

Geppetto – fra Windows til Web

Kravdokumentasjon

Versjon 2.0

Dokumentet er basert på Kravdokumentasjon utarbeidet ved NTNU. Revisjon og tilpasninger til bruk ved IDER, DATA-INF utført av Carsten Gunnar Helgesen, Svein-Ivar Lillehaug og Per Christian Engdal. Dokumentet finnes også i engelsk utgave.

REVISJONSHISTORIE

Dato	Versjon	Beskrivelse	Forfatter
21.02.2023	0.5	Innledning, funksjonalitet og use-cases beskrevet.	Markus Gustavsen, Håkon Grimen
22.02.2023	0.9	Prototype, mindre korrigeringer	Håkon Grimen
23.02.2023	1.0	Ferdigstilt iterasjon 1	Håkon Grimen, Markus Gustavsen, Håkon Markhus
13.04.2023	2.0	Små korrigering og rettelser.	Håkon Grimen



INNHOLDSFORTEGNELSE

1	INNLEDNING	1
2	FUNKSJONALITET	2
3	KOMPONENTDIAGRAM	4
4	PROTOTYPER	5
4.1	BRUKERFLYT	5
5	REFERANSER	7

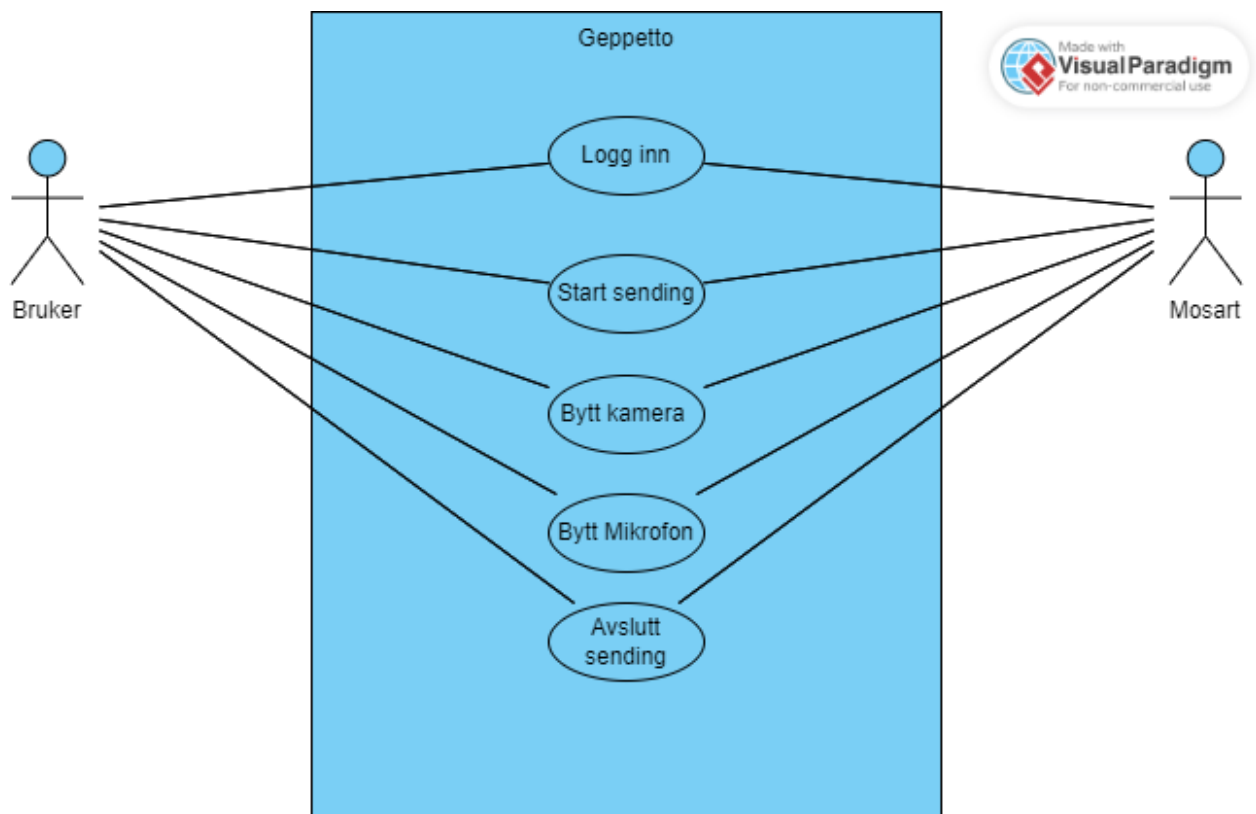
1 INNLEDNING

Dette dokumentet er skrevet i forbindelse med et bachelor-prosjekt som gjennomføres ved Høgskulen på Vestlandet, vårsemesteret 2023. Oppgaven går ut på å utvikle en web-applikasjon som skal kommunisere med et preeksisterende backend-system som håndterer gjennomføringen av medie-sendinger.

I utformingen av prosjektet er det lagt frem noen krav fra oppdragsgiver, men i hovedsak er det krav vi har satt oss selv i utformingen av oppgaven. Disse kravene omhandler både funksjonelle og estetiske, selv om hovedfokuset til gruppen har vært