan direkte tilgang til Document Object Model⁸ (DOM), uten bruk av Reacts virtuelle DOM, forårsake ytelsesproblemer.

For å håndtere disse sikkerhetsutfordringene, ble det brukt Reacts API, som inkluderer funksjonaliteter som "states" for oppdatering av tilstand, og "refs" for å referere til elementer på nettstedet og oppdatere DOM. Implementeringen av denne løsningen krevde en dyptgående forståelse av Reacts API. Selv om dette innebar en betydelig tids- og innsatsinvestering, resulterte endringen i en mer robust, effektiv og dynamisk applikasjon.

Et sentralt element i denne utfordringen var det begrensede kjennskapet til og erfaringen med React, ettersom den eksisterende kompetansen primært var relatert til HTML og konvensjonell JavaScript. Dette medførte en lengre innlæringskurve for å tilegne seg kunnskap om Reacts API og tilpasse seg dets bruk.

6.1.3 Database

På grunn av tidsbegrensninger ble ikke implementeringen av databasen gjennomført innenfor prosjektets tidsramme. Imidlertid ble applikasjonen konstruert med tanke på en fremtidig integrering av en database, idet all data som skal lagres blir generert som en JSON-streng og vist i konsollen. Prioriteringene i prosjektet lå ikke i implementeringen av en database, ettersom det fantes andre funksjonelle krav som ble ansett som viktigere. Oppdragsgiver har kapasitet til å håndtere implementeringen av databasen selv. Til tross for manglende implementering av en database, gjøres nettsiden fortsatt tilgjengelig via Firebase-plattformen.

⁸ Programmeringsgrensesnitt for å få tilgang til og manipulere strukturen til nettside dokumenter

6.2 Design valg

I dette kapitlet blir de viktigste designvalgene beskrevet sammen med konsekvensene og løsningene på problematiske valg.

6.2.1 Skalering av applikasjonen

Implementeringen av skalerbarhet og responsiv design i applikasjonen utgjorde en betydelig og utfordrende beslutning. Denne funksjonaliteten viste seg å være et av de mest krevende ved hele utviklingsprosessen. For å realisere dette ble det nødvendig å foreta en omfattende restrukturering av prosjektets oppsett, inkludert en grunnleggende gjenoppbygging av applikasjonen. Til tross for at denne prosessen ble påbegynt noe ut i utviklingsfasen, kunne en betydelig andel av den eksisterende koden gjenbrukes og integreres i det nye prosjektoppsettet. Implementeringen av skalerbarhet og responsivitet krevde betydelige tidsressurser og innsats for å oppnå de ønskede resultatene. Selv om oppdragsgiveren tilsynelatende ikke mente dette aspektet skulle prioriteres, betraktet utviklerne ved prosjektet det som avgjørende for å forbedre applikasjonens brukeropplevelse og funksjonalitet. Dette medførte også at instrumentenes koordinater på scenen måtte omregnes til prosent for å sikre kompatibilitet for alle skjermstørrelser.

6.2.2 Fargevalg

Basert på tilbakemelding fra oppdragsgiver, som er diskutert i kapittel 5.2.2 (sluttvurdering av prosjektet), ble det pekt på at fargekontrasten mellom scenen og resten av nettsiden var noe svak. Dette kan potensielt utgjøre en utfordring for personer med nedsatt synsevne. Imidlertid kan dette korrigeres ved å justere en enkel farge. Denne observasjonen kunne trolig vært identifisert tidligere hvis det hadde blitt gjennomført flere brukertester. Ved dyp involvering i utviklingsprosessen kan det være lett å overse slike detaljer. Gjennom å utføre en mindre designtest før overlevering, kunne dette problemet sannsynligvis vært unngått.

6.2.3 Videre design arbeid

Ved mer tilgjengelig tid i prosjektets sluttstadium, kunne finjustering av CSS potensielt vært utført. Den fastsatte overleveringsfristen resulterte i manglende tid til å gjennomføre siste

innstillinger, gitt at tid også var reservert for rapportskriving. I denne konteksten kunne ytterligere brukertesting vært nyttig, ettersom det kan bidra til å identifisere eventuelle justeringsbehov.

6.3 Evaluering

Det eksisterer forskjellige tilnærminger til evaluering. Dette kapittelet utforsker valgte evalueringmetoder, diskuterer suksessfaktorer og identifiserer områder for forbedring.

6.3.1 Oppdragsgiver

Planlagte møter med oppdragsgiver var satt til å avholdes hver fredag kl. 10:00. Imidlertid, på grunn av oppdragsgivers andre forpliktelser og situasjoner hvor det ikke hadde vært vesentlige oppdateringer siden forrige møte, ble ikke alltid disse møtene gjennomført som planlagt. Møtene ble vurdert som viktige for å sikre korrekt prosjektutvikling. Hyppige og regelmessige møter spilte en avgjørende rolle i å fremme prosjektets fremdrift, da det var et mål å demonstrere fremgang fra møte til møte.

Fordelen med regelmessige møter ligger i muligheten for oppdragsgiver til å gi hyppige tilbakemeldinger og foreslå nye ideer som kan forbedre applikasjonen. Det er likevel viktig å opprettholde en balanse, da implementeringen av alle tilbakemeldinger og ytterlige forbedringer kan føre til kontinuerlig utvidelse av prosjektet. Dermed ble det etablert en grense for utvikling som fastsatte at ingen ytterligere arbeid skulle utføres etter prosjektets overlevering. Selv om de nye forbedringene kunne være relevante, ble hovedfokus rettet mot de opprinnelige kravene.

6.3.2 Veileder

Møtene med veileder ble gjennomført på en fleksibel måte etter behov, frekvensen av disse møtene var ikke like hyppig som møtene med oppdragsgiver. Denne måten var passende, da den tidlige fasen av prosjektet var preget av fokus på utviklingsarbeidet fremfor skrivearbeidet i rapporten. Når tiden nærmet seg for innlevering av rapportutkastet, ble det imidlertid vurdert som hensiktsmessig å avholde flere møter med veilederen for å sikre at rapporten var i tråd med forventningene og at den gikk i riktig retning.

