

Kan 3D-visualisering av EKG-bølgeformer være gunstig? Systemutviklingsprosjekt



*Institutt for datateknologi og informatikk, NTNU
Systemutviklingsprosjekt, TDAT3022*

3D-visualisering av EKG-bølgeformer: 3CG

Dalheim, William; Kalstad, Mikael; Tronstad, Henrik

Forord

Oppgaven ble gitt av Infiniwell AS, og gikk ut på å utvikle et web-basert system for 3D-visualisering av EKG-bølgeformer. Fra faget TDAT2003 (Systemutvikling 2 med web-applikasjoner) fikk gruppemedlemmene lære om det å utvikle nettsider ved bruk av bibliotek som React. Gruppen ønsket å utfordre seg selv og lære nye teknologier ved å bygge på erfaringene gjennom dette prosjektet. Av disse grunnene var medlemmene enige om at oppgaven fra Infiniwell AS virket spennende. På denne måten var det mulig å utforske hvilke muligheter som eksisterer innenfor 3D-visualisering gjennom web-utvikling.

Arbeidsprosessen har vært påvirket av koronavirus-pandemien. Arbeidet har for det meste foregått digitalt, hvor gruppemedlemmene kommuniserte over Discord (Discord Inc., 2020), med unntak av noen fysiske møter tidlig i prosjektet. Gruppen jobbet sammen ved bruk av lean utviklingsmetode.

Vi ønsker å takke veileder Alexander Holt for gode råd underveis. I tillegg takker vi Odd Sandbekkhaug fra Infiniwell for gode tilbakemeldinger og oppfølging gjennom hele prosjektet. Gruppen har vært i kontakt med disse partene gjennom digitale møter. Vi ønsker å takke Hilde Søvik og hennes kollegaer ved Nordlandssykehuset som stilte opp til brukertester. Deres kunnskaper innenfor kardiologi har vært til stor hjelp for oss i det å undersøke hvor gunstig visualisering av EKG-bølgeformer er i tre dimensjoner.

Dato: 12.12.2020

Sted: Trondheim

Dalheim, William

William Dalheim

Kalstad, Mikael

Mikael Kalstad

Tronstad, Henrik

Henrik Tronstad

Oppgavetekst

I tabellen under er oppgaveteksten slik den ble publisert på BlackBoard under emnet TDAT3022 Systemutviklingsprosjekt.

Arbeidstitel:	3D visualisering av EKG bølgeformer
<p>Hensikten med oppgaven:</p> <p>Infiniwell er et AI selskap som spesialiserer i analyse av pasient data, for eksempel EKG bølgeformer. EKG er målinger av forandringer i elektropotensialet over hjertet, og hvert hjerteslag blir representert i en enkel 2D signal bølgeform som former mer eller mindre repeterende mønster. Tolking av EKG er en spesialitet som kardiologer bruker mye tid på, og vår AI motor gjør dette arbeidet lettere. EKG bølgeformer skrives vanligvis ut på rutepapir med standard rute-enheter etc. som gjør at alle EKG utskrifter har et standard format.</p> <p>Infiniwell utvikler web-baserte verktøy som visualiserer disse bølgeform signalene samt AI-analysen av signalene og vi utfordrer et student-team til å utvikle en ny/bedre måte å visualisere, inspisere og behandle disse signalene på. For eksempel så må en kardiolog kunne korrigere AI-analysen der den har tatt feil. Dette innebærer at visualiseringen blir interaktiv.</p> <p>Dette er jo ikke nødvendigvis 3D men vi er nysgjerrige på hvordan 3D kan gjøre visualisering og arbeidsflyt bedre. OpenGL/WebGL/ThreeJS kan også være en veldig god plattform for high-performance 2D grafikk – men vi ser gjerne at vi tar i bruk 3D for å få bedre visualiseringer.</p>	
<p>Kort beskrivelse av oppgaveforslag:</p> <p>Basert på EKG datafiler fra Infiniwell, teamet vil utvikle et effektivt system som kan visualisere EKG bølgeformer samt behandle (merke, korrigere, annotere, ...) signalene.</p> <p>Prosjektet bør helst kjøres i nettleser for at vi kan lettere integrere inn i våre systemer, men vi er åpne for desktop OpenGL implementasjoner om dette er letter for teamet.</p> <p>EKG signaler kan ha mange forskjellige «kanaler» med signaler, typisk 1, 2, 3, 6 eller 12 avhengig av hvor mange elektroder som er brukt i målingen. Det vil si at det er opp til 12 forskjellige bølgeformer i parallell pluss en eller flere AI-analyse annoteringer som skal visualiseres, og dette kan gi oss litt spillerom for kreativitet i 3 dimensjoner.</p> <p>I første omgang så vil teamet jobbe med eksisterende datafiler, men som et stretch-goal så kan vi tenke litt på hvordan det vil virke om data strømmer inn i systemet i real-time.</p>	
Oppgaven passer for (kryss av de(t) som passer og skriv evt. en kommentar til oss):	<ul style="list-style-type: none"> - Prosjektoppgave - Bacheloroppgave
Kan oppgavestiller stille arbeidsplass med nødvendig utstyr og programvare:	Infiniwell kan bidra med datafiler
Hvis ikke, hva kreves av maskin og programvare:	Studenter må stille med egen PC
Oppgaven passer best for, antall studenter:	<ul style="list-style-type: none"> - 1 - 2 - 3 - 4

Opplysninger om oppgavestiller	
Navn på bedrift eller organisasjon:	Infiniwell AS
Adresse	AI Village, Pirsenteret, Trondheim
Postnummer	7010
Poststed	Trondheim
Navn på kontaktperson/veileder:	Odd Sandbekkhaug
Telefon:	46923967
Epost:	odd@infiniwell.ai
Utfyllende kommentarer til hva oppgaven gjelder:	
Dette blir en litt eksperimentell oppgave med stor kreativ frihet til teamet. Det viktigste her er å tenke nytt, men fremdeles forankre visualiseringen litt i dagens måte å visualisere EKG på slik at leger og kardiologer kan føle seg litt hjemme. Om vi kommer i mål med det, så vil visualiseringen bli integrert i Infiniwell's løsninger.	

I tillegg til kravene gitt i oppgaveteksten, ble det også utarbeidet et visjonsdokument i samarbeid mellom gruppemedlemmer og oppgavestiller. Visjonsdokumentet er lagt til rapporten som vedlegg [C](#). I dette dokumentet er brukernes behov og funksjonelle egenskaper ved systemet gitt i større detalj.

Sammendrag

Kan 3D-visualisering av EKG-bølgeformer være gunstig? Vi har utviklet et web-basert system for 3D-visualisering av EKG-data med tre ulike visualiseringsmetoder med React og ThreeJS. Systemet er produsert for Infiniwell AS. Resultatet fra brukertester med fagpersonell tilsier at en 3D-visualisering ikke er gunstigere enn en 2D-visualisering. Feltet er uutforsket og kan fortsatt ha potensiale til å være nyttig for helsearbeidere.

Innholdsfortegnelse og ev. figur- og tabelliste

Forord	1
Oppgavetekst	2
Sammendrag	4
Innholdsfortegnelse og ev. figur- og tabelliste	5
Kapittel 1: Introduksjon og relevans	7
Akronymer og forkortelser	8
Kapittel 2: Teori	9
ECG	9
VCG	9
Kapittel 3: Valg av teknologi og metode	11
Arbeids- og rollefordeling	11
Systemutviklingsprosess	11
Visualisering av EKG-data	12
Valg av teknologier	13
Implementasjon på ulike plattformer	13
Brukergrensesnitt	14
3D-visualisering	14
ThreeJS	14
React-three-fiber	14
Tilstandsbehandling	15
Brukertest	16
Universell utforming	16
Oppfattelighet	17
Brukbarhet	17
Forståelighet	17
Kapittel 4: Resultater	19
Ingeniørfaglige resultater	19
Produktet	19
Mål og status	20

Brukertester	21
Administrative resultater	24
Tidsforbruk	24
Kapittel 5: Diskusjon	26
Brukertest	26
Klinisk tilbakemelding fra målgruppen	27
Krav og mangler	28
Svakheter	28
Styrker	29
Tidsforbruk	30
Gruppearbeidet	30
Kapittel 6: Konklusjon og videre arbeid	32
Referanser	33
Vedlegg	34
A - Kravdokumentasjon	35
B - Systemdokumentasjon	49
C - Visjonsdokument	63

Kapittel 1: Introduksjon og relevans

Infiniwell AS jobber blant annet med utvikling av verktøy for visualisering og analyse av EKG-bølgeformer. Bedriften ønsket en ny eller bedre måte å visualisere EKG-bølgeformene på. Tolking av EKG er noe som kardiologer bruker mye tid på ifølge oppgaveteksten. Ved å utvikle et nytt system kan det være til stor hjelp for kardiologer, og spare kritisk tid for pasientene. Hvis den nye visualiseringen er intuitiv, kan det føre til at andre helsearbeidere også kan bruke systemet.

Vår problemstilling er “Kan 3D-visualisering av EKG-bølgeformer være gunstig?”, og i den sammenheng har vi utviklet et system som visualiserer EKG-bølgeformer i tre dimensjoner. Dagens løsning hos Infiniwell AS er i to dimensjoner. Målet var å produsere en ny og enestående løsning for visualisering av EKG-data.

Rapporten er delt opp i seks kapitler. I kapittel 2 gjennomgås teorien som er nødvendig for å kunne forstå konseptene som diskuteres. Kapittel 3 inneholder beskrivelser av de valgene vi gjorde underveis, hvordan vi løste problemer og hvilke teknologier vi tok i bruk. I kapittel 4 presenterer vi resultatene som er produsert gjennom prosjektet, og disse diskuterer vi i kapittel 5. I kapittel 6 setter vi strek under prosjektet, og snakker om eventuelt videre arbeid. Etter disse kapitlene presenteres kildene vi har tatt i bruk, og alle vedlegg følger etter.

Vedleggene er delt opp med bokstaver for å kategorisere dem. Vedlegg [A](#) er kravdokumentasjon, [B](#) er systemdokumentasjon og [C](#) er visjonsdokumentet. Prosjekthåndboka er lagt ved som separat fil etter ønske fra veileder.

Akronymer og forkortelser

2D - Todimensjonal

3D - Tredimensjonal

AI - Artificial Intelligence

AS - Aksjeselskap

EKG - Elektrokardiografi

ECG - Electrocardiogram

GUI - Graphical User Interface

HTML - HyperText Markup Language

JS - JavaScript

JSON - JavaScript Object Notation

JSX - JavaScript XML (Extensible Markup Language)

NTNU - Norges teknisk-naturvitenskapelige universitet

PC - Personal Computer

UI - User Interface

VCG - Vectorcardiogram

Kapittel 2: Teori

ECG

Electrocardiogram (på norsk: elektrokardiografi) brukes til å måle spenningsforskjeller mellom deler av kroppen mens hjertet slår. Ved en EKG måles det vanligvis 12 ulike verdier. På fagspråk kalles disse for standardavledninger, mens vi forholder oss til kanaler. De tre første har navn I, II og III. I måler spenningen mellom høyre arm og venstre arm, II mellom høyre arm og venstre bein og III mellom venstre arm og venstre bein. Etterfulgt av disse er kanalene aVR, aVL og aVF. Disse er summene av de tre første målt fra tre ulike punkter på kroppen. De seks siste kanalene er V1 til V6 og måler verdier rundt brystområdet. (Arnesen, 2018) En EKG-måling består av flere samplinger for disse kanalene som vi behandler på listeform.

VCG

Vectorcardiogram (på norsk: vektorkardiografi) er en måte å visualisere EKG-data på. Ved å se på hjertets elektriske impulser i vektorform, vil man se på hjertets aktivitet basert på retning og størrelse. Vektorkardiogrammer beskriver altså retning og størrelse på hjertets elektromotoriske krefter. Vektoren vil variere i hver hjertesykklus, og kalles derfor en øyeblikksvektor. Ved å se på vektorens endepunkt over tid, vil man ende opp med en kurve. Denne kurven er en vektorsløyfe. (Arnesen, 2019)

Punktene i et vektorkardiogram beregnes ved hjelp av en matrisetransformasjons som utføres på et bestemt utvalg av kanalene. Disse kanalene er i rett rekkefølge: V1, V2, V3, V4, V5, V6, I og II. Resultatet av en transformasjon er en vektor med tre komponenter, en for hver dimensjon.

$$\begin{bmatrix} X & Y & Z \end{bmatrix}^T = D^{-1} \begin{bmatrix} V1 & V2 & V3 & V4 & V5 & V6 & I & II \end{bmatrix}^T$$

I ligningen over er D^{-1} transformasjonsmatrisen. Hver av komponentene i vektorene inneholder flere verdier, og kan derfor anses som lister. Fra boken *Computers in Cardiology* presenteres tre ulike transformasjoner som anvendes på disse kanalene. (Guillem et al., 2006, s. 252).

Dowers inverse transformasjon:

$$D^{-1} = \begin{bmatrix} -0.172 & -0.073 & 0.122 & 0.231 & 0.239 & 0.193 & 0.156 & -0.009 \\ 0.057 & -0.019 & -0.106 & -0.022 & 0.04 & 0.048 & -0.227 & 0.887 \\ -0.229 & -0.31 & -0.246 & -0.063 & 0.055 & 0.108 & 0.022 & 0.102 \end{bmatrix}$$

PLSV inverse transformasjon:

$$D^{-1} = \begin{bmatrix} -0.266 & 0.027 & 0.065 & 0.131 & 0.203 & 0.22 & 0.37 & -0.154 \\ 0.088 & -0.088 & 0.003 & 0.042 & 0.047 & 0.067 & -0.131 & 0.717 \\ -0.319 & -0.198 & -0.167 & -0.099 & -0.009 & 0.06 & 0.184 & -0.114 \end{bmatrix}$$

QLSV inverse transformasjon:

$$D^{-1} = \begin{bmatrix} -0.147 & -0.058 & 0.037 & 0.139 & 0.232 & 0.226 & 0.199 & -0.018 \\ 0.023 & -0.085 & -0.003 & 0.033 & 0.06 & 0.104 & -0.146 & 0.503 \\ -0.184 & -0.163 & -0.19 & -0.119 & -0.023 & 0.043 & 0.085 & -0.13 \end{bmatrix}$$

Kapittel 3: Valg av teknologi og metode

Valgene av teknologi og metode, ble besluttet av gruppen ut fra kravene spesifisert i visjonsdokumentet. I tillegg var valgene basert på tilbakemeldinger fra oppgavestiller og veileder, våre tidligere erfaringer med prosjektarbeid, og ny lærdom vi fikk underveis.

Arbeids- og rollefordeling

Vi som har jobbet i lag med dette prosjektet valgte å ikke ha noen teamleder. Grunnen til dette er at vi ikke så noe behov for å ha en person som delegerer arbeidsoppgaver. Sammen har vi fordelt arbeidsoppgavene i egne hovedområder innenfor utvikling:

- Ansvarlig for UI og alt av grensesnitt som ikke er 3D: Mikael Kalstad
- Ansvarlig for 3D-visualisering: William Dalheim
- Ansvarlig for UI direkte tilkoblet 3D-visualisering: Henrik Tronstad

Mer spesifikke arbeidsoppgaver ble spesifisert i et “issue board” i GitLab repositoriet, hvor hver arbeidsoppgave fikk tildelt en ansvarlig person. Slike arbeidsoppgaver lå åpne og ble tatt av dem som ønsket å utføre den spesifikke oppgaven.