på det tekniske og funksjonelle. Fra vår side har kravene stort sett omhandlet funksjonaliteten til det ferdige produktet vi ønsker å levere, og vært vektet ut ifra hvor essensielle vi bedømmer de som i sluttproduktet.

Rent håndfast er det de mest basale kravene som er de viktigste, og disse omhandler de grunnleggende funksjonene som er nødvendige for bruk. I første omgang er kravet at en kan logge seg på den aktuelle sendingen, og styre de aspektene en trenger i det tilfellet. Her vil det være nødvendig å kunne bytte kamera, bytte mikrofon, og gå til neste i et allerede planlagt tidsskjema. Videre har vi ønsker om funksjonalitet vi vil utvikle, men de er ikke gjeldende som krav i samme forstand.

2 Funksjonalitet



Figur 1: brukstilfellediagram, illustrert via Visual Paradigm. (Visual Paradigm, 2023).

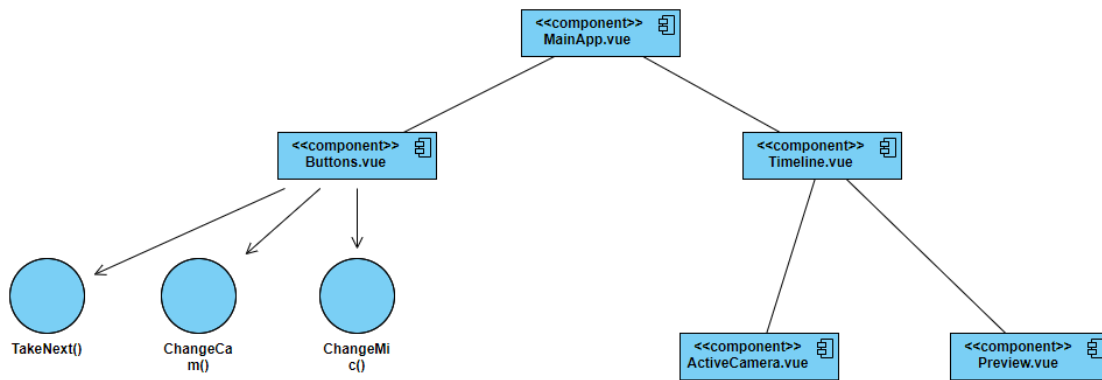
Ovenfor vises første utkast til et brukstilfellediagram for en bruker som ønsker å gjennomføre en mediesending via vår applikasjon. Den starter med en innlogging, som autentiseres via Viz Mosart (Vizrt, 2023) back-end systemet vi kommuniserer med, og som deretter gir brukeren muligheten til å avholde en sending. Derifra vil brukeren interagere med applikasjonen, som vil videreformidle handlinger til Viz Mosart, som vil gjennomføre de. Brukstilfellet avsluttes ved at brukeren avslutter sendingen i applikasjonen.