6.3.3 Brukertesting

Som omtalt i kapittel 5.1.2 (brukerundersøkelser og brukertesting), ble brukertesting implementert gjennom en runde av fjernstyrt, umoderert testing. Ideelt sett kunne det vært hensiktsmessig å utføre flere runder av brukertesting, som inkluderer modererte økter for å tillate direkte dialog med deltakerne og håndtere eventuelle spørsmål som måtte oppstå. Selv om moderert testing ville ha fordelene av direkte interaksjon, tilførte den valgte testmetoden verdifull innsikt for prosjektet. Oppdragsgiveren var ansvarlig for å distribuere testen til potensielle brukere, gitt deres eksisterende kontaktnettverk med tidligere brukere av applikasjonen. Selv om testen ble distribuert til et utvalg individer, var responsen begrenset, med bare seks innsendte svar.

Den begrensede responsen på brukertesting, som inneholdt kun seks deltakere, er generelt utilstrekkelig for å representere applikasjonens faktiske brukergruppe. For å stimulere til større deltagelse i brukertesten og avverge lav responsrate, kunne alternative valg vært gjort. For eksempel, kunne brukertesten vært igangsatt tidligere i utviklingsprosessen, noe som ville gitt brukerne mer tid til å delta. Dette ville også ha åpnet for muligheten til å gjennomføre en ny runde med brukertesting for å evaluere responsen etter implementering av endringer basert på første runde testresultater. Økt distribusjon av testen og påminnelser kunne begge ha stimulert til større respons. Det bør bemerkes at selve brukertesten var noe tidkrevende, og det er mulig at flere deltakere ville vært villige til å delta dersom testen hadde vært mindre omfattende.

6.4 Forskningsspørsmål

Forskingsspørsmålet som er omtalt i kapittel 1.4 (problembeskrivelse, mål og forskningsspørsmål) tar for seg visjon om å forbedre applikasjonens brukeropplevelse, redusere drifts- og vedlikeholdskostnader og øke kundetilfredsheten. Ved å analysere resultatene av prosjektet kan det observeres en god flyt i applikasjonen, som også ble bekreftet gjennom brukertesting og tilbakemeldinger fra oppdragsgiver. Dette kan være grunnet effektiv organisering av instrumenter i lister, noe som bidrar til brukervennlighet. En betydelig designfeil i oppdragsgivers tidligere løsning var strukturen på kontrollpanelet. Alle funksjonelle innstillinger og instrumenter var plassert i en lang liste som krevde vertikal rulling, noe som gjorde justering av innstillinger og innsending av oppsett utfordrende.

Når det kommer til minimering av drifts- og vedlikeholdskostnader, viste det seg i praksis å ikke være av så stor betydning som tidligere antatt. Den økonomiske forskjellen var ikke en kritisk faktor for oppdragsgiver i avgjørelsen om å oppgradere applikasjonen. Ved konsekvent bruk av samme type tjenester og programmeringsspråk gjennom hele øvresystemet som applikasjonen skal integreres i, unngås ytterligere vedlikeholdskostnader knyttet til koden, tilgjengeliggjøring av applikasjonen og databaseløsninger. Applikasjonen, som er utviklet i React, tilgjengeliggjøres via Firebase-plattformen, og en potensiell senere implementering av databaseløsning via Cloud Firestore, kan oppdragsgiver vedlikeholde prosjektet internt.

Resultatene fra brukertestene indikerer en høy grad av kundetilfredshet, noe som kan ansees som en indikasjon på at prosjektet var suksessfullt. Gjennom testingen ble det også identifisert noen områder hvor brukerne ikke var helt fornøyde, og denne tilbakemeldingen ble tatt i betraktning for videreutvikling av applikasjonen.

Applikasjonen gir en forbedret brukeropplevelse og økt tilgjengelighet for brukerne. Ved bruken av valgte teknologier og metodikker, adresserer dette prosjektet viktige aspekter som produktivitet, brukerengasjement og kostnadsreduksjon. Videre kan den forbedrede brukeropplevelsen gi økt brukerengasjement og tilfredshet, noe som potensielt kan medføre en økt bruk av applikasjonen.

Denne samlede effekten av prosjektet kan bidra til å styrke organisasjonens konkurransevne og bærekraft i et stadig mer digitalisert og konkurransepreget miljø. Ved å forbedre brukeropplevelsen og redusere vedlikeholdskostnader bidrar den nye applikasjonen til en mer bærekraftig forretningsmodell, noe som kan ha en positiv innvirkning på organisasjonens langsiktige suksess og vekst. Til slutt viser tilbakemeldingene fra brukertestene et høyt nivå av kundetilfredshet, noe som understreker applikasjonens evne til å møte brukernes behov og forventninger. Denne suksessen viser applikasjonens potensial for å øke bruken av applikasjonen og bruker lojalitet, som igjen kan eventuelt føre til høyere inntekter og større kundebase.

6.5 Forbedringspotensialet

Den underliggende kompleksiteten av applikasjonen er betydelig, til tross for dens tilsynelatende enkelhet. Utviklingen av denne enkle applikasjonen krevde omfattende tid og opplæring. Jo mer sømløs og enkel applikasjonen skal virke for brukerne, desto mer tid måtte dedikeres til grundig planlegging og utvikling. Dersom prosjektet skulle gjøres på nytt, ville det være mulig å fullføre applikasjonen på nærheten av én måned i stedet for fire, da mye av kunnskapen ble opparbeidet mot slutten av utviklingsprosessen. Ved å investere mer tid i å utforske utviklingsverktøyet React og applikasjonsoppsett tidligere i prosjektet, ville det potensielt vært mulig å unngå betydelige justeringer underveis i utviklingen.