I tillegg har vi fordelt det administrative arbeidet:

- Ansvarlig for alt som omhandler møter: Henrik Tronstad
- Ansvarlig for ukentlig statusrapport: Mikael Kalstad
- Ansvarlig for Git og repositoryet: William Dalheim

Hver enkelt deltaker i prosjektgruppen var ansvarlig for å lage egne timelister.

Systemutviklingsprosess

Systemutviklingsprosessen var i stor grad inspirert av forskjellige smidige metoder. Tidligere har vi jobbet med scrum. I dette prosjektet var vi et lite team, på tre personer, så vi kunne for eksempel ikke gjennomføre scrum med en scrum-master. Vi gjennomførte heller ikke noen daglige møter med teamet for å avklare hva vi har gjort, og hva vi skal gjøre videre. Siden vi bare var tre personer, jobbet vi tett med hverandre. Vi møttes flere dager i uken digitalt, vanligvis mandager, onsdager og fredager.

Kravene til systemet ble utarbeidet sammen med oppdragsgiver i visjonsdokumentet. Dette dokumentet besto av de kravene som ble oppgitt i oppgaveteksten, og de kravene som dukket opp underveis i prosjektet. Use-case og user-stories ble skrevet tidlig i planleggingsfasen, og ble oppdatert underveis.

Vi benyttet oss av et issue board i GitLab, hvor hvert team-medlem tok ansvar for ulike arbeidsoppgaver. Issue board-et besto i hovedsak av kravene som ble utarbeidet i visjonsdokumentet. I tillegg ble det lagt til flere issues underveis av teamet der det var nødvendig, som for eksempel rettet mot spesifikke bugs. Vi gjennomførte møter med oppdragsgiver og veileder omtrent annenhver uke, for å holde styr på fremdriften.

Det ble utarbeidet en wireframe tidlig i prosessen. Wireframen ble gjennomgått sammen med oppdragsgiver og veileder. Vi valgte å ikke bruke storyboards, siden det ikke var særlig relevant for dette prosjektet. Systemet er svært visuelt, og en wireframe var derfor et bedre alternativ for å kommunisere vår visjon av systemet.

Hvert team-medlem skrev daglige timelister med en kort beskrivelse av hva team-medlemmet jobbet med. Mot slutten av hver uke skrev teamet en statusrapport hvor hvert enkelt medlem skrev om hva som ble gjort den uken.

Vi produserte bare kode ettersom det trengtes, og fokuserte på oppfylle behov fra visjonsdokumentet med høyest prioritet først. Vi satte oss godt inn i problemdomenet i forkant av utvikling for å sikre kvalitet på kode og systemet i sin helhet. Vi var svært fleksible under utviklingsprosessen, noe som gjenspeiles i den jevnlige kommunikasjon vi hadde med oppgavestiller og måten vi tok til oss tilbakemelding.

Vi benyttet oss altså av en lean utviklingsmetode. Det fungerte godt i et lite team som jobbet tett på hverandre. Pandemien førte til at vi stort sett jobbet digitalt. Vi hadde også mulighet til å møtes på skolen noen få ganger hvis det var noe viktig som måtte gjøres. For eksempel vår del av underskrivingen av kontrakten med oppdragsgiver og NTNU.

Visualisering av EKG-data

I oppgaveteksten ble det spesifisert at: “Det viktigste her er å tenke nytt, men fremdeles forankre visualiseringen litt i dagens måte å visualisere EKG på slik at leger og kardiologer kan føle seg litt hjemme.” Med “dagens måte” tolker vi det som å presentere dataene i form av en tabell, hvor

hver celle inneholder et grafområde med hver kanal. Med ønsket fra oppgaveteksten i betraktning, kom vi fram til tre måter å visualisere EKG-dataene på. I den første ligger kanalene ved siden av hverandre. Den andre er vektorkardiografi, hvor det utføres transformasjoner på dataene for å lage tredimensjonale punkter. Vektorkardiografi er beskrevet nærmere i teorikapittelet. Videre i rapporten kaller vi disse metodene henholdsvis for ECG- og VCG-visualisering. En del av oppgaven var å tenke nytt, og i denne sammenhengen utviklet vi det vi kaller for sirkelvisualisering. Her er en kanal tegnet flere ganger i sirkler slik at dens verdier overlapper tidligere verdier.

I ECG-visualiseringen forsøkte vi å opprettholde ryddigheten av dagens metode. Kanalene ligger ved siden av hverandre i hvert sitt XY-plan med en distanse mellom. ECG-dataene har en standard for hvordan grafen representeres. Man kan se for seg grafen plassert i et rutenett. I X-aksen er hvert steg 0.04 sekund, og i Y-aksen er hvert steg 0.1 millivolt (Infiniwell AS, n.d.). Rutenettet i visualisering ble designet ut fra standardene for EKG. Rutenettet er også skalerbart. Hvis brukeren forlenger tidslinjen, vil også rutenettet forlenges.

I teorikapittelet introduserte vi tre ulike matrisetransformasjoner for vektorkardiografi presentert i boken *Computers in Cardiology* (Guillem et al., 2006, 252). Det er disse transformasjonene vi valgte å implementere.

For begge visualiseringene tenkte vi det ville være nyttig å kunne endre på fargene. For å kunne se bølgene i sammenheng med tiden, implementerte vi et fargevalg hvor fargene endrer seg over det valgte intervallet. Både ECG- og VCG-visualisering gir mulighet for denne metoden. Et annet fargevalg som er mulig for ECG, er farge basert på amplitude. Det siste fargevalget vi implementerte var også for ECG, og bruker grupperinger av såkalte “SNOMED CT Code” med tilknyttet farge for hver gruppering. “SNOMED CT Code” er en kode som identifiserer type annotering. Denne grupperingsfilen er utlevert av oppgavestiller.

Valg av teknologier

Implementasjon på ulike plattformer

Fra tidlig i prosjektet var planen å gjøre nettsiden tilgjengelig på mobil, som illustrert i wireframes som ligger i vedlegg [A](#). Vi opplevde problemer hvor linjer i 3D-visualiseringen ikke ble presentert riktig. Vi antok at dette var forårsaket av et avvik innenfor biblioteket for

3D-visualisering vi brukte. Etter å ha snakket med oppgavestiller ble vi i fellesskap enige om å fokusere på skrivebordsversjonen av nettsiden.

Brukergrensesnitt

Å utvikle en kompleks nettside med mange ulike komponenter kan være vanskelig hvis man ikke benytter seg av et rammeverk som deler opp koden og gjør det mer oversiktlig. Det var lett for oss å velge rammeverk siden vi allerede har brukt React i flere prosjekter tidligere, og er dermed godt kjent med det. Den største fordelene med å bruke et rammeverk som React, er at man kan dele opp en applikasjon (nettsiden) i flere ulike deler. Disse delene kalles komponenter og er gjenbrukbare. Dette gjør at man kan lage dynamiske nettsider som er meget aktuelt siden vi har mange ulike 3D-komponenter som inngår i visualiseringen. React gjør det mulig å håndtere mange komponenter på en oversiktlig og enkel måte, som er grunnen til at vi valgte å bruke dette rammeverket.

3D-visualisering

ThreeJS

En essensiell del av dette prosjektet er 3D-visualisering. Derfor var det viktig å velge rett API/bibliotek. Oppgaven som vi fikk tildelt spesifiserte tre slike biblioteker som kunne være aktuelle å bruke; OpenGL, WebGL, ThreeJS. Valget falt etterhvert på ThreeJS. Hovedgrunnen til at vi valgte ThreeJS ovenfor de andre var at ThreeJS er lettere å sette seg inn i hvis man ikke har noen forkunnskaper til OpenGL eller WebGL (Three.js Fundamentals, n.d.). WebGL er et bibliotek som er basert på OpenGL (Jackson, 2014). ThreeJS er et bibliotek som er bygget oppå WebGL (Three.js Fundamentals, n.d.). Det er altså mer abstrakt og enklere å ta i bruk ThreeJS for oss, siden vi har lite forkunnskaper med OpenGL og WebGL.

React-three-fiber

ThreeJS er et bibliotek for JS og kan brukes med rammeverket React, men man kan ikke bruke ThreeJS i komponenter slik som man bør i et React-prosjekt. React-three-fiber er et bibliotek som løser dette problemet ved å pakke inn ThreeJS-klasser inn i egne gjenbrukbare React komponenter (react-three-fiber, 2020). Dette gjør at man kan lage egne komponenter for 3D-visualisering som gjenbrukes i applikasjonen.

Forskjellen ligger i hvordan komponentene er bygd opp. Med kun ThreeJS vil mesteparten av logikken og koden ligge i egne livssyklusmetoder, og deretter kobles opp mot et HTML-element som er det eneste JSX i komponenten. Dette er ansett som dårlig struktur i React komponenter (Facebook Inc., n.d.). Med react-three-fiber kan man derimot bruke alle ThreeJS objekter direkte i JSX som egne komponenter. Dette gjør at komponentene blir enklere å lage, mer oversiktlig og gjør det lettere å finne eventuelle feil. Alt dette ligger til grunnlag for at vi valgte å ta i bruk react-three-fiber som et bindeledd mellom React med JSX, og ThreeJS med JS objekter.

Tilstandsbehandling

En viktig del av strukturen til et React-prosjekt er hvordan man håndterer tilstanden til ulike komponenter. Under utviklingsfasen av prosjektet så vi et behov for en global tilstandsbehandler. Dette er for å unngå å måtte sende tilstand fra en “forelder-komponent” nedover blant mange “barn-komponenter” for å få tilstanden i den aktuelle komponenten. Dette fører både til uoversiktlige komponenter, og spesielt til at alle “barn-komponenter” vil render på nytt hvis tilstanden endres i “foreldre-komponenten”. Det er ofte at man ønsker at en komponent skal render på nytt, men med mange komponenter som inneholder 3D-grafikk, vil man holde dette til et minimum for å oppnå god ytelse i applikasjonen.

Vi valgte derfor å bruke en rask, skalerbar og slank tilstandsbehandler kalt Zustand (*zustand*, n.d.). Dette biblioteket bruker React hooks for å lage “stores” som man kan ta i bruk i ulike komponenter, ved å importere den aktuelle “store”. Dette betyr at alle tilstander som flere komponenter skal benytte seg av, ligger i Store.js filen i ulike “stores”.

Et av målene med høy prioritet i visjonsdokumentet var å “spesifisere et tidsintervall på bølgeformen som skal visualiseres”. Vi ønsket at sluttbrukerne av systemet skulle kunne endre på tidsintervallet i sanntid ved å dra på tidslinjen. Dette kan være vanskelig siden systemet har mange komplekse sammenkoblede komponenter, som må oppdateres flere ganger i sekundet. En av de største grunnene til at vi tok i bruk Zustand fremfor andre tilstands-biblioteker, var at med Zustand kan man abonnere til endringer i tilstanden. Denne tilstanden kan lagres i en referanse (React ref) og oppdateres, uten at hele komponenten renderer på nytt, men kun de delene som trenger å oppdateres. Denne funksjonaliteten kalles “transient updates” i Zustand. Dette har en enorm forskjell i ytelse for 3D-komponentene. Man kan dermed oppdatere blant annet start og

slutt tid, til flere 3D-komponenter og oppdatere disse 60 ganger i sekunder (med useFrame metoden fra react-three-fiber), uten at det har negativ effekt på ytelsen.

Brukertestning

Til brukertestning konstruerte vi et sett med oppgaver til å gi testpersonene samt et skjema vi ønsket de skulle fylle ut. For vår del var det viktig å få tilbakemelding på det medisinske aspektet ved problemstillingen vår. Vi var derfor avhengig av å utføre brukertester hvor testpersonene hadde kunnskap innenfor kardiologi, altså personer innenfor målgruppen vår. Vi var usikre på om vi ville komme i kontakt med et nok antall personer i målgruppen. Derfor utformet vi brukertestene slik at personer utenfor målgruppen kunne gi oss tilbakemelding på systemets grad av brukervennlighet og hvor intuitivt det var.

For å sette opp oppgavene tok vi utgangspunkt i en mal for et observatørskjema fra forfatterne av boken *Praktisk brukertestning* gitt i sammenheng med øving 10 fra faget TDAT2003 (Øving 10 (MMI) - Brukertest, n.d.). Observatørskjemaet består av tolv oppgaver som går gjennom den grunnleggende funksjonaliteten av systemet. Den siste oppgaven gir testpersonen to minutter til å utforske systemet som den selv vil. På denne måten vil testpersonen kunne fordøye dens inntrykk av systemet uten å oppleve stress av oppgavene.

Etter brukeren hadde utført oppgavene, ga vi dem et skjema vi ønsket de skulle fylle ut. Skjemaet inneholder fire spørsmål hvor de to første gir testpersonen mulighet til å fortelle hva den likte, og hva den ikke likte ved systemet. Disse er rettet mot alle typer testpersoner. De siste to spørsmålene er rettet i større grad mot målgruppen. Gjennom disse spørsmålene fikk vi direkte tilbakemelding på selve visualiseringen av EKG-data.

Oppdragsgiver slet med å skaffe fagpersoner til brukertestning. Vi forsøkte å nå ut til hjerteavdelingen ved St. Olavs Hospital, men vi fikk ingen svar uten å få noe svar. Vi klarte derimot å få tak i relevante brukere fra Nordlandssykehuset.

Universell utforming

Dette systemet er rettet mot leger og helsearbeidere. Etter avtale med oppgavestiller ble vi enige om å fokusere på desktop-implementasjon, som betyr at nettsiden ikke er optimalisert for smarttelefoner og andre mobile enheter.

Vi implementerte ingen verktøy for tolkning av informasjon på nettsiden, som for eksempel text-to-speech. Slike spesifikke verktøy tenkte vi kan brukes eksternt for å tolke informasjonen på nettsiden. Det er derimot ikke mulig å kunne bruke slikt for å tolke det som befinner seg i 3D-visualiseringen. Systemet er svært tilpasselig når det gjelder farger, i tilfelle noen av brukerne skulle være fargeblinde. Vi har derfor gjort det mulig å endre farger på bakgrunnen og bølgene.

Oppfattelighet

Nettsiden har tekst-alternativer på knapper og andre viktige funksjoner som befinner seg i GUI. Dette er nyttig hvis man for eksempel bruker text-to-speech eller lignende løsninger. Det er lett å forstå at man er inne på menyen siden bakgrunnen utenfor menyen blir mørkere når menyen er åpen. Vi gjorde det slik at knapper som ikke består av noe tekst, har såkalte tooltips knyttet til seg. Dette gjør at brukerne kan forstå funksjonen av en knapp, uten å måtte prøve den.