Navn:	Gjennomføre sending
Aktører:	Bruker, Mosart
Hensikt/Målsetting:	Logge inn og gjennomføre sending
Normalflyt:	<ol style="list-style-type: none"> 1. Bruker logger inn i web-applikasjonen 2. Bruker velger ferdiglaget sending og starter sendingen 3. Bruker gjennomfører en sending ved bruk av knappene i applikasjonen. 4. Bruker avslutter sendingen etter den er ferdig.
Alternativ flyt [#1]:	<ul style="list-style-type: none"> • Om brukeren ikke vil følge det planlagte tidsløpet kan hen benytte seg av knappene som regulerer sendingen i direktetid.
Unntaksflyt [#1]:	<ul style="list-style-type: none"> • Dersom man mister tilkoblingen til internett vil man ikke kunne styre sendingen lenger, og den planlagte tidslinjen følges dersom mulig.

3 KOMPONENTDIAGRAM

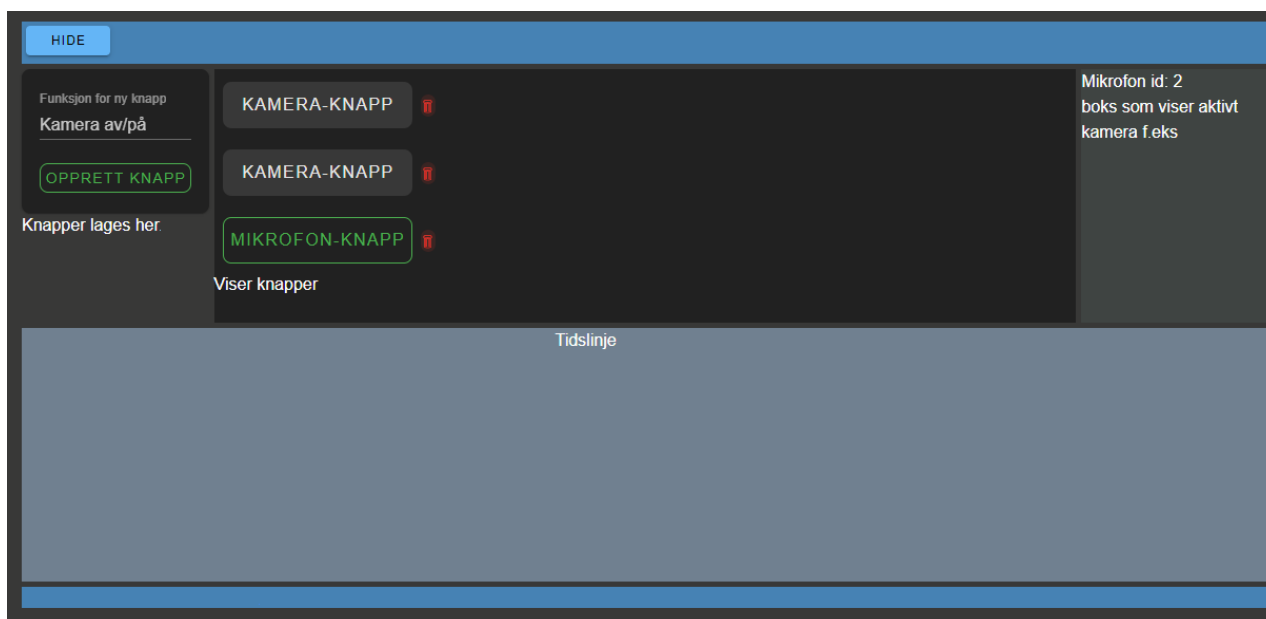
For gruppens prosjekt ble det besluttet at det var mer relevant med et komponentdiagram, ettersom vi jobber i et komponent-basert rammeverk (VueJs, 2023). Under er det tegnet opp de innledende komponentene som er planlagt i applikasjonen, med forbehold om at det kan bli nødvendig å lage nye.