I de innledende og mellomliggende fasene av prosjektet, ville det ha vært fordelaktig å etablere faste arbeidsdager og -tider for prosjektet, ettersom det periodevis var utfordrende å koordinere felles arbeidstid. Selv om denne utfordringen gradvis ble overkommet, ville tidlig etablering av slike rutiner vært fordelaktig. Dette kunne ha blitt unngått ved å avtale arbeidsdager og varighet på forhånd.

Det kunne ha vært gunstig å investere mer tid i parprogrammering fremfor individuelt arbeid, ettersom dette ville ha bidratt til en dypere forståelse blant utviklerne. Individuelt arbeid kan ofte føre til misforståelser rundt implementering av endringer mellom utviklerne, og det kan oppstå utfordringer knyttet til versjonskontroll ved sammenslåing av prosjektets ulike deler.

Beslutningen om å benytte Kanban-metoden i utviklingen av prosjektet, fremfor å bruke et Gantt-diagram, viste seg å være gunstig. Kanban-tilnærmingen viste seg å være mer hensiktsmessig som et planleggingsverktøy, gitt sin dynamiske tilnærming og letthet ved fordeling av oppgaver mellom utviklerne.

7 Konklusjon og videre arbeid

Dette kapittelet omhandler konklusjoner fra prosjektet og gir en oversikt over potensielle tiltak for fremtidig videreføring av prosjektet.

7.1 Konklusjon

Dette prosjektet hadde som mål å rekonstruere en eksisterende applikasjon ved å implementere den på nytt ved hjelp av React, med Node og Firebase som underliggende teknologier.

Målet er oppnådd, da oppdragsgiver er tilfreds med resultatet og har planer om å videreutvikle prosjektet for å integrere det i deres system. Dette vil resultere i en enklere applikasjon å vedlikeholde for oppdragsgiver, da deres øvrige løsninger også er basert på React, Node og Firebase. For oppdragsgiver har denne oppgaven stor betydning, og den ville uansett måtte gjennomføres på et senere tidspunkt ettersom mesteparten av deres eldre system skal over til de brukte teknologiene. Derfor var det positivt at den kunne realiseres som en bacheloroppgave. Oppgaven er også realistisk og relevant med tanke på utviklingsarbeidet i arbeidslivet.

I starten av prosjektet var begrensningen av kunnskap om React en utfordring som påvirket fremdriften. Det ble dedikert betydelig tid til å forstå og lære React. Mot slutten av prosjektet begynte forståelsen og mestringen av React å utvikle seg, men dette krevde betydelig innsats og flere forsøk. Prosjektets kompleksitet ville vært betydelig redusert hvis det hadde eksistert en solid kunnskapsbase om React på forhånd.

Gjennomføringen av oppgaven har vært svært lærerik og har bidratt til en grundig forståelse av React, som vil være anvendelig i fremtidige arbeidsmiljøer. Oppdragsgiveren uttrykker også tilfredshet med prosjektet og har positive forventninger til brukernes respons på den nye løsningen.

Dette prosjektet har betydelig verdi for oppdragsgiveren, da mesteparten av applikasjonen allerede er ferdigstilt, og det gjenstår kun noen få endringer for å integrere den i deres system. Gjennom kravene til oppgaven kan oppdragsgiver se hva som er igjen på agendaen.

Dette prosjektet har også potensial til å være lærerikt for andre grupper som ønsker å gjøre lignende prosjekter. Prosjektet kan fungere som en veiledende mal for å håndtere generelle problemstillinger, da fremgangsmåten er relativt lik på tvers av ulike typer utfordringer.

I lys av diskusjon i kapittel 6.4 (forskningsspørsmål) kan det konkluderes med at prosjektet adresserer forskningsspørsmålet: "Hvordan kan oppgradering og utvidelse av Styreportalens eksisterende applikasjon 'Sceneoppsett' forbedre brukeropplevelsen, redusere drifts- og vedlikeholdskostnadene, samt øke kundetilfredsheten ved hjelp av en ny teknologisk tilnærming?" som er diskutert i denne rapporten. Til tross for visse forbedringsområder, indikerer de overordnede funnene en positiv utvikling for prosjektet, og et sterkt fundament for fremtidig forbedring og vekst.

7.2 Videre arbeid

Hvis det hadde vært mer tid til rådighet for prosjektet, ville det ha vært mulig å gjennomføre en rekke forbedringer. I første omgang ville dette ha inkludert implementering av databasen i applikasjonen, en oppgave som ble utsatt på grunn av tidsbegrensninger mot prosjektets slutfase. Denne funksjonaliteten kan relativt enkelt gjennomføres, da applikasjonen allerede har tilrettelagt for en slik implementering.

Et av de opprinnelige, men lavt prioriterte kravene var å inkludere en beskrivelse av dirigentens utstyr, som for eksempel et podium og et notestativ, ved siden av dirigentikonet. På grunn av tidsbegrensninger ble dette kravet nedprioritert. Imidlertid er det mulig å implementere dette kravet med relativ letthet, da det allerede er etablert et 'div'-element for dette formålet i applikasjonens oppsett. Når oppsettet lagres, blir det generert en JSON streng hvor det i strengen står om dirigenten har podium eller ikke. Ved å bruke samme tilnærming kan visningen av dirigent beskrivelse enkelt implementeres.

Oppdragsgiver har også uttrykt interesse for flerspråklig støtte for applikasjonen. Dette kravet hadde også lav prioritet og ble vurdert som en ekstra funksjon som kunne legges til hvis tiden tillot det. Dette er en funksjon som oppdragsgiver eventuelt kan implementere på et senere tidspunkt ved behov.

Under et av møtene med oppdragsgiver ble det gitt detaljert tilbakemelding om bruken av guider i applikasjonen, som er ment for å hjelpe brukerne med å plassere instrumenter på riktig sted i forhold til gitt korpsoppsett. Oppdragsgiveren uttrykte ønske om å kunne endre størrelsen og plasseringen av disse guidene. Dette utgjør en interessant mulighet for videre utvikling av applikasjonen.