Brukbarhet

Systemet kan brukes med tastatur, det er derimot noen begrensninger. De viktigste funksjonene er tilgjengelige ved å bla gjennom med tab-knappen på tastaturet. Når HTML-elementene får fokus, vil de markeres, slik at man kan se hvor på siden man befinner seg. Vi forsøkte å gjøre alle funksjonaliteter tilgjengelig på denne måten, men noen komponenter fikk vi ikke til å gjøre fokuserbar. Av funksjonaliteter innenfor 3D-visualiseringen, er det mulig å bevege seg rundt, men ikke rotere. Det var heller ikke mulig for oss å implementere alternativer for bruk av datamus som foregår innenfor visualiseringen, som for eksempel markering for å legge til annoteringer. Man kan derimot legge til annoteringer fra menyen.

Man kan endre på fargene til bølgene basert på hvilken fargemodus man har valgt. Det er ikke mulig å endre fargene når man har valgt diagnose-fargemodus. Fargen til bakgrunnen til 3D-visualiseringen kan også endres. Dette gjør at man kan endre denne sammen med bølgene slik at man oppnår god kontrast. Det kan være personer som er svaksynte, eller er fargeblinde og dermed trenger god kontrast for å skille bølgene fra bakgrunnen.

Forståelighet

God leselighet av informasjonen er essensielt for en god brukeropplevelse. Teksten i menyene og resten av GUI-en står i god kontrast med bakgrunnen. Standard-fargene på bølgene i

visualiseringen har god kontrast, og medfører god leselighet. Teksten har en passelig størrelse, og er plassert slik at det er enkelt å lese.

Hele nettsiden er på engelsk, slik at brukerne skal kunne forholde seg til bare ett språk. Menyen er inndelt i faner, som gjør det lettere å finne fram til de innstillingene brukeren ønsker å benytte seg av.

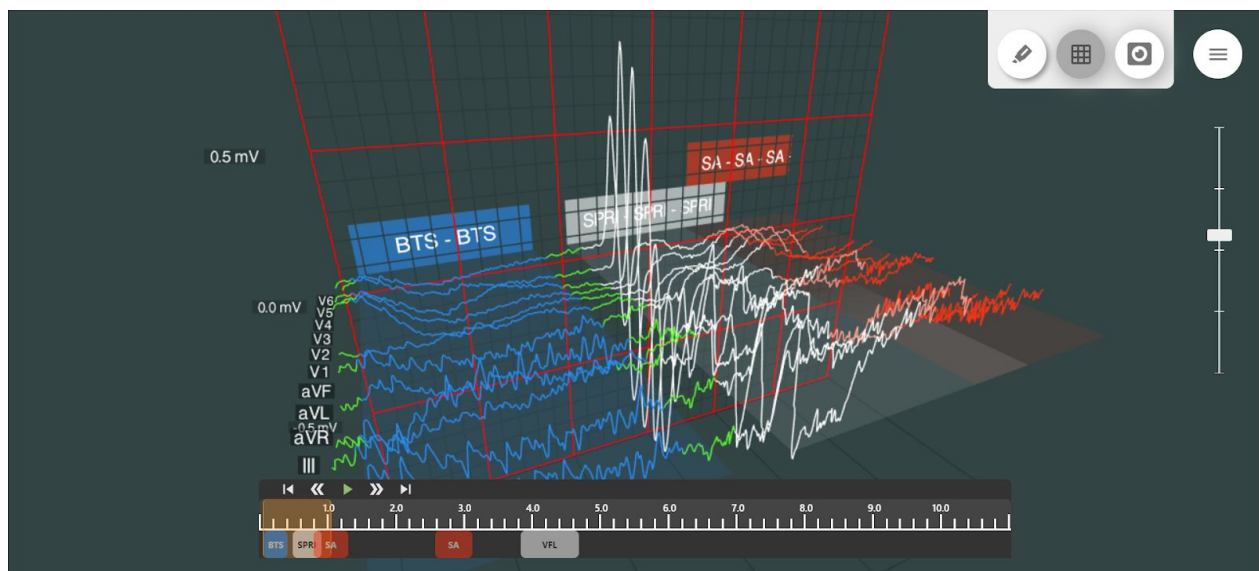
Input-assistanse er implementert for å gjøre det lettere for brukeren å navigere gjennom systemet, og bruke de forskjellige funksjonene. Knapper tilknyttet funksjonaliteter som kan slås av og på forandrer design. På denne måten forstår brukeren om funksjonaliteten er aktiv. Når brukeren utfører forskjellige oppgaver som å lage en ny annotering, får brukeren informative tekstbokser for assistanse. Ved filopplastning får brukeren en feilmelding hvis fil-typen ikke stemmer. Det er også mulig å se et enkelt eksempel for hvordan JSON-filene skal være strukturert. Når brukeren er i ferd med å utføre irreversible handlinger, vil det dukke opp en dialogboks som ber om videre bekreftelse. Et eksempel er hvis man laster opp en EKG-fil, vil systemet advare brukeren om at eksisterende annoteringer vil slettes.

Kapittel 4: Resultater

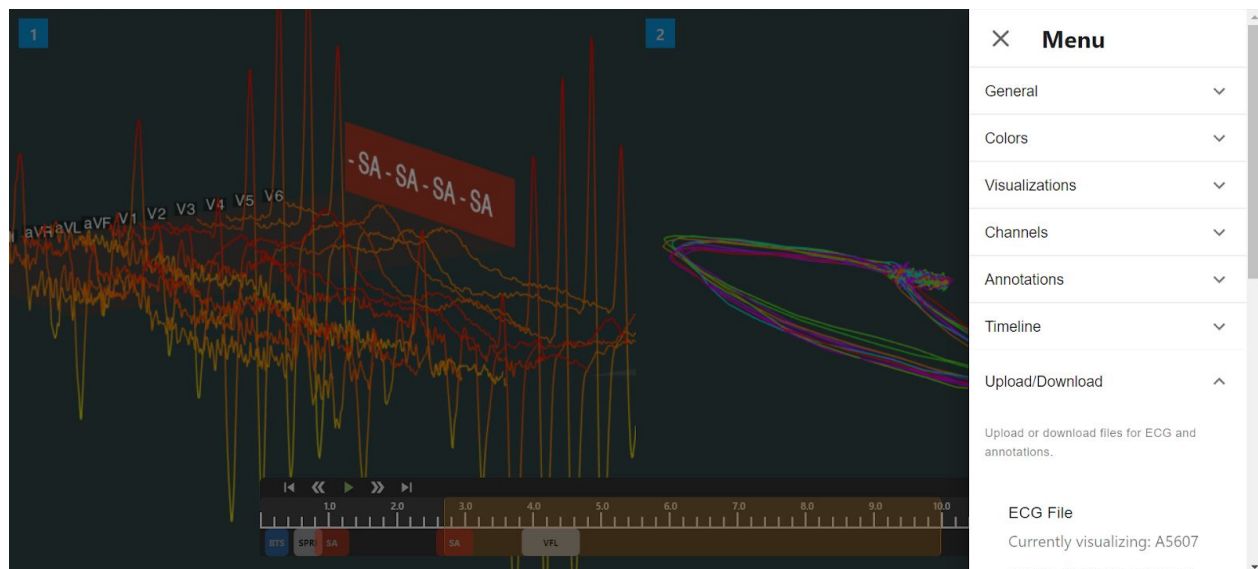
Ingeniørfaglige resultater

Produktet

Sluttresultatet er en nettside som kjører på desktop. Nettsiden inneholder flere 3D-visualiseringer av EKG-data, samt verktøy og innstillinger for å endre/manipulere denne visuelle representasjonen. Man kan blant annet endre start og slutt tiden til representasjonen ved å bruke tidslinjen, legge til annoteringer, endre farger, og se punkt-verdier ved å inspisere enkelte kanaler. Nettsiden skal være et fullstendig system for å analysere EKG-data i 3D. Nedenfor er det presentert to skjermbilder av systemet.



Skjermbilde av nettsiden hvor bølgene er farget basert på diagnosegrupper. Her er rutenettet slått på.



Skjerm bilde av nettsiden hvor menyen er åpen. I bakgrunnen er to visualiseringer framstilt samtidig.

Mål og status

Målene for prosjektet er utarbeidet i visjonsdokumentet. I tabellen under er hvert mål listet opp med status ved leveringstidspunktet.

Mål	Prioritet	Fullført	Status
Lett tilgjengelig	Høy	Ja	Systemet er tilgjengelig som en nettside.
Markere deler av visualiseringen	Høy	Ja	Klikk og dra-markering av deler av visualiseringen.
Legge til annoteringer på bølgeformer	Høy	Ja	Annoteringer legges til på markerte områder i visualiseringen. Området kan også spesifiseres i menyen.
Slette og endre annoteringer	Middels	Delvis	Ingen funksjonalitet for å slette eller endre hver enkelt annotering. Annoteringer slettes ved å laste opp ny fil med annoteringer.
Navigere til annotering	Middels	Ja	Listen med annoteringer, og annoteringer i tidslinjen kan trykkes på for å navigere til den valgte annoteringen.

Lese AI-annoteringer	Middels	Ja	AI-annoteringer lastes opp gjennom fil.
Endre på tidslinjen (Navigere fram og tilbake i tid)	Høy	Ja	Horisontal tidslinje med utvidet funksjonalitet..
Spesifisere et tidsintervall på bølgeformen som skal visualiseres	Høy	Ja	Brukeren kan justere tidslinjen for å velge tidsintervallet.
Zoome inn og ut på visualiseringen	Høy	Ja	Vertikal scrollbar. Kan også bruke touch-funksjoner på mobilskjerm eller touchpad.
Rotere på visualiseringen	Høy	Ja	Visualiseringen kan roteres.
Filtrere bort kanaler	Høy	Ja	Trykke på kanalen og inspisere denne nærmere. Liste over kanaler som visualiseres, hvor man kan velge bort de man ønsker.
Laste opp fil med EKG-bølger	Lav	Ja	GUI for opplasting av en slik fil. Filen erstatter den eksisterende filen.
Laste opp fil med annoteringer	Lav	Ja	GUI for opplasting av en slik fil. Opplastede annoteringer legges til sammen med de eksisterende.
Lagre annoteringer til fil	Høy	Ja	AI- og bruker-annoteringer lagres separat.

Brukertester

Vi utførte til sammen seks brukertester, hvor fire av testpersonene var personer innenfor målgruppen. I tabellen under er yrke og alder til hver testperson tilknyttet deres bruker-ID.

Bruker-ID	Alder	Yrke
1	55	Sykepleier
2	52	Intensivsykepleier på hjertemedisinsk klinikk
3	22	Student
4	46	Sykepleier og jobber på sengepost med hjertepasienter

5	24	Lærer
6	66	Pensjonert sykepleier med utdannelse i hjerteintensiv

Under er de viktigste observasjonene fra brukertestene listet opp. Disse gikk igjen for flere brukere.

- Zoomer bare med zoombaren.
- Ingen problemer med å slå på rutenett og ortografisk kamera.
- Mye forvirring i løpet av de første oppgavene.
- Vanskeligheter med å forstå forskjell mellom verktøyslinje og hovedmeny.
- Forsøker å filtrere bort en annotering ved å lage en ny annotering over den.

Svarene brukerne hadde på spørsmålene etter brukertestene er presentert under.

Spørsmål	Bruker-ID (M: Målgruppe)	Svar
Spørsmål 1: Hva likte du av systemet?	1 (M)	At man kan forstørre å se nærmere på komplekser At man kan måle avstander lett
	2 (M)	Tilsynelatende god grafikkoppløsning ved forstørring. Ville sannsynligvis kommet ganske raskt inn i bruken av programmet hvis jeg fikk "leke" med det en stund.
	3	Layouten er fin. Siden ser bra ut estetisk.
	4 (M)	Klare tydelige farger godt markert med alle avledninger. Bra zoome funksjon.
	5	Jeg likte at det var godt visualisert.
	6 (M)	<i>Ingen svar</i>
Spørsmål 2: Hva likte du ikke av systemet?	1 (M)	Vanskelig å si, skjønte ikke helt vcg.
	2 (M)	Burde være mulig å starte/stoppe måling av forskjellige tidsintervaller ved å f.eks. høyre/venstre-klikke på musa.

	3	Var veldig mye å forstå, mye å lese. Jeg foretrekker at det er lettvin, men forstår at kompleksiteten kan være nødvendig for leger. Det var vanskelig å skjønne hva verktøylinja var. Symbol for markeringsmodus er ikke intuitivt. Burde heller vært en firkant men en mindre firkant inni med en musepil som trekker i den minste firkanten.
	4 (M)	Vanskelig å forstå enkelte av oppgaver da jeg ikke fant de i hovedmeny.
	5	Jeg likte ikke knappene for filtrering av annotering.
	6 (M)	<i>Ingen svar</i>
Spørsmål 3: Hva synes du om måten EKG-dataene ble visualisert på? Fikk du den informasjon du ønsket? Var det oversiktlig eller vanskelig å forstå?	1 (M)	Ja det synes jeg var informativt Litt vanskelig første gang, men tror det blir mye lettere når man har prøvd flere ganger.
	2 (M)	Så ut til å være en del støy/artefakter på ekstremitetsavledninger (I,II,III, aVf, aVr, aVI). Så ingen tydelig P-bølge i noen avledninger. Som tidligere nevnt så det ut til å være ganske høyoppløselig grafikk.
	3	Grafene så bra ut, men jeg forstår ikke fargene, hva symboliserer det?
	4 (M)	Sikkert ikke så vanskelig om en hadde fått opplæring i å bruke systemet. Det jeg syntes var vanskeligst var å forstå knapp øverst til høyre som jeg tolket som en hovedmeny.
	5	Den var fin.
	6 (M)	Har jobbet med EKG i 40 år både på hjerteovervåking og analysert tusenvis av holtere siden 1983, men bare brukt EKG i 2D, så dette ble ikke så interessant for meg. Kanskje hadde det vært litt mer oversiktlig hvis det var en skikkelig sinusrytme på skjermen.
Spørsmål 4: Ville du brukt dette systemet ovenfor et som visualiserer i 2D? Hvorfor?	1 (M)	Jo hvis jeg hadde blitt mer vant/kjent med systemet og litt sikrere i bruken. Synes det var informativt
	2 (M)	Vanskelig å si, umiddelbart ser jeg ingen fordeler med 3D framstilling av 12-avlednings EKG. Bruker

		pr i dag et system fra Schiller Medilog (Darwin Echo View) hvor komplekser løpende plasseres etter hverandre i en slags 3D-fremstilling. Dette gir et godt oversiktsbilde av endringer i hjerterytme i en tidsakse (eks. Atrieflimmer, hyppige VES, m.m.).
	3	Ja. Det kan være nyttig å sammenligne grafene samtidig. Så kan man innsistere en enkelt graf, hvis man må se nærmere. Det er mer effektivt, spesielt hvis man skal se på 12 grafer samtidig.
	4 (M)	Ikke noe klart svar på dette.
	5	Ja dette er nok sikkert mye mer oversiktlig.
	6 (M)	<i>Ingen svar</i>

Administrative resultater

Tidsforbruk

I Gantt-diagrammet, som ligger i prosjekthåndboka, estimerte vi at planleggingsfasen kom til å ta 40 timer, men vi brukte bare litt i overkant av halvparten av denne allokerede tiden med 24 timer.

Utviklingsfasen vår besto av fire deler; 3D-visualisering, GUI, brukertesting og bug fixing.