De to hovedkomponentene applikasjonen vil bestå av er utvalgte knapper og en tidslinje. De utvalgte knappene vil være komponenter en kan interagere med via web-applikasjonen, og de vil samhandle med en funksjon i backend, etter valgt ønske. Disse valgene vil bli representert som forandringer i en foreløpig tidslinje, som er den andre komponenten applikasjonen består av. Denne komponenten vil være en dynamisk tidslinje som til enhver tid viser aktuelt kamera med tilhørende mikrofon, samt den videre tidsplanen for sendingen.



Figur 2: komponentdiagram

4 PROTOTYPER

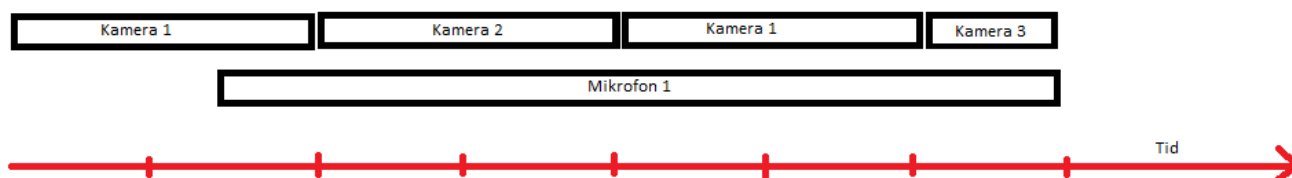


Figur 3: applikasjonens hovedside, tidlig utkast

Figur 1 viser en prototype gruppen har jobbet med i oppstartsfasen av dette prosjektet. Den viser grunnleggende layout som blir det gruppen planlegger å bygge videre på. Oppdeling av forskjellige områder er gjort for å tydeliggjøre hvor hver enkelt del, eller komponent, hører hjemme.

Dette vil videreutvikles til et produkt som skaleres korrekt, uavhengig av hvilken enhet man bruker. Rask, reaktiv og visuelt tiltrekkende er noen av verdiene gruppen har med seg gjennom arbeidet i dette prosjektet. Brukerens opplevelse anses som svært viktig i denne applikasjonen.

I den blå boksen, merket med "Tidslinje", skal det være en kjøreplan som viser hvilket kamera og mikrofon(er) som skal være i bruk til enhver tid. Applikasjonen skal selv bytte til det kameraet og den mikrofonen tidsplanen til enhver tid viser, dersom slik funksjonalitet er aktivert.