Oppdragsgiveren har uttrykt tilfredshet med prosjektets resultat og er fornøyd med det utførte arbeidet. Videre utvikling av applikasjonen vil bli håndtert av oppdragsgiveren selv, for å legge til de siste funksjonalitetene, inkludert lagring til databasen, og for å integrere applikasjonen i det øvresystemet, Program.no.

Hvis det hadde vært mer tid tilgjengelig for videreutvikling av applikasjonen, kunne alle disse oppgavene ha blitt utført, i tillegg til en ytterligere runde med brukertesting. Likevel ble prosjektet gjennomført på en tilfredsstillende måte innenfor de gitte tidsrammer og begrensninger.

8 Referanser

Bell, J. T. (2001) Extreme programming, *Thinking for Innovation*.

Christensen, C. M. (2016) *The innovator's dilemma: when new technologies cause great firms to fail*. Paperback. Boston, Massachusetts: Harvard Business Review Press.

Christina, B. (2022) The 60:30:10 color rule for UI design *Medium*. Tilgjengelig fra: <https://bootcamp.uxdesign.cc/the-60-30-10-color-rule-for-ui-design-32695d04a7c2> (Hentet: 2023-04-24 15:23:56).

Eductive (2023) *What is a unified process model?* Tilgjengelig fra: <https://www.educative.io/answers/what-is-a-unified-process-model> (Hentet: 2023-03-10).

Fowler, M. og Parsons, R. (2011) *Domain-specific languages*. Upper Saddle River, NJ: Addison-Wesley.

Goldsmith, S. og PRP Architects (Firm) (2000) *Universal design : a manual of practical guidance for architects*. Oxford ; Boston: Architectural Press.

Google (2023a) *Firestore*. Tilgjengelig fra: <https://firebase.google.com/> (Hentet: 2023-04-17).

Google (2023b) *Firestore Pricing*. Tilgjengelig fra: <https://firebase.google.com/pricing> (Hentet: 2023-05-13).

Hampton, M. (2017) Why a Content First Strategy is Important *Medium*. Tilgjengelig fra: <https://himaya.medium.com/content-first-strategy-df2ef60df409> (Hentet: 2023-04-24 15:23:09).

JGraph_Ltd (2023) *Draw.io*. Tilgjengelig fra: <https://www.drawio.com/> (Hentet: 2023-04-16).

Justinmind (2019) The complete guide to User Testing. Tilgjengelig fra: <https://www.justinmind.com/blog/user-testing/> (Hentet: 2023-05-14 13:42:57).

Krug, S. (2014) *Don't make me think, revisited : a common sense approach to Web usability*. Third edition. Berkeley, Calif.: New Riders.

Larman, C. (2004) *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley.

Laudon, K. C. og Laudon, J. P. (2014) *Management information systems: managing the digital firm*. 13. ed., global ed. Boston Munich: Pearson.

Lidwell, W., Holden, K. og Butler, J. (2003) *Universal principles of design*. Gloucester, Mass.: Rockport.

Lucid_Software_Inc (2023) *Lucidchart*. Tilgjengelig fra: <https://www.lucidchart.com> (Hentet: 2023-04-16).

- Marinescu, F. og Avram, A. (2007) *Domain-driven design Quickly*. Lulu. com.
- Meta (2023) *React*. Tilgjengelig fra: <https://react.dev/> (Hentet: 2023-04-17).
- Microsoft (2023a) *GitHub Pages*. Tilgjengelig fra: <https://pages.github.com/> (Hentet: 2023-05-16).
- Microsoft (2023b) *Microsoft Visio*. Tilgjengelig fra: <https://www.microsoft.com/nb-no/microsoft-365/visio/flowchart-software> (Hentet: 2023-04-20).
- Microsoft (2023c) *Accessibility Insights*. Tilgjengelig fra: <https://accessibilityinsights.io>.
- Microsoft (2023d) *Microsoft Planner*. Tilgjengelig fra: <https://www.microsoft.com/nb-no/microsoft-365/business/task-management-software> (Hentet: 2023-04-13).
- Microsoft (2023e) *Microsoft Forms*. Tilgjengelig fra: <https://www.microsoft.com/nb-no/microsoft-365/online-surveys-polls-quizzes> (Hentet: 2023-04-13).
- Miro (2023) *Miro.com*. Tilgjengelig fra: <https://miro.com/> (Hentet: 2023-04-16).
- OpenJS_Foundation (2023a) *Node.js*. Tilgjengelig fra: <https://nodejs.org/en> (Hentet: 2023-04-17).
- OpenJS_Foundation (2023b) *Express.js*. Tilgjengelig fra: <https://expressjs.com/> (Hentet: 2023-04-17).
- Patterson, D. A. og Hennessey, J. L. (2021) *Computer organization and design: the hardware/software interface*. Sixth edition. Amsterdam [Netherlands] ; Boston [MA]: Morgan Kaufmann.
- Perez-Castillo, R. *et al.* (2011) Reengineering Technologies, *IEEE Software*, 28(6), s. 13-17. <https://doi.org/10.1109/MS.2011.145>
- Proff.no (2023) *Styreportalen AS - 917273898 - Nesttun - Se Regnskap, Roller og mer* (Hentet: 2023-03-08).
- Seacord, R. C., Plakosh, D. og Lewis, G. A. (2003) *Modernizing legacy systems: software technologies, engineering processes, and business practices*. Boston: Addison-Wesley.
- Specification, O. A. (2007) *Omg unified modeling language (omg uml), superstructure, v2. 1.2, Object Management Group*, 70.
- Thalheim, B. (2013) *Entity-relationship modeling: foundations of database technology*. Springer Science & Business Media.
- The_Interaction_Design_Foundation (2023) *What is Keep It Simple, Stupid (KISS)? The Interaction Design Foundation* (b. 2023). Tilgjengelig fra: <https://www.interaction-design.org/literature/topics/keep-it-simple-stupid> (Hentet: 2023-04-24 11:45:23).