3D-visualiseringen tok ikke like lang tid som forventet, med 167,5 av 183 timer. Det samme gjelder for GUI med 200 av 220 timer. Brukertesting tok derimot over tre ganger så lang tid som planlagt, med 10 timer planlagt og 31 timer brukt. Vi planla kun 20 timer til bug fixing, men brukte totalt 82,5 timer på denne aktiviteten.

Utviklingsfasen i dette prosjektet tok totalt 512 timer.

Tabellen under viser faktisk tidsforbruk for hver aktivitet og som en total sum for hver del-fase i Gantt-diagrammet.

Aktivitet	Total akk. SUM
Planlegging	
Wireframes	7,0t
UML	10,0t
Visjonsdokument	8,0t
Oppstartsmøter	5,0t
Total	30,0t
3D-visualisering	
Forskning på 3D-visualiserings API	13,0t
Valg av 3D-API	3,0t
3D-EKG visualisering av EKG-bølger	136,0t
Eksperimentelle 3D-visualiseringer	15,5t
Total	167,5t
GUI	
Innstillinger	70,0t
Tidslinje	58,0t
Grid	72,0t
Total	200,0t
Brukertesting	
Utforme oppgaver til brukertest	20,0t
Utføre brukertester	11,0t
Total	31,0t
Bug fixing	
Fikse bugs	82,5t
Total	82,5t
Dokumentasjon	
Systemdokumentasjon	31,0t
Hovedrapport	70,5t
Presentasjon	4,0t
Total	105,5t
Møter	
Møter	16,0t
Total	16,0t

Kapittel 5: Diskusjon

Det er to aspekter ved problemstillingen vår. Det ene er i hvor stor grad det tekniske gjør det mulig å kunne visualisere EKG-dataene. Det andre er hvor gunstig det er i forhold til klinisk analyse. Det tekniske aspektet er vi som ingeniørstudenter i stand til å vurdere selv. Dette handler om verktøyene som trengs for å oppnå en oversiktlig visualisering er til stede. For å vurdere det kliniske aspektet var vi derfor svært avhengig av å motta tilbakemelding fra eksterne personer med relevant kunnskap innenfor kardiologi.

Brukertest

Etter å ha gjennomført brukertester på personer med og uten relevant kompetanse, har vi fått gode tilbakemeldinger på systemet vi har utviklet. Først og fremst var de fleste fornøyde med selve designet på nettsiden, men en bruker syntes det var mye å fordøye fra nettsiden, og dermed opplevde den som noe komplisert. Noen av tilbakemeldingene var direkte rettet mot uklarheter ved designet, som vi tok til oss fortløpende og løste før neste brukertest tok sted. Et eksempel på dette er bruker 3, som ikke oppfattet symbolet for markeringsmodus som intuitivt. Symbolet var da en merkelapp, og vi endret det til en markeringstusj.

Vi observerte at alle utenom én, tok i bruk zoom-baren når de skulle utføre oppgaven som omhandlet å zoome inn og ut på visualiseringen. Vi anser derfor denne funksjonaliteten som forsterkende for brukervennligheten, da det kan være unaturlig for enkelte å bruke scroll-hjulet for å zoome. Få av brukerne hadde problemer med å aktivere rutenett og ortografisk kamera, selv om de hadde vanskeligheter med å forstå forskjellen mellom verktøyslinja og hovedmenyen. Disse oppgavene var mot slutten av oppgavesettet, som betyr at brukerne hadde fått litt tid til å forstå systemet og utforsket litt. Vi observerte at flere brukere beveget musepekeren over knappene for å lese tooltipsene, og det er mulig at de husket funksjonalitetene fra disse fra tidligere oppgaver. Etter at brukerne hadde lagt til en annotering, ba vi dem om å filtrere denne bort. Bruker 2, 3 og 6 forsøkte å markere over den nye annoteringen for å filtrere den bort. Løsningen var derimot å gjøre det i menyen. Vi som utviklere var ikke innom tanken om at funksjonaliteten kunne misforstås på denne måten. Dette tilfellet poengterer viktigheten av brukertester.

De fleste deltakerne i brukertestene hadde en generell oppfatning at de hadde forstått systemet bedre og muligens sett fordelene med det, hvis de hadde fått mer tid til å utforske nettsiden og fått litt opplæring. Etter første brukertest la vi til en ekstra oppgave hvor brukerne fikk utforske systemet som de selv ønsket på to minutter. Det er tydelig at dette ikke var nok tid basert på noen av tilbakemeldingene fra brukertestene, men vi ønsket ikke bruke for mye tid av testpersonene våre. Resultatene fra disse brukertestene skal være et grunnlag for å komme fram til en konklusjon og et endelig svar på vår problemstilling. Det kan derimot være vanskelig å komme fram til en konklusjon når brukerne har ulike meninger. Da må man heller vurdere om man skal vekte meningen til de ulike brukerne basert på kompetansen og erfaringen de innehar.

Klinisk tilbakemelding fra målgruppen

Bruker 1, 2, 4 og 6 er dem som innehar relevant kompetanse for å kunne vurdere nyttheten av 3D-visualisering av EKG-data. Alle disse brukerne har erfaring med å analysere EKG-data gjennom yrket deres. Bruker 2 er ganske klar i tilbakemeldingen sin; personen ser ingen fordeler med å fremstille 12-avlednings EKG-data i 3D. Bruker 4 og 6 har ingen tydelige svar på dette.

Bruker 1 og 6 opplevde EKG-dataene som ble presentert som irrelevante. Bruker 1 sa under brukertesten at: “Bølgene ser meningsløse ut”. Vi ser samme problemet gjennom svaret fra bruker 6 på spørsmål 3: “Kanskje hadde det vært litt mer oversiktlig hvis det var en skikkelig sinusrytme på skjermen”. Bruker 2 svarer på spørsmål 2 at den merket støy ved ekstremitetsavledningene. Brukeren så heller ikke noen P-bølger i noen av avledningene. Disse tilbakemeldingene tyder på at datafilene vi mottok fra Infiniwell ikke gjenspeiler realistiske data som målgruppen forholder seg til i arbeidslivet. Dette kan bety at innholdet i datafilene som ble brukt forhindret testpersonene å gi tilbakemelding på visualiseringsmetodene. Hadde EKG-dataene inneholdt for eksempel sinusbølger eller P-bølger, er det mulig at testpersonene kunne vurdere hvor gunstig det var å tolke slike egenskaper ved bølgene gjennom visualiseringen.

Testpersonene som befant seg i målgruppen var over 40 år. Gjennom observasjonene så vi at i flere av dem ikke var særlig teknisk anlagte. Det kan ha en påvirkning for resultatet av brukertestene. Fra brukertest 6 har vi sitatet “Dette er for litt spesielt interesserte”. Bruker 4 var redd for å ødelegge noe ved bruk av systemet. Det er mulig at enkelte av brukerne var skeptisk til systemet, som gjorde at de fikk en negativ oppfattelse.

Noen av tilbakemeldingene var vanskelig å forstå, siden vi ikke innehar nødvendig kompetanse. Bruker 1 nevnte blant annet at det var positivt at man kunne forstørre og se nærmere på komplekser. Nøyaktig hva dette betyr og innebærer er vanskelig for oss å si noe om. Vi har derimot gjort vårt beste for å ta til oss tilbakemeldingene fra brukertestene til å vurdere hvor gunstig systemet er.

Krav og mangler

Vi har fullført omtrent alle kravene som vi utarbeidet og fikk godkjent av oppdragsgiver i visjonsdokumentet. Det eneste vi kan påpeke er at man ikke kan endre på eller slette en enkelt annotering. Hvis man laster opp en ny fil så vil alle annoteringene slettes, men det finnes ingen funksjonalitet for å slette en annotering som allerede er lagd eller lastet opp. Grunnen til at denne funksjonaliteten ikke ble implementert var at vi hadde begrenset med tid, og hadde andre mål som var prioritert høyere. Dessuten så kom det ikke spesifikt fram fra oppdragsgiver under møtene at denne funksjonaliteten burde prioriteres fremfor andre arbeidsoppgaver.

Svakheter

Vi er fornøyd med løsningen vi har utviklet, men vi legger ikke skjul på at den har noen svakheter. En av disse svakhetene vi har avdekket er bruken av store datafiler. Systemet vårt er ikke optimalisert for slike store datafiler når det kommer til ytelse og design. Selve prosesseringen av datafilen kan ta lang tid, avhengig av maskinvaren som kjører nettsiden. I tillegg så blir avstanden mellom tidene i tidslinjen veldig kompakt.

En annen svakhet er ytelsen når man bruker avspillingsmodus. Tanken bak denne modusen var at man skulle kunne la datafilen avspilles uten å måtte endre på tidslinjen. Dette fungerer slik som løsningen er i dag, men det kan virke litt tregt eller hakkete. Grunnen til at dette skjer er at man oppdaterer start- og slutt-tid, fremfor å bevege selve grafen. Man får derfor kun oppdateringer etterhvert som nye punkter kalkuleres, og ikke en jevn bevegelse.

Oppgaven vi fikk fra oppdragsgiver i Infiniwell gikk i stor grad ut på å eksperimentere med 3D-visualisering av EKG-data. Vi har få visualiseringer som vi ville kalt eksperimentelle, i den forstand at selve visualiseringen ikke er utbredt eller kjent i fagmiljøet. Visualiseringen med EKG-bølger i 3D er kanskje ikke nødvendigvis mye brukt blant personer med kompetanse, men de har kjennskap til dette siden det er en 3D representasjon av noe som vanligvis vises i 2D.

Vektorkardiografi-visualiseringen (VCG) er litt mer eksperimentell, men den er også kjent i fagmiljøet.

Sirkelvisualiseringen er visualiseringen hvor vi virkelig prøvde noe eksperimentelt. Tanken var at en slik visualisering kunne brukes til å undersøke perioden på enkelte utslag på bølgeformene som repeterte seg selv. Under demonstrasjon av denne funksjonen på et møte sa oppgavestiller at han ikke kunne tenke på noen umiddelbar praktisk fordel med en slik visualisering. Vi valgte derfor å fokusere mindre på denne. Dette var et eksempel der vi forsøkte å komme opp med nye måter å visualisere slik data på. Vanskeligheten med en slik prosess er at vi som påtroppende dataingeniører mangler kunnskap innenfor kardiologi til å tenke oss frem til hvilke egenskaper ved EKG-data som bør fremheves gjennom en visualisering. En annen faktor som spilte en rolle i denne sammenhengen, var at vi ikke hadde noen med faglig kompetanse som kunne styre oss i riktig retning på eventuelle andre eksperimentelle visualiseringer. På grunnlaget av disse faktorene så valgte vi å fokusere mer på ECG- og VCG-visualiseringen, og heller legge til mer funksjonalitet for å manipulere disse.

Styrker

På grunnlag av våre egne meninger, og tilbakemeldingen fra brukertestene, kan vi trekke frem noen styrker ved prosjektet. Helt fra starten har vi hatt fokus på brukergrensesnitt og design. Det var viktig for oss at nettsiden var funksjonell, ryddig og oversiktlig. Tilbakemeldingene fra brukertestene poengterte dette.

I tillegg har vi hatt stort fokus på tilpasning. Vi ønsket at man skulle kunne tilpasse systemet slik som man selv ønsket, og det er derfor vi har lagt til så mange innstillinger. Selv om vi ikke fikk spesifikke tilbakemeldinger på graden av tilpasning anser vi det selv som at det vil gjøre brukeropplevelsen bedre i det lange løp.

Mot slutten av prosjektet stilte vi oppdragsgiver spørsmålet: “Fikk oppdragsgiver det som var forventet?”. Svaret er gitt under.

Mht svar på spørsmål så er det et klart «ja, jeg fikk det jeg forventet og mere». Når vi startet prosjektet så var vi jo ikke sikker på hva som skulle skje mht. 3D rendering av EKG; det var jo et slags utforskningsprosjekt. Det er ikke mye å se på nettet om 3D EKG rendering så dette er jo et relativt nytt område. Jeg er veldig fornøyd med det dere oppnådde og føler at vi har en god

baseline vi kan vise til som startpunkt for videre visualiseringsarbeid senere. Det er trist at vi ikke kunne få klinikere med på en videreutvikling, men sånn gikk det denne gang. Jeg har ihvertfall fremdeles tro på at prosjektet deres er et såkorn til noe større. I denne omgang så kom vi så langt vi kunne uten innspill fra klinikere så vi bør være veldig fornøyde. (Sandbekkhaug, 10.12.2020)

Basert på tilbakemeldingen fra oppdragsgiver er det tydelig at han ser potensialet i systemet. Det er mulig vi hadde fått til et system mer relevant for målgruppen hvis klinikere var del av utviklingsprosessen fra starten av. Vi er enige med oppgavestiller 3D-visualisering av EKG-bølgeformer er et relativt utforsket felt, og trenger eksperimentering.

Vi er glad for at vi valgte å bruke biblioteket “react-three-fiber”, siden dette gjorde at vi kunne ta i bruk tidligere erfaring og kunnskap med React inn i dette prosjektet. Vi tror det hadde vært utfordrende å organisere koden hvis vi ikke hadde brukt et rammeverk slik som React. Grunnen til dette er det høye antallet komponenter og den komplekse logikken som ligger i grunn for systemet. Ved å ha koden delt opp med et slikt hierarki bestående av komponenter, er det enkelt å finne fram til hvor eventuelle feil har oppstått.

Tidsforbruk

Vi synes selv at vi har gjort gode estimeringer i forhold til tidsforbruk for ulike aktiviteter, men noen av estimeringene var ikke helt nøyaktige hvis man ser på faktisk tidsforbruk. Vi hadde en tilnærming til prosjektet hvor det var viktig å planlegge godt og grundig, men også viktig å komme raskt i gang med utviklingsfasen. Det oppsto enkelte problemer underveis som vi måtte håndtere. Disse var ikke en del av planen som ble utarbeidet i starten.

En av aktivitetene som vi estimerte mest feil på var bug fixing. Grunnen er at under planleggingen tok vi ikke hensyn til at vi måtte utføre bug fixing underveis i prosjektet. De fleste timene med bug fixing var underveis i prosjektet, og ikke etter at vi kjørte brukertestene. Det viser seg derimot at bug fixing etter at vi startet å kjøre brukertester i uke 48. er nært estimeringen med 19,5 av 20 timer.

Gruppearbeidet

Arbeidet med prosjektet har gått fint hele veien. Det foregikk for det meste digitalt på grunn av pandemien. I begynnelsen av prosjektet kunne vi møtes fysisk på skolen for å jobbe sammen. Etter som smittespredningen økte utover høsten, ble det nesten kun digital jobbing. Vi er glad for

at vi kunne gjennomføre planleggingsfasen fysisk. Å jobbe digitalt har gått veldig greit. Gruppemedlemmene har jobbet mye sammen før, og vi har hatt god flyt i arbeidet.

Kapittel 6: Konklusjon og videre arbeid

Det er ikke et tydelig svar på om 3D-visualisering av EKG-bølgeformer er gunstig.

Testpersonene var skeptiske til systemet, og flere var usikre når det gjaldt det å ta i bruk et slikt system. En grunn til dette kan være at systemet er svært ulikt det de er vant med fra før.

