

mahanc, thomaafl, simontha, miriamhi, ayeshasi, dionk

# Værvarsel for svaksynte

## *SightCast*

Av Maha Nisha Chaudhry, Thomas Aleksander Flier, Simon Tidemann Halvorsen, Miriam Schumacher Hillesund, Ayesha Suman Ishaq, Dion Kumnova

Team nummer: 22

Veiledere: Andrea Arstad Helmersen, Fredrik Kin Shing Ma



IN2000

Software Engineering med prosjektarbeid

Vår 2023

Ved IFI - UiO

# Innholdsfortegnelse

<b>1. Presentasjon</b>	<b>4</b>
1.1 Introduksjon til rapporten	4
1.2 Presentasjon av case	4
1.3 Beskrivelse av team	5
<b>2. Brukerdokumentasjon</b>	<b>7</b>
2.1 Målgruppe	7
2.2 Applikasjonens funksjonalitet	8
2.3 Aksessering av applikasjon	9
<b>3. Brukerundersøkelse</b>	<b>10</b>
3.1 Overordnet metodikk for datainnsamling	10
3.1 Første brukerundersøkelse	10
3.1.1 Metode	10
3.1.2 Presentasjon av kandidatene	11
3.1.3 Presentasjon av funn	11
3.2 Andre brukerundersøkelse	13
3.2.1 Metode	13
3.2.2 Presentasjon av kandidatene	13
3.2.3. Presentasjon av funn	14
3.3 Tredje brukerundersøkelse	15
3.3.1 Metode	15
3.3.3. Presentasjon av funn	16
<b>4. Kravspesifikasjon og modellering</b>	<b>17</b>
4.1 Kravhåndteringsprosessen	17
4.2 Brukerhistorier	18
4.3 Endelige krav	18
4.3.1 Endelige funksjonelle krav	18
4.3.2 Endelige ikke-funksjonelle krav	19
4.4 Endringer og implementering av krav	20
4.5 Modellering	21
4.5.1. Use-case modellering	22
4.5.2. Use-case tekstlig beskrivelse	22
4.5.3 Sekvensdiagram	24
4.5.4 Klassediagram	24
4.6 Universell utforming	26
4.6.1 WCAG	26
4.6.2 Mangler ved WCAG og rollen til brukertesting	27
<b>5. Prosessdokumentasjon</b>	<b>28</b>
5.1 Metode og utviklingsmetodikk	28
5.2 Risikovurdering	30

5.3 Beskrivelse av oppstartsprosessen	31
5.4 Sprinter	31
5.5 Teknisk gjeld	34
<b>6. Produktdokumentasjon</b>	<b>35</b>
6.1 Arkitektur	35
6.1.1 Model-View-ViewModel (MVVM)	35
6.1.2 Front-end- og back-end-utvikling i SightCast	36
6.2 API	37
6.2.1 Valg av API nivå	37
6.2.2 Implementerte APIer	38
6.3 Teknologier og verktøy	39
6.3.1 Kommunikasjon	39
6.3.2 Utvikling	39
6.4 Kvalitetsegenskaper ved appen	40
6.4.1 Brukbarhet	40
6.4.2 Ytelse	40
6.4.3 Sikkerhet	40
6.5 Videreutvikling	41
6.6 Warnings	42
<b>7. Testdokumentasjon</b>	<b>42</b>
7.1. Planlegging og gjennomføring	42
7.2. Enhetstesting	43
7.3. Integrasjonstesting	45
7.4. Testing av universell utforming	46
<b>8. Refleksjon</b>	<b>48</b>
8.1 Samarbeid	48
8.2 Utvikling	49
8.3 Konklusjon	50
<b>9. Vedlegg</b>	<b>50</b>
9.1 Teamavtale	50
9.2 Samtykkeerklæring	51
9.3 Intervjuguide	53
9.4 Prototyper	54
9.5 Fullverdig kravspesifikasjon	57
9.5.1 Brukerhistorier	57
9.5.2 Funksjonelle krav som ble implementert	58
9.5.3 Funksjonelle krav som ikke ble implementert	60
9.5.4 Ikke-funksjonelle krav	61
9.6 Oppfylte kriterier (WCAG)	62
<b>10. Litteraturliste</b>	<b>63</b>

# 1. Presentasjon

## 1.1 Introduksjon til rapporten

I emnet IN2000 - *Software Engineering med prosjektarbeid* ved Universitetet i Oslo våren 2023 har studentene blitt organisert i grupper og fått i oppgave å utvikle en applikasjon basert på et valgt case gitt fra Meteorologisk Institutt. Vi i Team 22 har utviklet en applikasjon ved navn *Sightcast*. I vår utviklingsprosess har vi fokusert på å skape en intuitiv og brukervennlig applikasjon ved å ta hensyn til faktorer som funksjonalitet, tilgjengelighet, brukervennlighet og brukerorientert design. Vårt mål gjennom dette prosjektarbeidet har vært å skape en applikasjon som er spesielt tilrettelagt for vår målgruppe, og en god løsning som kan skape engasjement rundt tilpasning for denne målgruppen.