Vercel (2023) *Next.js*. Tilgjengelig fra: <https://nodejs.org/en> (Hentet: 2023-04-17).

Wensberg, M. (2023) Artikkel - Ulike korpsbesetninger *Digitale Dirigentstudier*. Tilgjengelig fra: <http://digitaledirigentstudier.uis.hopto.me/modul-4-instrumentkunnskap/artikkel-ulike-korpsbesetninger/> (Hentet: 2023-05-13 11:21:55).

9 Illustrasjonslister

Under følgende kapittel er det gitt oversikt over alle figurer og tabeller brukt i rapporten.

9.1 Figurliste

Figur 2-1 Design av nåværende applikasjon Sceneoppsett (testside fra Styreportalen)	7
Figur 3-1 UML Domenemodell over applikasjon Sceneoppsett (modellert ved hjelp av draw.io) 16	
Figur 3-2 Gantt-diagram over fremdriftsplan laget i Microsoft Excel	24
Figur 3-3 Risikomatrix for utregning av risikovurdering utformet i Microsoft Excel	25
Figur 4-1 Brukstilfellediagram.....	28
Figur 4-2 Komponent plassering ift. nettleservinduet.....	30
Figur 4-3 Kanban statistikk over prosjektet generert i Microsoft Planner.....	35
Figur 4-4 Utklipp over commits til prosjektet fra GitKraken	36
Figur 4-5 Eksempel på dokumentering av funksjon i kode v.h.a. JSDoc	37
Figur 4-6 Utklipp fra README dokumentet for prosjektet	38
Figur 5-1 Utforming av applikasjon under testing	40
Figur 5-2 Eksempel på designalternativ (utformet i Figma).....	41
Figur 5-3 Oppsummering av respons fra brukerundersøkelse	43
Figur 5-4 Oppsett nr.4 fra brukerundersøkelse	44
Figur 5-5 Design av ferdigstilt applikasjon.....	46
Figur 5-6 Utklipp fra README til GitHub lagringssted.....	49
Figur 5-7 Utklipp fra JSDoc applikasjons dokumentasjon	50

9.2 Tabelliste

Tabell 1-1 Oppbygging av rapport	5
Tabell 2-1 Oversikt over utgangspunkt for funksjonelle krav	8
Tabell 3-1 Risikovurdering illustrert ved hjelp av Microsoft Excel	25
Tabell 5-1 Oversikt over implementering av funksjonelle krav	48

10 Vedlegg

Dette kapittelet inneholder referanser til vedlegg som er omtalt i rapporten. Noen av disse vedleggene er inkludert som eksterne filer og er tilgjengelige ved innleveringen av prosjektet. I tillegg er det også gitt referanser til GitHub lagringsstedet for Sceneoppsett-applikasjonen, som suppleres med en videodemonstrasjon av applikasjonen. Ytterligere vedlegg inkluderer sluttvurdering fra oppdragsgiver og brukerundersøkelse samt testing med tilhørende resultater.

10.1 Visjonsdokument

Inkludert som et eksternt vedlegg ved innleveringen av prosjektet.

10.2 Prosjekthåndbok

Inkludert som et eksternt vedlegg ved innleveringen av prosjektet.

10.3 Kravdokument

Inkludert som et eksternt vedlegg ved innleveringen av prosjektet.

10.4 Systemdokumentasjon

Inkludert som et eksternt vedlegg ved innleveringen av prosjektet.

10.5 GitHub lagringssted for prosjektet

Følgende lenke gir lesertilgang for GitHub lagringssted for prosjektet som inneholder README dokumentet som forklarer både applikasjonens funksjoner og bruk, hvilke teknologier som er brukt, videre lenker til applikasjonen under testing, den opprinnelige applikasjonens testside gitt av oppdragsgiver og midlertidig tilgang til koden er gitt for prosjektets vurdering ved bruk av verktøyet GitFront.

Lenke til lagringsstedet: <https://gitfront.io/r/589664/pthBPvjciGnu/Sceneoppsett/>

10.6 Videodemonstrasjon av applikasjonen

Videodemonstrasjon av applikasjon Sceneoppsett finnes ved følgende lenke:

https://youtu.be/MA0fw_G_ygw

10.7 Sluttevaluering fra oppdragsgiver

- Svart på grått gjør at kontrasten blir dårligere. Dette kan være et problem for folk med svakt syn.
- Når man skal velge instrumenter så drar man disse inn på canvas. Kanskje man også skulle kunne dobbeltklikke på dem, så kommer de inn på canvas?
- Antall rader burde stå til 1 som standard.
- Stoltype: Stol med notestativ burde være standard. (Velgeren burde være større så man ser hele teksten).
- Legge til stoler. Når man legger til f.eks. 8 stoler, og så 8 til. Hvis man ikke har flyttet de 8 første stolene mister man oversikten over hvor mange stoler som er der. Burde forskyve litt i x, y retning.
- Dirigent med podiet. Der burde firkanten være litt større. Det vil se bedre ut. Med notestativ burde være standard.
- OFF, Brass, Jan - Her burde buene vært litt nærmere hverandre, samt litt nærmere dirigenten.
- Veldig flott med Sceneoversikt på høyre side. Der vises det hvor mange det er av hver enkelt, men stolene burde alltid vises først i listen.
- En kjekk funksjonalitet som kunne utvikles er at når man trykker på Legg til stoler, så ble de plassert på buen-brass eller buen-jan. Så trekke på den buen for å plassere.

Alt i alt er det veldig bra. Dette er noe vi kan jobbe videre med.

Stian Sømoe 15.05.2023

10.8 Brukerundersøkelsesskjema

Brukertesting av webapplikasjon "Sceneoppsett" for Styreportalen AS

Kjære mottaker,

Vi inviterer deg herved til å delta i brukertesting av designflyt og demoversjonen av vår "Sceneoppsett"-applikasjon. Vi vil sette stor pris på om du kan gjennomføre testingen og deretter sende inn svarene dine ved hjelp av dette skjemaet så raskt som mulig. På forhånd takk for ditt bidrag.