Resultatet fra brukertestene tilsier at systemet ikke er et gunstig verktøy for EKG-analyse. De fleste sier derimot at mer opplæring i systemet kunne gitt dem bedre oversikt. Det ble gjennomført svært få brukertester enn det som var ønsket. Dette gjorde at resultatene ikke ble særlig nøyaktige.

Målet med prosjektet var ikke å komme fram til et system hvor 3D-visualiseringen er gunstig, men heller undersøke og eksperimentere feltet. Derfor er det like nyttig for oppgavestiller om det er gunstig, som at det ikke er det. Vår vurdering av systemet som helhet, og resultatene fra brukertestene, er at 3D-visualisering av EKG-bølgeformer ikke er gunstigere enn dagens løsning. Feltet er derimot utforsket og det er fortsatt muligheter for å ta prosjektet videre.

Vårt valgt av systemutviklingsmetode viste seg å være fornuftig. Vi er positive på at dette var en effektiv metode, med tanke på tilstandene. Nesten alle krav i visjonsdokumentet ble oppfylt. Oppdragsgiver er svært fornøyd med resultatet, noe teamet også er.

Videre arbeid i prosjektet ville omhandlet å fikse svakhetene som vi avdekket. Noe av arbeidet ville derfor ha vært å optimalisere ytelsen til systemet. I tillegg kunne det være nyttig å undersøke flere eksperimentelle og ukjente måter å visualisere EKG-data på. Enda mer brukertesting og mer tilbakemelding på systemet fra målgruppen ville vært hensiktsmessig for å gi et bedre svar på problemstillingen.

Referanser

Arnesen, H. (2018, oktober 29). *EKG*. Store norske leksikon. Retrieved desember 7, 2020, from <https://sml.snl.no/EKG>

Arnesen, H. (2019, mai 1). *Vektorkardiografi*. Store norske leksikon. Retrieved desember 8, 2020, from <https://sml.snl.no/vektorkardiografi>

Discord Inc. (2020). *Discord*. <http://discord.com>

Facebook Inc. (n.d.). *Thinking in React*. <https://reactjs.org/docs/thinking-in-react.html>

Guillem, M. S., Sahakian, A. V., & Swiryn, S. (2006). Derivation of orthogonal leads from the 12-lead ECG. accuracy of a single transform for the derivation of atrial and ventricular waves. In *Computers in Cardiology* (pp. 249-252). <http://cinc.org/archives/2006/pdf/0249.pdf>

Infiniwell AS. (n.d.). *EKG Standard Måleenheter*.

Jackson, D. (2014). *WebGL Specification*. <https://www.khronos.org/registry/webgl/specs/1.0/>

Observatørskjema. (n.d.). In *Praktisk brukertesting*.

react-three-fiber. (2020). <https://inspiring-wiles-b4ffe0.netlify.app/>

Three.js Fundamentals. (n.d.). Retrieved Desember 4, 2020, from <https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html>

zustand. (n.d.). <https://github.com/pmndrs/zustand>

Vedlegg

Prosjekthåndboken ligger i en separat fil etter ønske fra veileder.

A - Kravdokumentasjon

Vedlegg A

3CG
Kravdokumentasjon

Versjon 1.2

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
14.09.2020	1.0	Første utkast	Dalheim, William Kalstad, Mikael Tronstad, Henrik
30.11.2020	1.1	Revidert versjon	Dalheim, William Kalstad, Mikael Tronstad, Henrik
06.12.2020	1.2	Oppdatert Wireframes til dem vist i første oppfølgingsmøte.	Dalheim, William

Innholdsfortegnelse

Introduksjon	A4
User Stories	A4
Domenemodell	A5
Sekvensdiagrammer	A6
Prototyper	A7
Wireframes	A7
Åpningsstilstand	A8
Meny åpnet	A8
Laste opp EKG-fil	A9
EKG-fil lastet opp - meny åpen	A9
EKG-fil lastet opp - meny lukket	A9
Alternativ hvor meny ligger på bunnen i desktop	A10
Alternativ med hamburgermeny på mobil - krav om landskapsmodus	A10
Meny åpen på mobil med hamburgermeny	A11
Referanser	A11

1. Introduksjon

Dette dokumentet er skrevet for å få på plass alle funksjonelle og ikke-funksjonelle krav for systemet. Hensikten er å få oversikten over de funksjonene systemet skal ha, og få en grundig forståelse av hva systemet skal inneholde. Dokumentet inneholder user stories, domenemodeller og prototyper.

2. User Stories

User story 1:

Som bruker
Ønsker jeg å laste opp en EKG-fil
Slik at systemet kan 3D-visualisere EKG-grafene.

Krav:

- Gyldig fil-type.

User story 2:

Som bruker
Ønsker jeg å rotere, zoome og flytte rundt EKG-grafene i 3D
Slik at brukeren kan studere EKG-resultatene.

Krav:

- Gyldig fil må være lastet opp.

User story 3:

Som bruker
Ønsker jeg å endre hvilke EKG-kanaler som skal vises
Slik at brukeren kan se nærmere på enkelte kanaler.

Krav:

- Gyldig fil må være lastet opp.

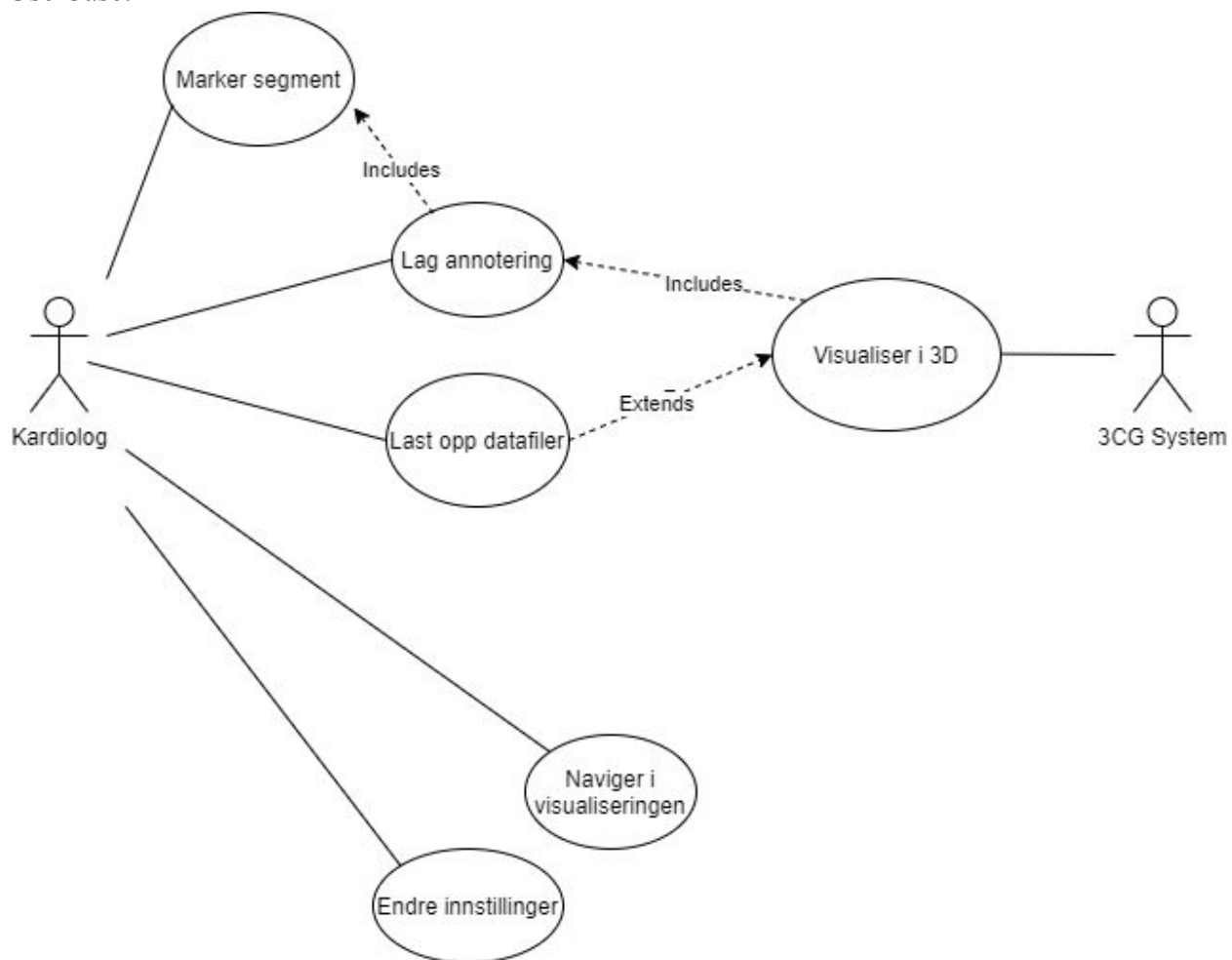
User story 4:

Som bruker
Ønsker jeg å annotere grafene
Slik at brukeren kan annotere grafene.

Krav:

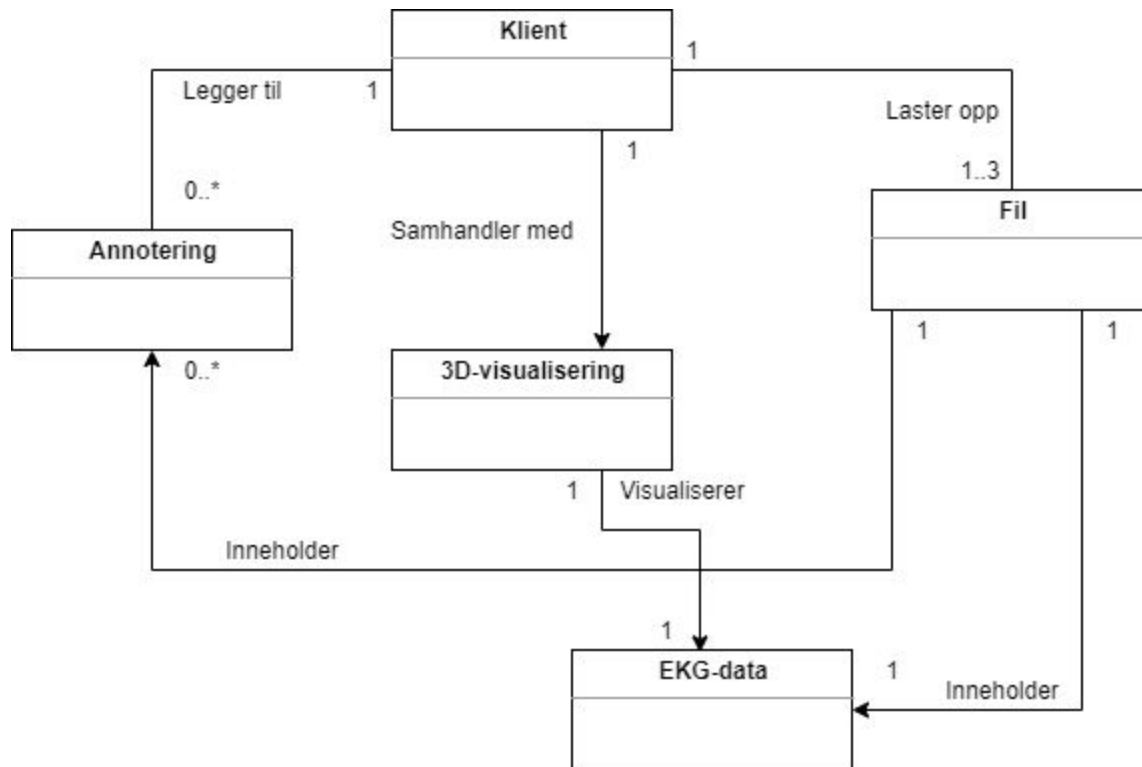
- Gyldig fil må være lastet opp.

Use Case:



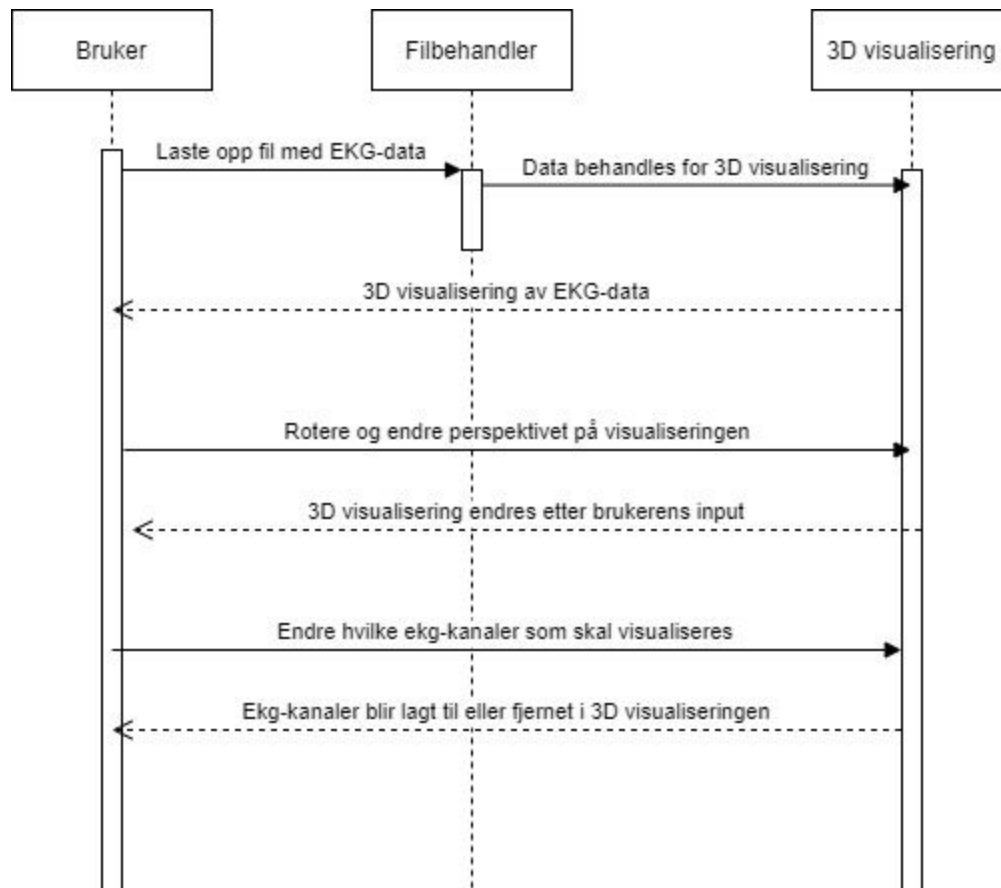
3. Domenemodell

Domenemodell 1



3.1 Sekvensdiagrammer

Sekvensdiagram 1

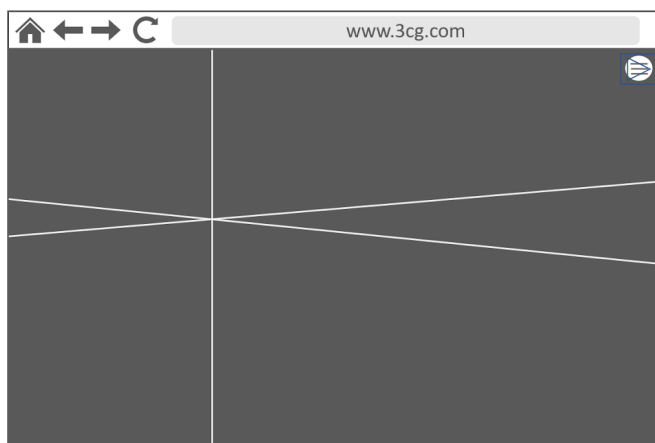
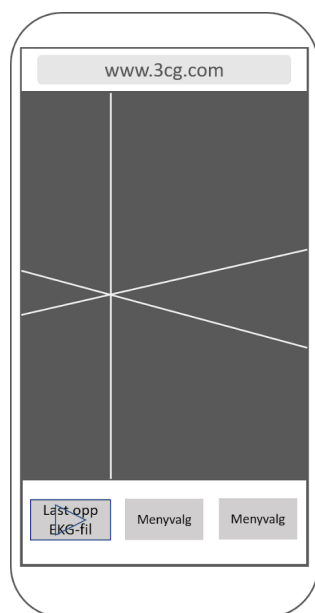