Denne rapporten gir en omfattende dokumentasjon av hele utviklingsprosessen fra planlegging, implementering, og til testing. Rapporten inneholder også våre refleksjoner og vurderinger av løsningen, inkludert potensielle endringer og utfordringer vi møtte underveis og funksjonalitet vi ville inkludert dersom vi hadde hatt mer tid. Ved å dele våre refleksjoner håper vi å gi en helhetlig forståelse av både styrker og svakheter ved vår løsning og utviklingsprosess.

## 1.2 Presentasjon av case

Vi har valgt case 3 - værvarsle for svaksynte. Ved å velge denne casen vil vi skape en universelt utformet løsning som ikke bare dekker behov svaksynte har i møte med teknologi, men som er spesielt tilpasset denne gruppen. I henhold til Verdens helseorganisasjon (WHO) antas det at det er 284 millioner mennesker i verden som lider av svaksynhet, mens 39 millioner lider av blindhet (Koyuncular, 2021). I Norge alene var det mer enn 320 000 personer som opplevde synshemninger i 2018, ifølge Norges Blindeforbund (NHI, 2023). Det finnes derfor et betydelig antall mennesker som ofte blir utelatt fra den digitale verden.

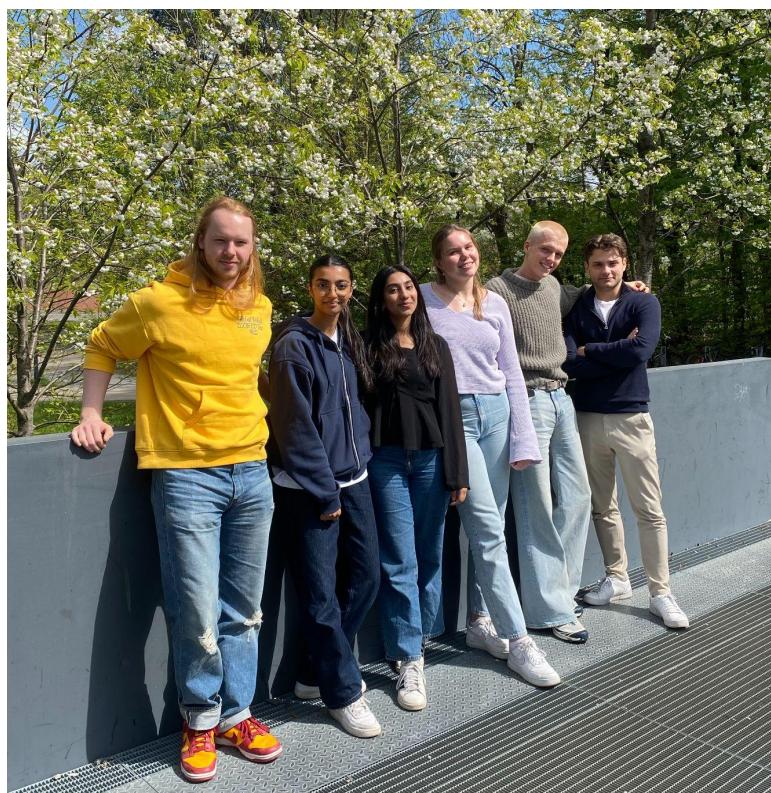
Teknologien har revolusjonert samfunnet vårt, og både offentlige og private tjenester har i stor grad blitt fullstendig digitalisert. Samtidig har det ofte vært forventninger om at alle brukere henger med i denne utviklingen, og samfunnet har muligens glemt eller nedprioritert å vektlegge behovet for at alle skal være inkludert. Det har oppstått et gap mellom de kravene

som er satt for den digitale verden og menneskets evne til å bruke teknologien. Denne avstanden mellom krav fra samfunnet og personers forutsetninger kan illustreres gjennom GAP-modellen, som gir en grafisk fremstilling av funksjonshemminger i møte med teknologi.

Ved å utvikle en applikasjon for svaksynte og blinde, har vi muligheten til å videreutvikle noe som allerede eksisterer slik at det er bedre egnet for denne målgruppen. Ved å sette oss inn i lovbestemt universell utforming og ved å lytte til menneskene som er en del av målgruppen håper vi på å lage en god og brukbar løsning som kan skape gode brukeropplevelser.

### 1.3 Beskrivelse av team

Vi er et team på seks personer fordelt over tre ulike linjer ved instituttet for informatikk.