Brukertestingprosessen er inndelt i to deler. Første del fokuserer på designelementene i applikasjonen, hvor målet er å evaluere plasseringen av ulike komponenter på skjermen for å identifisere det mest hensiktsmessige brukergrensesnittet og arbeidsflyt. Den andre delen er beregnet på å teste funksjonene som er implementert per nåværende tidspunkt og for å undersøke hvorvidt det er enkelt eller utfordrende for brukere å forstå hvordan de skal samhandle med applikasjonen.

Vi ser frem til din verdifulle tilbakemelding og takker deg på forhånd for din innsats i dette prosjektet.

Med vennlig hilsen,
Oppdragsteamet

Section 1

Del 1:

Vi vil først presentere deg for fem forskjellige illustrasjoner som viser potensielle fremtidige utforminger av sceneapplikasjonen. Under hvert bilde, vennligst ranger din tilfredshet med designet ved å gi det en score fra 1 til 5 stjerner. Deretter vil vi be deg om å gi en kort kommentar om hvilken illustrasjon du foretrakk og begrunne ditt valg. Vi ber deg spesielt vurdere rekkefølgen av handlingene du ville ha utført i hvert design, og gjerne kommentere hvorfor arbeidsflyten føltes bedre i de enkelte oppsettene.

1

*

SCENEOPPSETT

Dirigent

Podie
 Notestativ

Stoler
Legg til

Sceneoversikt

Instrument1 103
Instrument2 23

Opplasting egen oppsett

Ingen valgt fil

Beskrivelse/
kommentar

Ta screenshot
Lagre

Instrumenter

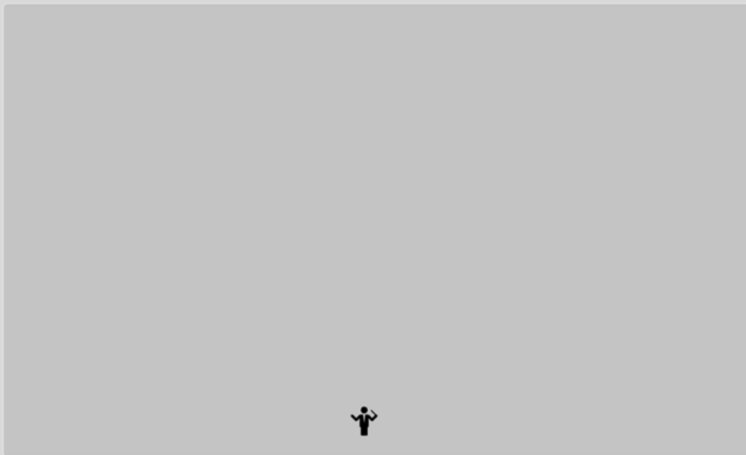
Kategori 1 Kategori 3

☆ ☆ ☆ ☆ ☆

2

*

SCENEOPPSETT



Dirigent

Podie


Notestativ

Stoler Select

Velg antall

Legg til

Opplasting egen oppsett

 **Last opp**

Ingen valgt fil

Beskrivelse/kommentar

Ta screenshot

Lagre



3

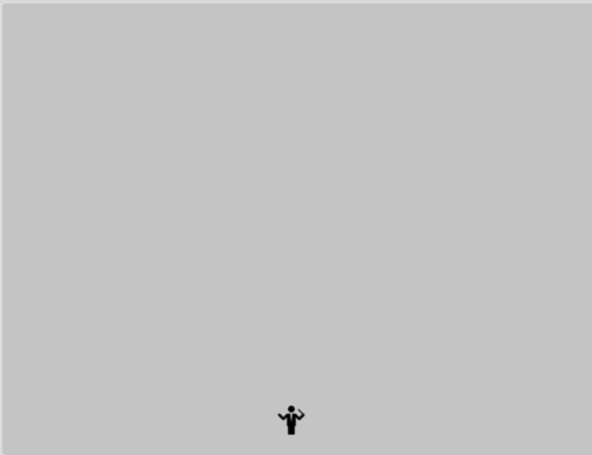
*

SCENEOPPSETT

Sceneoversikt

Instrument1 103

Instrument2 23



Dirigent

Podie


Notestativ

Stoler Select

Velg antall

Legg til

Opplasting egen oppsett

 **Last opp**

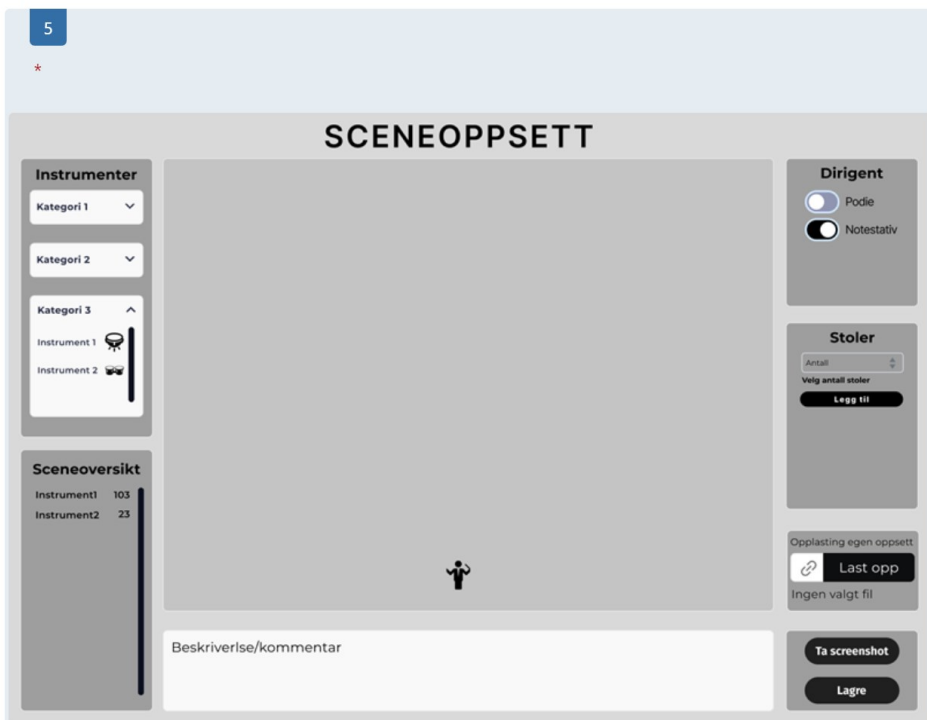
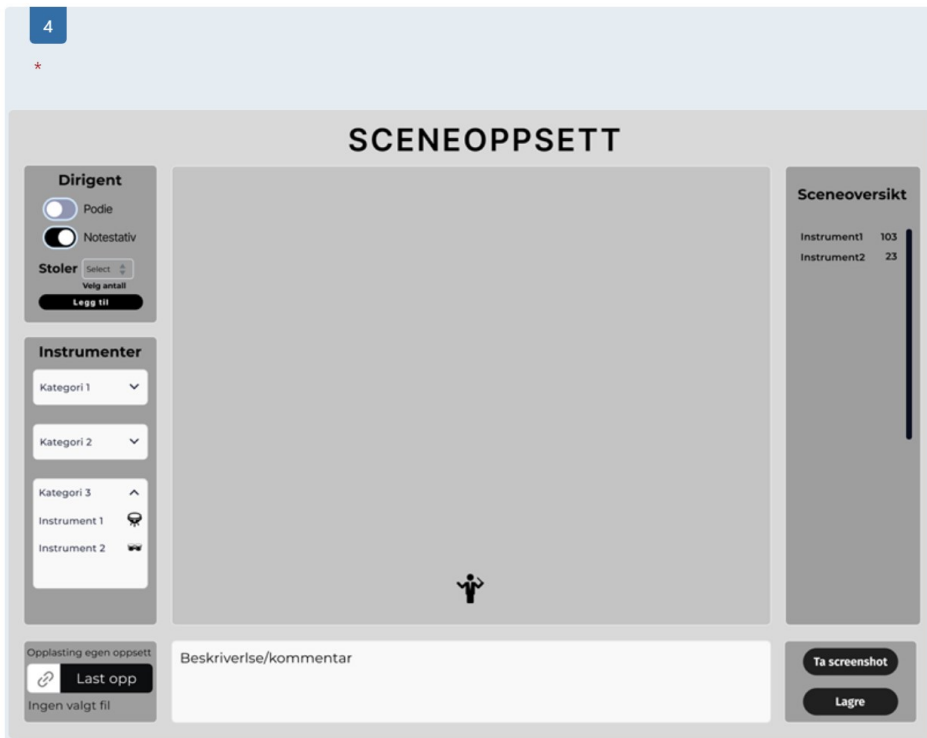
Ingen valgt fil

Beskrivelse/kommentar

Ta screenshot

Lagre





6

Hvilket oppsett likte du best? Hvorfor? (Skriv nummeret på det oppsettet du likte best) *