4. Prototyper

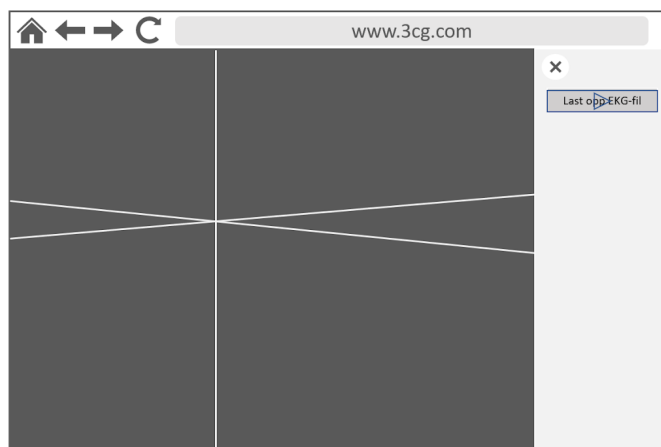
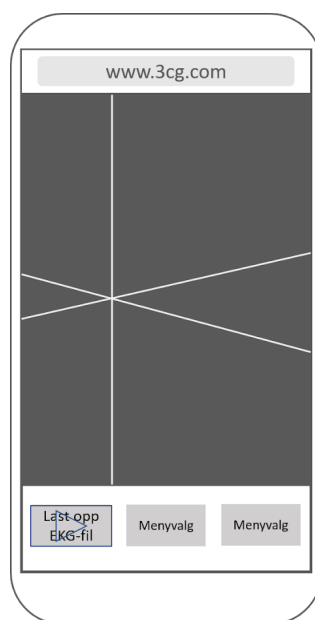
4.1 Wireframes

Wireframes er konstruert i PowerPoint (Microsoft, n.d.).

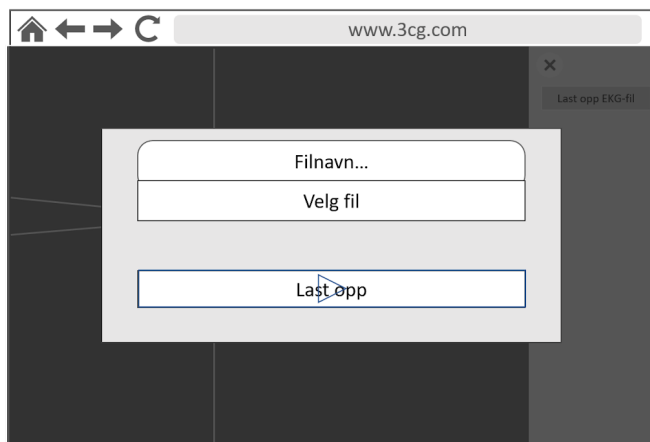
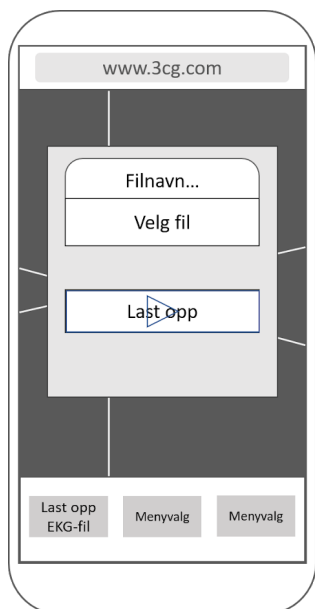
Åpningsstilstand



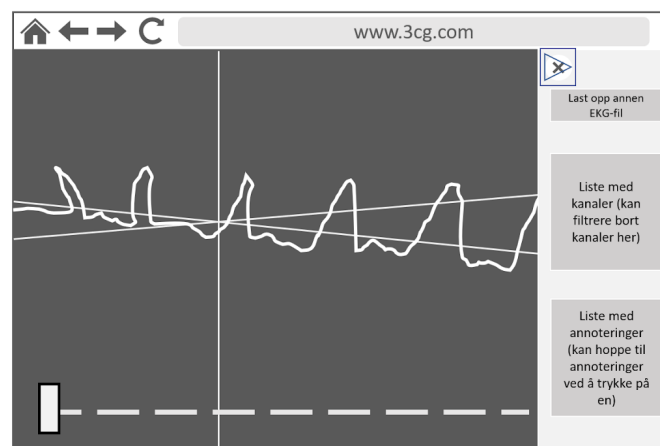
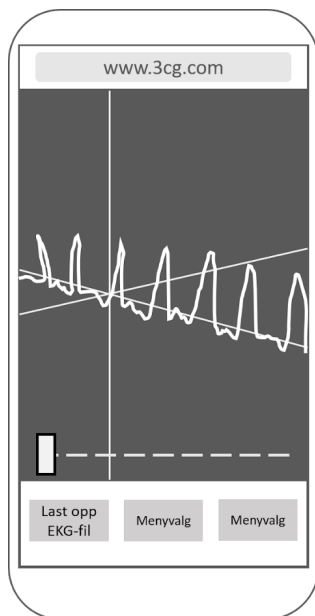
Meny åpnet



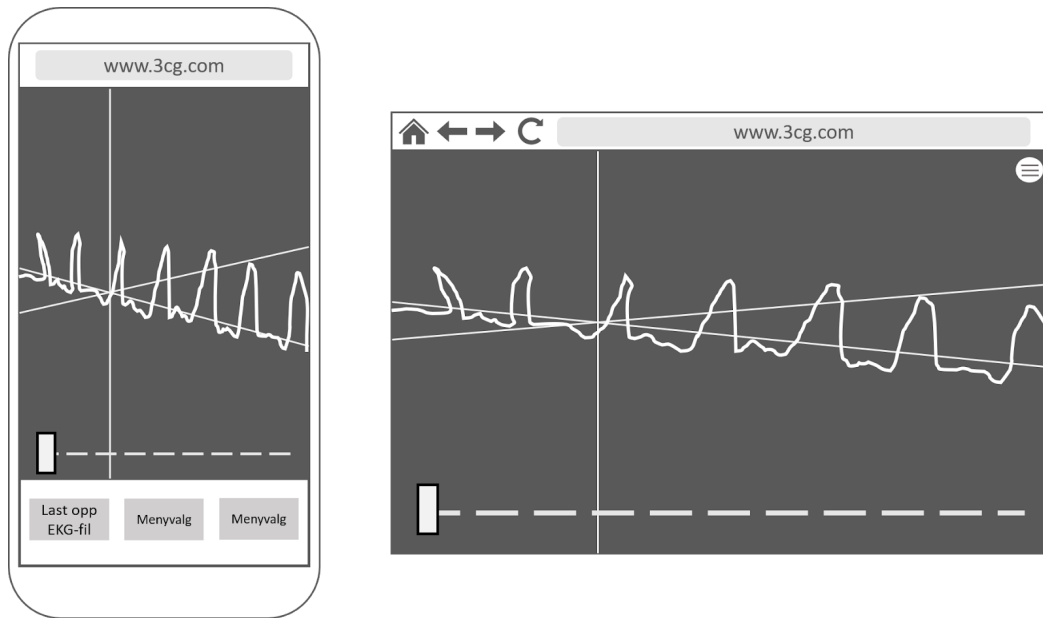
Laste opp EKG-fil



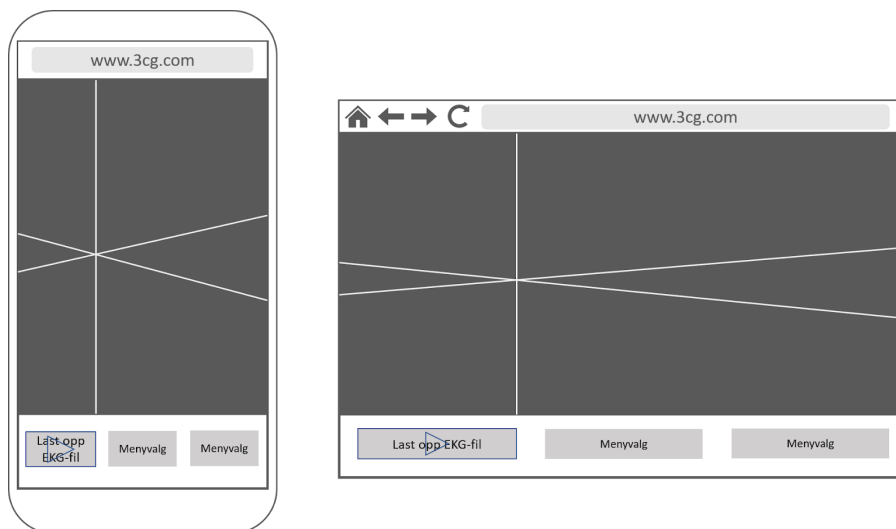
EKG-fil lastet opp - meny åpen



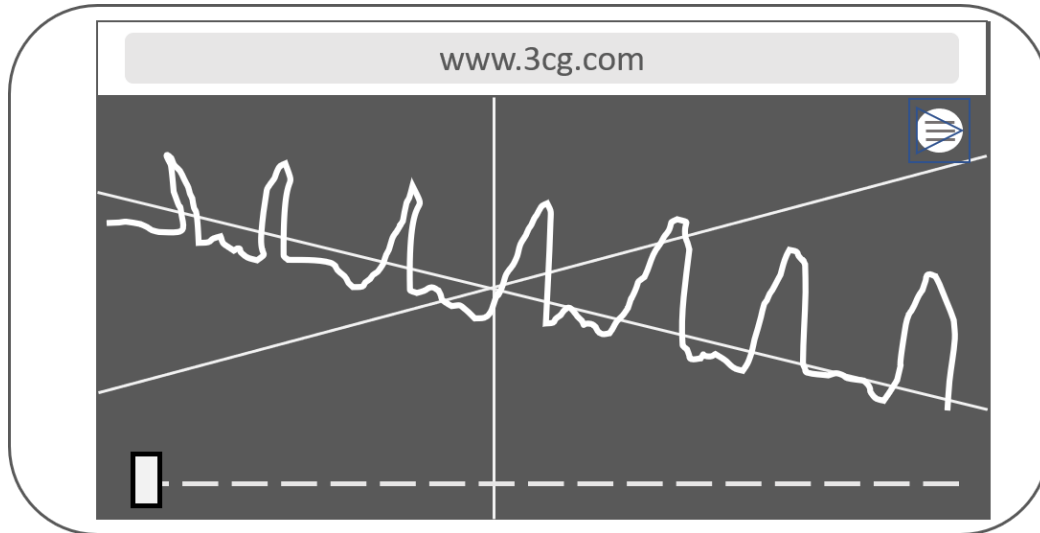
EKG-fil lastet opp - meny lukket



Alternativ hvor meny ligger på bunnen i desktop



Alternativ med hamburgermeny på mobil - krav om landskapsmodus



Meny åpen på mobil med hamburgermeny



5. Referanser

Microsoft. (n.d.). *Microsoft PowerPoint*. <https://www.microsoft.com/en-us/microsoft-365/powerpoint>

B - Systemdokumentasjon

3D-visualisering av EKG-bølgeformer

Systemdokumentasjon

Versjon 1.2

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
21.09.20	1.0	Første utkast	Dalheim, William Kalstad, Mikael Tronstad, Henrik
30.11.2020	1.1	Revidert versjon.	Dalheim, William Kalstad, Mikael Tronstad, Henrik
07.12.2020	1.2	Revidert versjon.	Dalheim, William Kalstad, Mikael Tronstad, Henrik

Innholdsfortegnelse

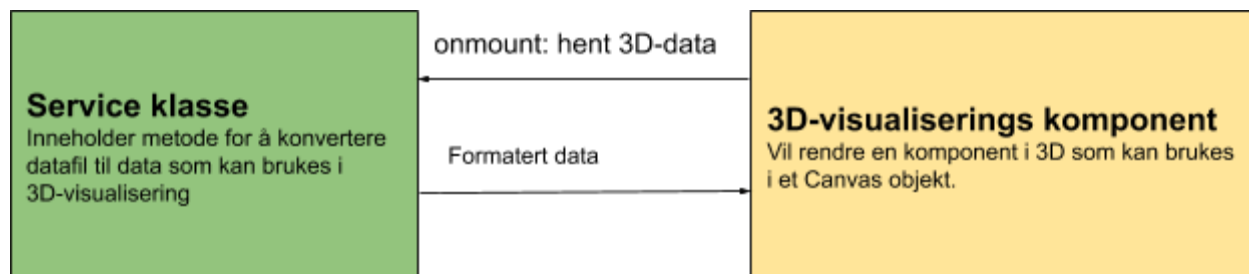
Introduksjon	B4
Arkitektur	B4
Prosjektstruktur	B5
Klassediagram	B8
Databasemodell	B10
Server-tjenester	B10
Sikkerhet	B11
Installasjon og kjøring	B11
Installasjonsveiledning	B12
Dokumentasjon av kildekode	B12
Kontinuerlig integrasjon og testing	B12
Oppsett for kontinuerlig integrasjon og kontinuerlig distribusjon	B13
Tester	B14
Hvordan kjøre testene	B14
Referanser	B14

Introduksjon

I dette prosjektet har gruppen laget et system som 3D-visualiserer EKG-data. Prosjektet er gjennomført for oppdragsgiver Infiniwell. Dette dokumentet er skrevet for å gi en oversikt over systemet som er laget. Dokumentet inneholder en beskrivelse av hvordan systemet strukturelt er bygget opp, og hvorfor vi har valgt å gjøre det slikt.

Arkitektur

Kobling mellom Serviceklasse og 3D-visualisering



Denne figuren viser strukturen for hvordan de viktigste komponentene samhandler.