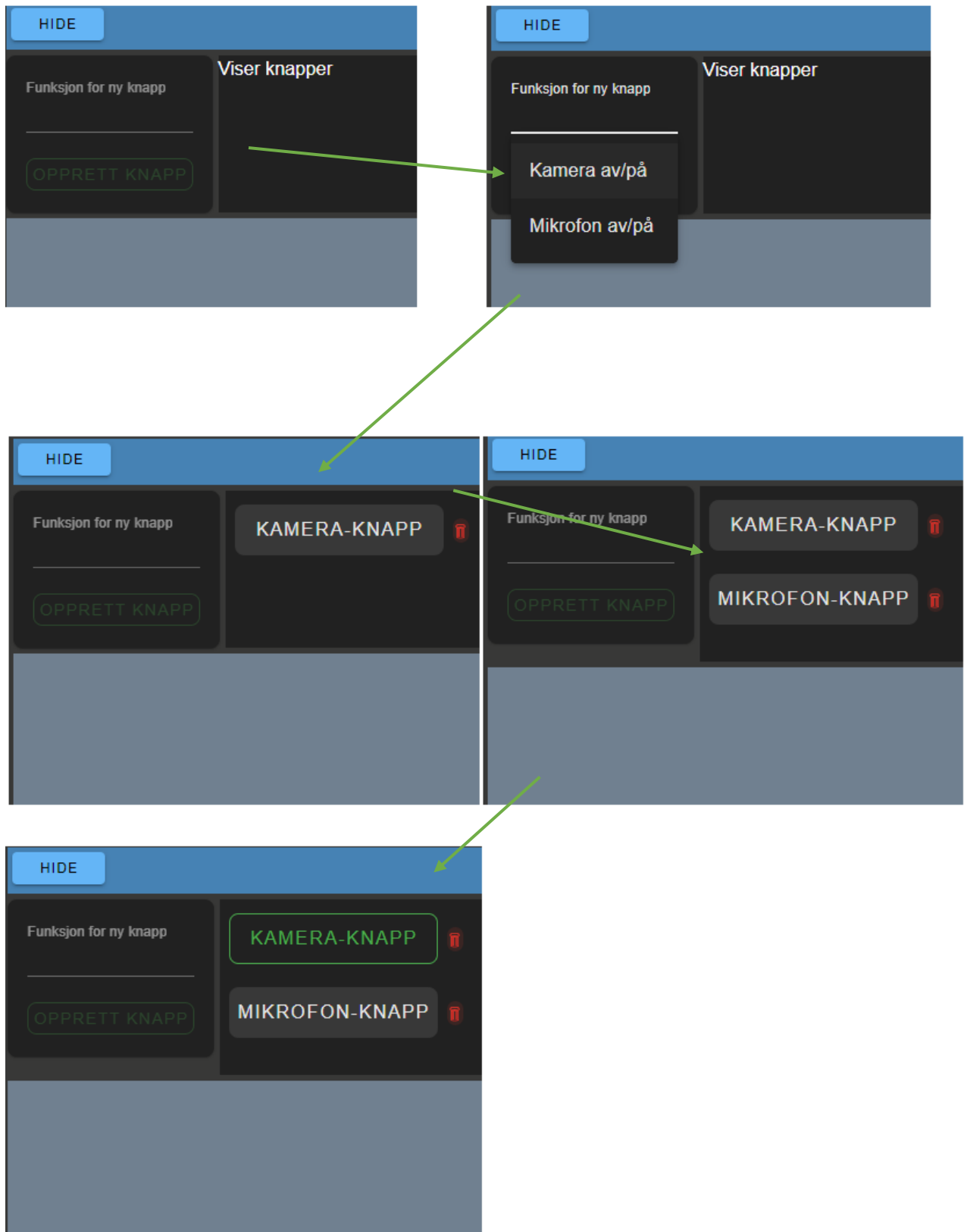


Figur 4: illustrasjon av tidslinje

4.1 Brukerflyt

Figur 5 viser en flyt der brukeren oppretter nye knapper, som så legges til i området for knapper. Deretter trykker brukeren på en kamera-knapp slik at den er aktiv. Grønn ramme indikerer her at

knappen og enheten er aktiv. Dersom brukeren trykker igjen vil den grønne rammen forsvinne, for å tydelig vise at denne knappen og enheten den styrer, ikke lenger er aktive.



Figur 5: eksempler på bruk av applikasjonen, fra en tidlig prototype

5 Referanser

Visual Paradigm, 2023. *Visual Paradigm*. [Internett]

Available at: <https://online.visual-paradigm.com>

Vizrt, 2023. *Vizrt.com*. [Internett]

Available at: <https://www.vizrt.com/products/viz-mosart>

[Funnet 22 Februar 2023].

VueJs, 2023. *Vue.js*. [Internett]

Available at: vuejs.org

[Funnet 22 Februar 2023].