Enter your answer

7

Har du ytterligere ideer eller betraktninger angående plassering av komponentene som ikke er representert i alternativene ovenfor?

Enter your answer

8

Er det et bestemt fargevalg og dimensjon for tekst, komponenter og sceneområdet som kunne bidra til en mer behagelig og effektiv brukeropplevelse med applikasjonen? Ville det være ønskelig å kunne tilpasse disse aspektene etter individuelle preferanser?

Enter your answer

Section 2

...

Del 2:

Du vil nå få tilgang til en demoversjon av webapplikasjonen, der du kan teste funksjonene vi har implementert så langt. Etter å ha testet demoversjonen, ber vi deg om å besvare åtte spørsmål knyttet til din opplevelse med applikasjonen.

Her er linken til applikasjonen for demoversjonen: <https://sceneoppsettbrukertesting.web.app/>

Foreløpig er 'submit'-knappen ment for å generere data fra sceneoppsettet for rapportoppretting. Følgelig vil det ikke vises noen informasjon på skjermen når knappen trykkes. Utover dette, er det oppfordret til å utforske og evaluere alle tilgjengelige funksjoner i applikasjonen.

Under er 7 spørsmål knyttet til applikasjonen.

9

1. Hva var din første tanke da du åpnet appen? *

Enter your answer

10

2. Følte du at appen var intuitiv å bruke? *

Enter your answer

11

3. Var det noen funksjoner som du ikke kunne finne eller var vanskelig å finne? *

Enter your answer

12

4. Hva syntes du om funksjonene i applikasjonen? *

Enter your answer

13

5. Fikk du noen feilmeldinger eller problemer mens du brukte appen? *

Enter your answer

14

6. Var det enkelt å fullføre handlingene du ønsket å gjøre i applikasjonen? *

Enter your answer


15

7. Er det noen funksjoner eller endringer du ville ønske var inkludert i appen for å forbedre opplevelsen? *

Enter your answer

10.9 Resultat av brukerundersøkelse og -testing

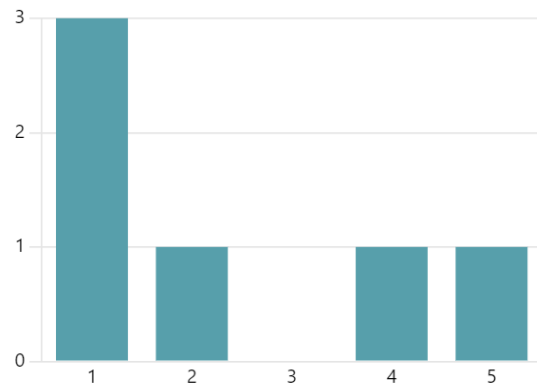
Brukertestning av webapplikasjon "Sceneoppsett" for Styreportalen AS

6 Responses	29:53 Average time to complete	Active Status
View results		 Open in Excel ⋮

1.