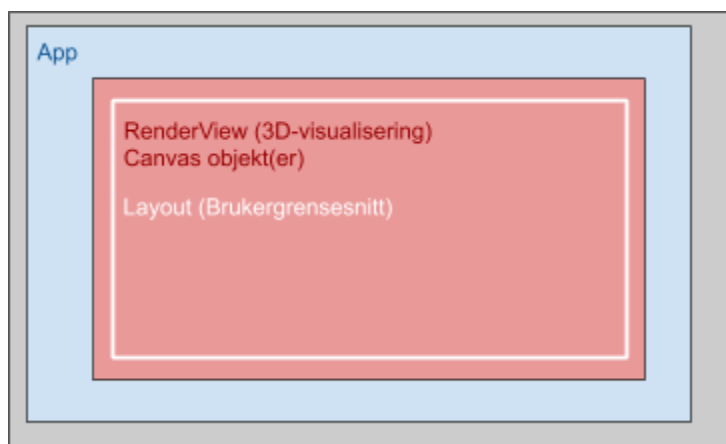
Service-klassen konverterer data som brukes til å visualisere dataen i 3D-visualiserings-komponenten.

Prosjektstruktu

Prosjektet er laget med javascript front-end biblioteket React. Dette biblioteket bygger på ideologien med bruk av gjenbrukbare komponenter. Prosjektstrukturen er derfor delt inn i ulike komponenter, i tillegg til andre filer som går utenfor front-end.

Front-end komponenter

Vi har delt opp komponentene i to hovedkategorier; grafikk (som regel 3D) komponenter som ligger innenfor Canvas objektet(ene), og komponenter (brukergrensesnitt) som ligger utenfor "oppå" Canvas. Grunnen til dette er at alle komponenter som skal brukes i et Canvas objekt, må være grafikk komponenter. Altså en "vanlig" React-komponent med html kan ikke brukes i et Canvas objekt.



Komponentene er grovt sett delt inn slik:

- **UI:** Komponenter som omhandler brukergrensesnittet som ligger “ovenfor” selve 3D-visualiseringen.
- **CanvasComponents:** Inneholder komponenter som brukes i selve 3D-visualiseringen.

Noen viktige komponenter:

- **App:** Foreldrekomponenten som står over alle andre komponenter.
- **Layout:** Denne komponenten samler alt av brukergrensesnitt som skal legges “oppå” Canvas objekt(ene).
- **RenderView:** Samling av alle Canvas som brukes i applikasjonen. Denne og Layout komponenten er de to komponentene høyst opp i prosjektstrukturen etter App som omhandler front-end.
- **ECG:** 3D-rendering av 12-kanals EKG signaler.
- **VCG:** 3D-rendering av 12-kanals EKG signaler med en transformasjon til et vektor kardiogram.

Andre prosjektfiler/mapper

En kort beskrivelse av andre viktige filer/mapper i prosjektet strukturen:

- **res:** lokalt data for testing og som brukes som standard når brukeren åpner applikasjonen. De fleste filene har vi fått tildelt fra Infiniwell. Resten er lagd som eksempler av oss i teamet.
- **Scripts:** Denne mappen inneholder kun en script fil som kalkulerer fargene for

3D-visualiseringene i formatet som støttes av 3D biblioteket vi har brukt.

- **Services:** Klasser med tilhørende metoder for lesing/skriving og behandling av dataobjekter hentet fra filer.
- **tests:** Inneholder tester for ulike deler av applikasjonen.
- **utils:** Andre funksjoner som ikke passer inn i noen andre mapper.

“State management”/tilstands behandler

En viktig del av strukturen til et React prosjekt er hvordan man håndterer tilstanden til ulike komponenter. Det viste seg fort i starten av prosjektet at en global tilstands behandler ville være nødvendig. Dette er for å unngå å måtte sende tilstand fra en “foreldre-komponent” nedover blant mange “barn-komponenter” for å få tilstanden i den aktuelle komponenten. Dette fører både til uoversiktlige komponenter, og desto viktigere til at alle “barn-komponenter” vil “rendre” på nytt hvis tilstanden endres i “foreldre-komponenten”. Det er ofte at man ønsker at en komponent skal “rendre” på nytt, men med mange komponenter som inneholder 3D-grafikk, vil man holde dette til et minimum for å oppnå god ytelse i applikasjonen.

Vi valgte derfor å bruke en rask og lettvekts tilstands behandler kalt Zustand. Dette biblioteket bruker React hooks for å lage “Stores” som man kan “Subscribe” til i ulike komponenter, bare ved å importere den aktuelle “Store” fra Store.js filen. Dette betyr at alle tilstander som flere komponenter skal benytte seg av, ligger i Store.js filen i ulike “Stores”.

“Transient updates”

En av de største grunnene til at vi tok i bruk Zustand fremfor andre tilstands-biblioteker, var at med Zustand kan man “Subscribe” til endringer i tilstanden uten at komponenten “renderer” på nytt. Dette har en enorm forskjell i ytelse for 3D-komponentene. Man kan dermed oppdatere start og slutt tid til flere 3D-komponenter og oppdatere disse 60 ganger i sekunder (med useFrame metoden), uten at React “renderer” disse komponentene hele tiden.

```
// Fetch initial time state
const startTimeRef = useRef(useTimeStore.getState().startTime);
const endTimeRef = useRef(useTimeStore.getState().endTime);

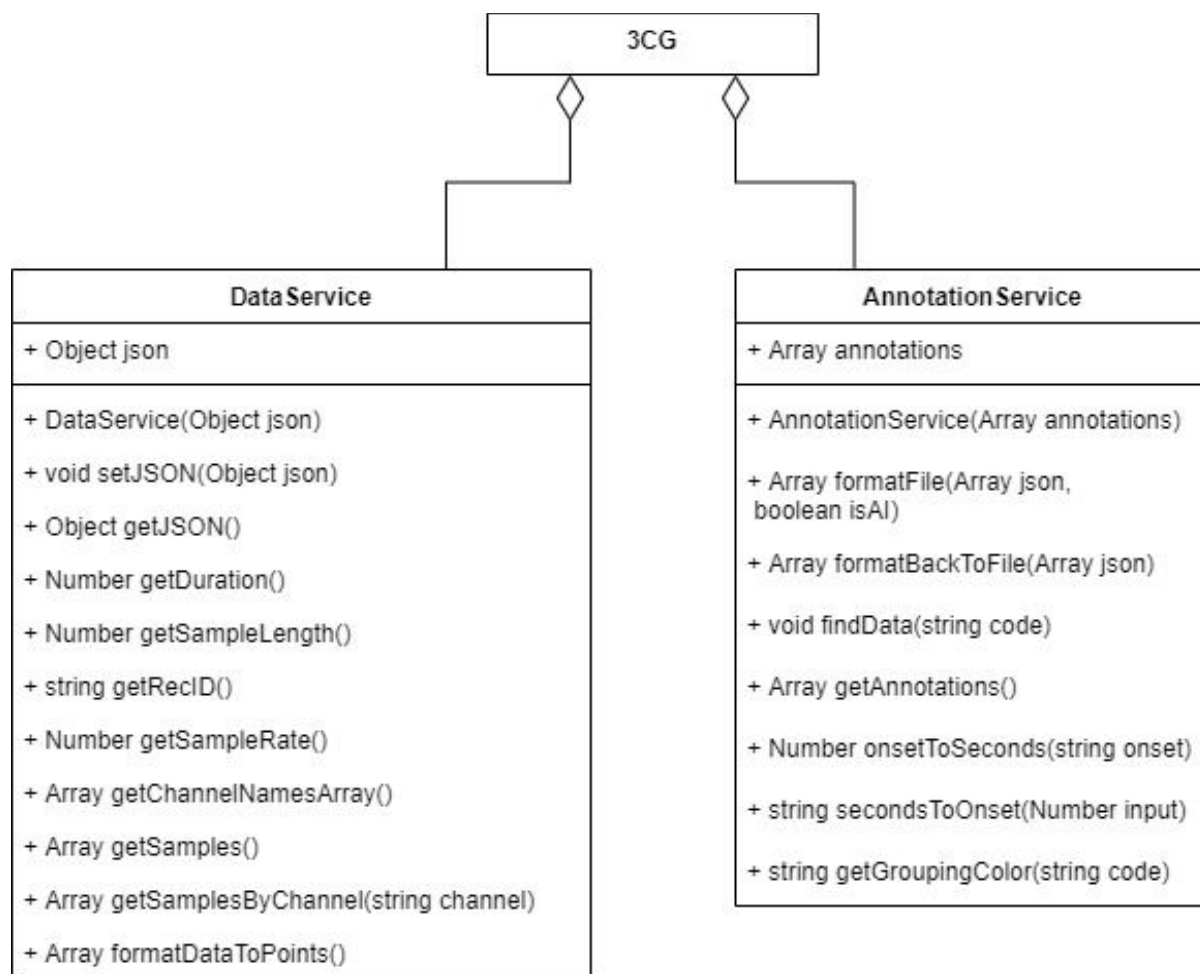
// Connect to the store on mount, disconnect on unmount, catch state-changes in a reference
useEffect(() => {
  useTimeStore.subscribe(
    (startTime) => (startTimeRef.current = startTime),
    (state) => state.startTime
  );

  useTimeStore.subscribe(
    (endTime) => (endTimeRef.current = endTime),
    (state) => state.endTime
  );
}, []);
```

Eksempel på "transient-updates" i prosjektet. Her hentes start og slutt tid fra "Store" uten at komponenten vil "rendre" på nytt.

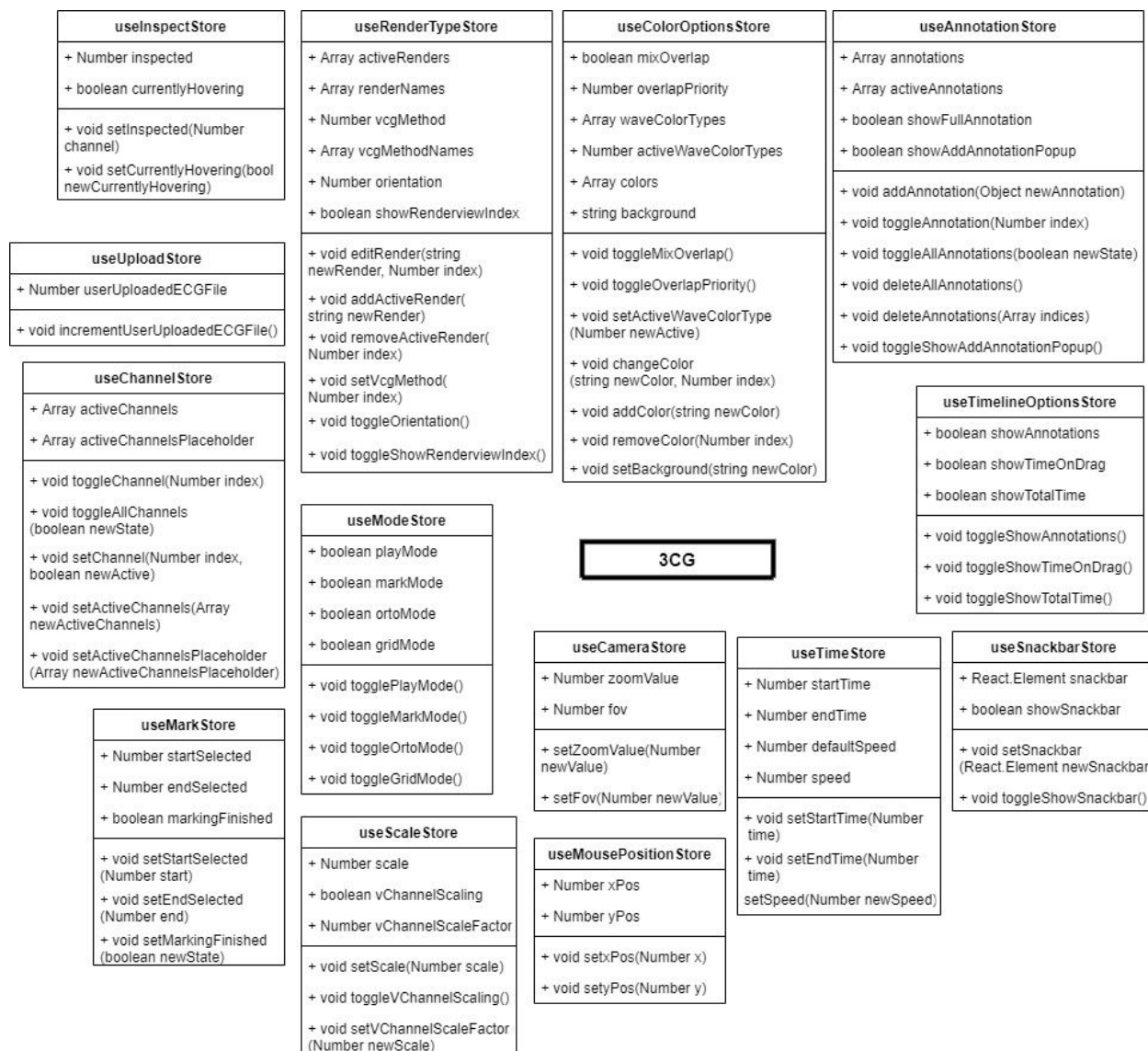
Klassediagram

Serviceklassene brukes til å skape et grensesnitt mellom filer og variabler. Det er to serviceklasser, AnnotationService og DataService. Begge tar inn en variabel som er en tolket JSON-struktur. Klassene inneholder da metoder for å hente ut attributter som befant seg i JSON-filen.



Klassediagram over serviceklasser

Store-objektene vi konstruerte med biblioteket Zustand kan også ses på som klasser. Alle disse klassene aggregerer til 3CG som befinner seg i midten, men disse pilene vises ikke for å gjøre figuren mer oversiktlig.



Databasemodell

Databasemodell er ikke aktuelt siden Infiniwell allerede har en løsning som systemet vil kunne integreres med hvis det er aktuelt. Slik som systemet er i dag er alt av data lagret på lokale filer som brukeren selv laster opp. Kobling til en database er utenfor omfanget til dette prosjektet.

Server-tjenester

Dataene systemet bruker er lokale filer distribuert av Infiniwell. Ved en integrasjon av systemet

med Infiniwells systemer vil dataene fås fra deres databaser. Slik som systemet står i dag er det ingen bruk for servertjenester, dette er dessuten utenfor prosjektets rammer, med hensyn til tiden allokert.

Sikkerhet

Systemet vi har utviklet har ingen form for innlogging eller sensitiv informasjon som hentes fra databaser. Brukeren vil selv laste opp datafiler og kjøre dette lokalt. Derfor er ikke sikkerhet i fokus for dette prosjektet. Det er altså ingen passord som blir overført eller lagret, ingen tokens lagret i localStorage eller cache, ingen SQL-spørringer som kan inneholde svakheter og ingen digitale sertifikater som trengs.

Installasjon og kjøring

De viktigste avhengighetene som er brukt er listet opp under

- NPM
 - Trengs for å laste ned bibliotekene. Hvis man har dette installert, vil den ta seg av alle de andre bibliotekene som trengs for å kjøre applikasjonen.
- React
 - Brukes for utvikling av brukergrensesnitt
- Three.js
 - Brukes for visualisering i tre dimensjoner

Installasjonsveiledning

1. Last ned NPM og Node.js fra <https://www.npmjs.com/get-npm>
2. Klon prosjektet i en mappe:

```
git clone https://gitlab.stud.idi.ntnu.no/williad/su3-3cg.git
```
3. Bytt til repositoryet:

```
cd su3-3cg
```
4. Last ned bibliotekene:

```
npm install
```

5. Kjør nettsiden:

```
npm start
```

Kildekoden er også bygget og lastet opp til en GitLab pages nettside som kan brukes uten noe form for installasjon. Brukeren trenger kun linken til denne nettsiden for å kjøre applikasjonen i nettleseren.