[More Details](#)

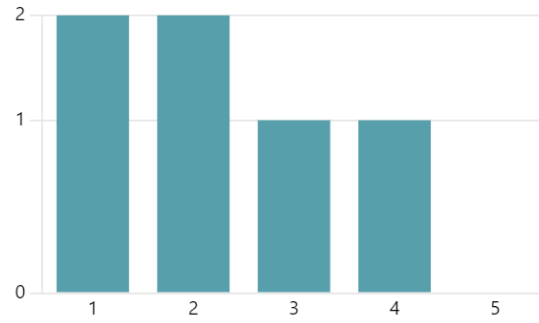
2.33
Average Rating



2.

[More Details](#)

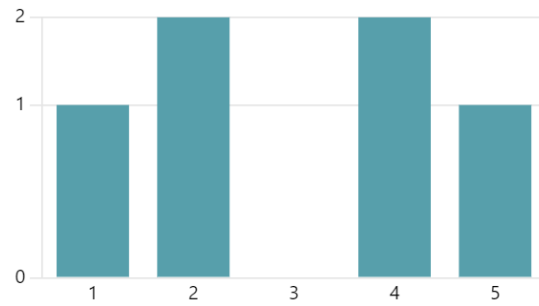
2.17
Average Rating



3.

[More Details](#)

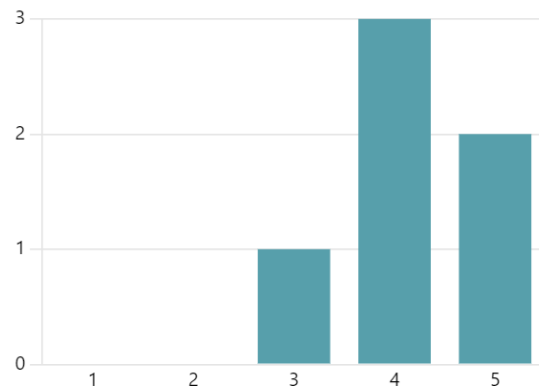
3.00
Average Rating



4.

[More Details](#)

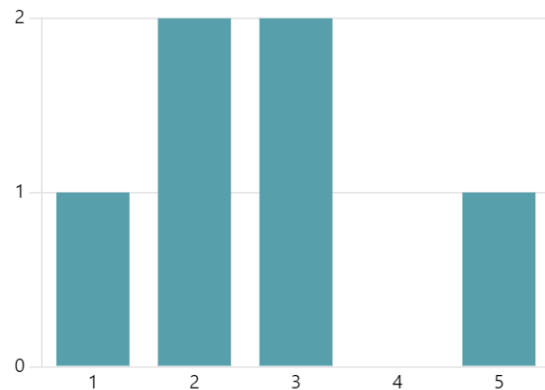
4.17
Average Rating



5.

[More Details](#)

2.67
Average Rating



6. Hvilket oppsett likte du best? Hvorfor? (Skriv nummeret på det oppsettet du likte best)

[More Details](#)

6
Responses

Latest Responses

"4"

"Jeg likte oppsett 4 best. Jeg likte det best fordi vi i den vestlige verden er va...

"4"

7. Har du ytterligere ideer eller betraktninger angående plassering av komponentene som ikke er representert i alternativene ovenfor?

[More Details](#)

5
Responses

Latest Responses

"Brukeren bør kunne flytte, velge, skjule og vise de ulike menyene."

"Kan det være et alternativ for raskere utfylling å velge antall rader med stol...

"Nope"

8. Er det et bestemt fargevalg og dimensjon for tekst, komponenter og sceneområdet som kunne bidra til en mer behagelig og effektiv brukeropplevelse med applikasjonen? Ville det være ønskelig å kunne tilpasse disse aspektene etter individuelle preferanser?

[More Details](#)

6
Responses

Latest Responses

"Eksemlene har jo ingen farger."

"Det ble veldig grått og svart nå. Det gjør at helptext-en i feltet for stoler nes...

"Viktig med god kontrast. Hvis man ser litt dårlig vil grå bakgrunn med svart...

9. 1. Hva var din første tanke da du åpnet appen?

[More Details](#)

6
Responses

Latest Responses

"Rotete"

"Skissene var penere enn det som er implementert. Både fargevalg og hvord...

"Mangler litt design. Litt rotete"

10. 2. Følte du at appen var intuitiv å bruke?

[More Details](#)

6
Responses

Latest Responses

"Nei"

"Ja"

"nesten"

11. 3. Var det noen funksjoner som du ikke kunne finne eller var vanskelig å finne?

[More Details](#)

6
Responses

Latest Responses

"Hvilke funksjoner burde jeg lete etter?"

"Nei, men det hadde vært lettere om siden var responsiv og tilpasset seg til ..."

"Nei"

12. 4. Hva syntes du om funksjonene i applikasjonen?

[More Details](#)

6
Responses

Latest Responses

"Vet ikke helt hva jeg skal mene her. Hva forventes?"

"Ok, men da jeg valgte 40 stoler så havnet de på en lang line som gjorde at ..."

"Fungerer greit"

13. 5. Fikk du noen feilmeldinger eller problemer mens du brukte appen?

[More Details](#)

6
Responses

Latest Responses

"Tror hele greien gikk og hengte seg. :)"

"Nei"

"Ved innleggelse av mange stoler kom de utenfor scenen."

14. 6. Var det enkelt å fullføre handlingene du ønsket å gjøre i applikasjonen?

[More Details](#)

6
Responses

Latest Responses

"Nei"

"Ja"

"ja"

15. 7. Er det noen funksjoner eller endringer du ville ønske var inkludert i appen for å forbedre opplevelsen?

[More Details](#)

6
Responses

Latest Responses

"Mange"

"Forslaget mitt fra tidligere om å håndtere antall stoler bedre. Når jeg velger...

"Ser det er ta skjermbilde som er bra. Men last ned som PDF hadde vært bed..."