Link til GitLab pages nettside: <http://williad.pages.stud.idi.ntnu.no/su3-3cg/>

Dokumentasjon av kildekode

Her kan dere beskrive og vise til dokumentasjon av kildekode. Ta med hvordan en eventuelt genererer og viser dokumentasjonen.

Kommenter og vis til brukseksempler også om det er aktuelt. For eksempel, i tilfeller hvor en har laget frittstående biblioteker som en del av løsningen, beskriv og vis til eksempel bruk av disse bibliotekene.

Kildekoden inneholder kommentarer for de fleste komponentene. I de viktigste komponentene går kommentarene i nærmere detalj.

Kontinuerlig integrasjon og testing

Oppsett for kontinuerlig integrasjon og kontinuerlig distribusjon

Vi brukte GitLabs løsning for kontinuerlig integrasjon. Vi kjører Node-tester med JEST i Docker. For kontinuerlig distribusjon bruker vi GitLab Pages. I denne sammenhengen hadde vi følgende .yml-fil.

```
image: node:alpine

cache:
  paths:
    - node_modules/

stages:
```

```

- test
- deploy

variables:
  PUBLIC_URL: /su3-3cg

before_script:
  - rm -rf build
  - CI=false npm install

test:
  stage: test
  script:
    - npm test

pages:
  stage: deploy
  script:
    - CI=false npm run build
    - cp build/index.html build/404.html
    - mv public _public
    - mv build public

artifacts:
  paths:
    - public
only:
  - master

```

CI/CD-opplettet vårt består av to steg. Den første, test er CI-delen av oppsettet, og deploy er CD-delen. Før begge disse stegene fjernes mappen build, og bibliotekene lastes ned gjennom npm. I teststeget utføres testene. I deploy-steget bygges prosjektet og build-mappen plasseres i mappen public. På denne måten blir produktet tilgjengelig gjennom GitLab Pages.

Tester

Testene som er laget tester serviceklassene og noen store-objekter. Av serviceklassene testes alle metodene, deriblant metodene som konverterer mellom strukturen brukt i filene fra Infiniwell, og strukturen som brukes i vårt system. Noen av store-objektene inneholder bare variabler, og set-metoder for disse. Noen andre har metoder som inneholder logikk, og det er disse som kjøres tester på.

Det er ikke utført noen tester på front-end delen av systemet. Det er mulig med tester av brukergrensesnittet som er ren HTML med for eksempel bruk av biblioteket Enzyme. Dette gjorde vi ikke, da menyene og inputtene i systemet er enkle. Testing av 3D-visualiseringen er derimot ikke gjort som integrasjonstesting eller enhetstesting. Denne delen av systemet er testet gjennom utforskende og destruktive tester ettersom nye funksjonaliteter ble lagt til. Den er også testet gjennom brukertester.

Hvordan kjøre testene

Først må man følge installasjonsveiledningen fram til og med punkt 4. Etter dette skriver man følgende for å kjøre enhetstestene:

```
npm test
```

Referanser

Dette dokumentet er basert på malen for systemdokumentasjon av Eidheim, Ole Christian og Tesdal, Nils.

C - Visjonsdokument

Vedlegg C
3D visualisering av EKG bølgeformer

Visjonsdokument

Versjon <1.2>

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
09.09.2020	1.0	Første utkast	Dalheim, William Kalstad, Mikael Tronstad, Henrik
15.09.2020	1.1	Lagt inn funksjonelle egenskaper	Sandbekkhaug, Odd
16.09.2020	1.2	Behov, brukermiljø	Dalheim, William Kalstad, Mikael Tronstad, Henrik

Innholdsfortegnelse

1.	Innledning	C4
2.	Sammendrag problem og produkt	C4
2.1	Problemsammendrag	C4
2.2	Produktsammendrag	C4
3.	Overordnet beskrivelse av interessenter og brukere	C5
3.1	Oppsummering interessenter	C5
3.2	Oppsummering brukere	C6
3.3	Brukermiljøet	C6
3.4	Sammendrag av brukernes behov	C6
3.5	Alternativer til vårt produkt	C8
4.	Produktoversikt	C8
4.1	Produktets rolle i brukermiljøet	C8
4.2	Forutsetninger og avhengigheter	C9
5.	Produktets funksjonelle egenskaper	C9
6.	Ikke-funksjonelle egenskaper og andre krav	C11
7.	Referanser	C11

1. Innledning

Dokumentet beskriver krav til oppgaven “3D-visualisering av EKG-bølgeformer” gitt i faget TDAT3022 Systemutviklingsprosjekt høstsemesteret 2020. Hensikten med dokumentet er å konkretisere tankene rundt produktet som skal utvikles.

Visjonsdokumentet utarbeides i samarbeid mellom utviklere og produkteier slik at alle parter blir enige om hvilke funksjoner som skal implementeres og krav som må være oppfylt.

2. Sammendrag problem og produkt

2.1 Problemsammendrag

Problem med	lite detaljerte visualiseringer av EKG-bølger.
berører	kardiologer og leger
som resultat av dette	er det vanskelig og tidskrevende å tolke EKG-resultater
en vellykket løsning vil	gi enkel tilgang til en webtjeneste hvor EKG-bølger visualiseres i tre dimensjoner

2.2 Produktsammendrag

For	Infiniwell
som	har behov for en ny måte å visualisere EKG-bølger
3CG	er et nettbasert system
som	visualiserer EKG-bølger i tre dimensjoner og gir mulighet for å skrive notater på dem, samt AI-analyse
I motsetning til	dagens system som har visualisering i to dimensjoner
Har vårt produkt	visualisering i tre dimensjoner.

3. Overordnet beskrivelse av interessenter og brukere

3.1 Oppsummering interessenter

Navn	Utdypende beskrivelse	Rolle under utviklingen
Infiniwell	Selskapet ønsker å integrere 3D-visualisering i deres eksisterende løsning.	Veiledning og tilgang til datasett

NTNU	Universitetet ønsker å dyrke kunnskap og gi studenter erfaring.	Tilbyr studentene et arbeidsmiljø
------	---	-----------------------------------

3.2 Oppsummering brukere

Navn	Utdypende beskrivelse	Rolle under utviklingen	Representert av
Kardiologer og leger	Analysere visualiseringer av bølgeformer og diagnostisere ut fra disse	Tilbakemelding gjennom brukertester	Seg selv
Generelle helsearbeidere	Vurdere resultatene ved hjelp av en lettere forståelig visualisering	Tilbakemelding gjennom brukertester	Seg selv

3.3 Brukermiljøet

Brukermiljøet vil være på sykehus og andre helsestasjoner hvor det utføres undersøkelser av EKG-bølgeformer.

3.4 Sammendrag av brukernes behov

Behov	Prioritet	Påvirker	Dagens løsning	Foreslått løsning
-------	-----------	----------	----------------	-------------------

Lett tilgjengelig	Høy	Alle	Ingen eksisterende løsning	Systemet er tilgjengelig som en nettside
Markere deler av visualiseringen	Høy	Alle	I 2D	Click and drag-markering av deler av visualiseringen.
Legge til annoteringer på bølgeformer	Høy	Alle	I 2D	Annoteringer legges til på markerte områder i visualiseringen.
Slette og endre annoteringer	Middels	Alle	Ingen	Liste med annoteringer
Navigere til annotering	Middels	Alle	Ingen	Liste med annoteringer
Lese AI-annoteringer	Middels	Alle	I 2D	AI-annoteringer legges til automatisk
Endre på tidslinjen (Navigere fram og tilbake i tid)	Høy	Alle	I 2D	Horisontal scroll-bar (2D)
Spesifisere et tidsintervall på bølgeformen som skal visualiseres	Høy	Alle	I 2D	Input hvor brukeren spesifiserer tidsintervall
Zoom inn og ut på visualiseringen	Høy	Alle	I 2D	Med feks. Vertikal scrollbar. Kan også bruke touchfunksjoner på

				mobilsjerm eller touchpad.
Rotere på visualiseringen	Høy	Alle	Ingen	Rotere visualiseringen
Filtrere bort kanaler	Høy	Alle	I 2D	Trykke på kanalen og inspisere denne nærmere. Feks. liste over kanaler som visualiseres, hvor man kan velge bort de man ønsker.
Laste opp fil med EKG-bølger	Lav	Alle	I 2D	GUI for opplasting av en slik fil. I første omgang vil en fil bli spesifisert gjennom koden.
Laste opp fil med annoteringer	Lav	Alle	I 2D	GUI for opplasting av en slik fil. I første omgang vil en fil bli spesifisert gjennom koden.
Lagre annoteringer til fil	Høy	Alle	I 2D	Nye annoteringer lagres på filen

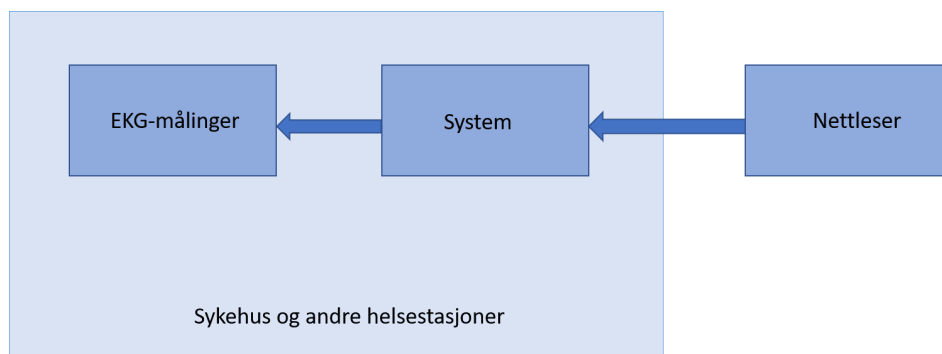
3.5 Alternativer til vårt produkt

Alternativer til vårt produkt er andre systemer som visualiserer EKG-bølger, enten i 3D eller 2D.

4. Produktoversikt

4.1 Produktets rolle i brukermiljøet

Skisse av produktet med hensyn til omgivelsene.



4.2 Forutsetninger og avhengigheter

Alle viktige forutsetninger og avhengigheter, som hvis de endres, vil endre noe som er skrevet i dette dokumentet

Visualiseringen av 3D ekg-bølger forutsetter at brukeren har en fil som inneholder EKG-data. Denne skal være av filtypen .json med headere slikt beskrevet under.

- rec_id
 - o Identifikasjonsnummer på filen
- sample_rate
 - o Nummer som angir hvor mange målinger som er tatt i sekundet
- duration
 - o Angir hvor lang tid målingen er gjort på
- samples
 - o Et objekt som inneholder ulike kanaler. Under hver kanal er det gitt en tabell med alle målingsverdiene.

I tilfellet hvor det brukes datamaskin, forutsettes det at brukeren har mus og tastatur til navigasjon.

5. Produktets funksjonelle egenskaper

Funksjon
Data Format
Data er lagret i JSON format, som inkluderer en header samt en eller flere «sample channels» alt etter hvor mange kanaler EKG opptaket her. Det er typisk 1, 3, 5 eller 12 kanaler i en EKG analyse.
Data kan lastes opp via fil eller via WebSocket. For dette prosjektet er det ikke så viktig hvordan data kommer inn i systemet.

En datafil kan inneholde noen sekunder eller mange minutter med data. Dette varierer veldig.
EKG signaler måles i Millivolt (mV) og har en typisk range på -5 til +5 mV.
Støy kan forekomme som vil gjøre at målte verdier kan være meget større. Visualisering kan i første omgang anta at baseline er rundt 0 mV men det kan forekomme datafiler som ikke er så «rene».
Et annet fenomen som kan forekomme er «wander» som tilsier at baseline verdiene for signalet blir midlertidig høyere eller lavere. Dvs. et signal som normalt måles til -3..+3 mV rundt baseline = 0 mV kan f.eks. plutselig forekomme som +1..+7 mV
Annoteringer (diagnoser eller spesielt interessante segmenter i signalet) vil komme i en separat JSON fil som har format Tidskode-start, lengde, kode
Hvordan data filer lastes inn i systemet er opp til teamet. For integrering inn i Infiniwell's system så er data allerede tilgjengelig.
På lengre sikt så kan data komme inn fortløpende, dvs. en gang i sekundet. Det er utenfor scope i dette prosjektet med mindre teamet har nok tid etter at hovedprosjektet er ferdig.
Navigasjon
Bruker må kunne navigere frem og tilbake i tid. EKG opptak kan variere fra noen sekunder til mange minutter eller timer, så en måte å navigere tidslinjer på er nødvendig. Her man en enkel 2D tidslinje være nok.
Bruker må kunne markere segmenter av EKG signalet som skal passe på skjermen. Dvs. om et EKG opptak er 30 minutter i lengde så må brukeren kunne velge hvor mye av signalet som skal på skjermen av gangen.
Det er fint om det er mulig å navigere til en spesifikk annotering via en slags snarvei.
Det skal være mulig for brukeren å velge/markere et segment av signalet for videre behandling senere. For eksempel, click+drag på bølgeformen for å markere segmentet.
Det skal være mulig for brukeren å legge inn nye annoteringer via GUI (basert på markert segment), enten via drop+down meny eller annen funksjon.
Mulighet til å zoome inn-ut på skjerm grafikken
Mulighet til å zoome inn-ut mht. valgt tidslinje.
Mulighet til å rotere og inspisere signalet fra forskjellige vinkler
Visualisering
Visualisering av EKG bølgeformer skal være i 3D. Dere står fritt til å være kreative her. ☺
Annoteringer skal visualiseres og synkroniseres med de områder i EKG signalet de gjelder for.
Bruk av farger er fint dersom farger har en mening/funksjon. Men husk at mange er fargeblinde, så farge bør ikke være en primær måte å fremheve informasjon.
Når det er sagt, så er det selvfølgelig ønskelig at visualiseringen ser bra, attraktiv og moderne ut.
Mulighet til å skru av/på enkelt kanaler, annoteringer etc.
Systemkrav
Visualisering skal skje i browser-vindu, dvs. full-screen er ikke nødvendig i første omgang.

Teamet står fritt til å velge teknologi rammeverk, men vi ønsker at prosjektet blir utført i et standard/tilgjengelig rammeverk slik at det man integreres inn i vårt system senere.

Vi kan anta at bruker har en relativt moderne PC/laptop

Desktop browser er prioritet som platform. Mobil/tablet er ikke nødvendig i første omgang men kan være interessant som et proof of concept.

6. Ikke-funksjonelle egenskaper og andre krav

Ikke-funksjonelle produkt egenskaper og andre krav til produktet, som krav til standarder etc. Sjekk ut FURPS+ og kvalitetsattributter.

Støtte for nettlesere som Chrome, Edge og Firefox.

Visualiseringen skal følge standardene for elektrokardiografi.

7. Referanser

1. Oppgavebeskrivelse gitt på BlackBoard.
2. Beskrivelser gitt på oppstartsmøte den 11.09.2020.