Fra venstre: Simon, Maha, Ayesha, Miriam, Thomas, Dion

Maha

- Linje: Programmering og systemarkitektur
- Beskrivelse: Jeg er svaksynt selv og har en stor familie, der flere sliter med ulike former for svaksynhet, dermed kan jeg komme med innspill rundt begreper og

opplevelser relatert til svaksynhet. Jeg liker å skrive, og kan derfor bidra med rapporten. Videre har jeg tidligere erfaring med programmering, noe som gjør at jeg også kan bidra med koding.

#### Thomas

- Linje: Programmering og systemarkitektur
- Beskrivelse: Jeg har drevet en del med koding utenom studiene, som sammen med min bakgrunn i prosa gjør at jeg kan bidra mye med på den fronten. Videre har jeg erfaring innen hjemmetjenesten, der svaksynhet er meget normalt hos de eldre. Slik har jeg sett hvordan de opplever og bruker visse applikasjoner, og hvordan vi bør utvikle appen vår basert på dette.

#### Simon

- Linje: Programmering og systemarkitektur
- Beskrivelse: Jeg har vært gruppelærer i to semestre nå, så jeg har opparbeidet meg en god evne til å feilsøke kode. Dette føler jeg er nyttig når vi er flere som jobber på samme kodelinje. Ut over dette er jeg glad i å programmere.

#### Miriam

- Linje: Design, bruk, interaksjon
- Beskrivelse: Jeg har hatt et prosjektfag før (IN1060 - *Bruksorientert design*), der vi lagde en fysisk prototype som skulle dekke behov hos eldre mennesker. Jeg har derfor god erfaring med både prosjektarbeid i grupper, og med å skape et brukerorientert system basert på brukerundersøkelser. Jeg har også jobbet i hjemmesykepleien og har derfor god erfaring med eldre, en gruppe som ofte er svaksynte.

#### Ayesha

- Linje: Programmering og systemarkitektur
- Beskrivelse: Med bakgrunn i programmering og systemarkitektur kan jeg bidra til kodingen ved å bruke tidligere kunnskap fra programmeringsfag. I tillegg gir min personlige erfaring med eldre familiemedlemmer som har nedsatt syn innsikt i de

utfordringene svaksynte ofte møter på ved bruk av teknologi. Dette gir meg muligheten til å bidra til appens formål og kvalitet.

Dion

- Linje: Digital økonomi og ledelse
- Beskrivelse: Med bakgrunn i digital økonomi og ledelse har jeg mye erfaring med skriving av rapporter og prosjektoppgaver. Slik kan jeg komme med gode innspill rundt dette.

## 2. Brukerdokumentasjon

### 2.1 Målgruppe

I caset vi har valgt har vi fått tildelt en tydelig målgruppe vi skal adressere, nemlig svaksynte og blinde. Svaksynhet er definert som nedsatt syn i så stor grad at den krever spesielle hjelpemedier for å følge vanlig undervisning eller utføre vanlig arbeid for seende (Sandvig & Høvding, 2020). Synsstyrken er vanligvis redusert til mellom 6/18 og 6/60 av normalt syn.

Personer som inngår i denne definisjonen av svaksynhet er vår overordnede målgruppe. Men på grunn av mangfoldet med typer synshemninger så vi et behov for å spisse målgruppen videre. Ulike typer synshemmning kan ha ulike behov, og for å sikre at applikasjonen imøtekommer disse behovene måtte vi konkretisere ulike typer synshemninger brukerne kan ha. Vi tok også med i betrakting at personer med synshemninger kan ha ulike nivåer med tekniske forkunnskaper. Vi forsøkte derfor å tilrettelegge for brukere med mindre erfaring innenfor bruk av smarttelefon og nettbrett. Under er en oversikt over de ulike målgruppene vi har tatt høyde for.

#### **Langsynte**

Langsynte kan se klart på fjerne objekter, men møter på utfordringer med objekter på nærmere hold. Dette er en gruppe som ofte har problemer med bruk av mobiltelefoner og tilhørende applikasjoner, siden de brukes på nært hold. (Interoptik, n.d.) I tillegg utgjør denne gruppen en solid andel av befolkningen i Norge. Det estimeres at 13% av 20-25 åringer

er langsynte (NHI, 2023). Tilrettelegging for denne målgruppens behov var av disse grunner et av våre sentrale mål.

### **Fargeblinde**

Fargeblinde er en annen målgruppe som utgjør en relativt stor andel av befolkningen, der 8 prosent av menn er diagnostert (Sandvig & Høvding, 2020). Fargeblindhet fører til at personen har en nedsatt evne til å se farger. Vi har tatt høyde for de vanligste formene for fargeblindhet i utformingen av vår applikasjon. Disse er achromatopsia (gråskala), deutanomali (grønnsvakhet), deutanopia (rød/grønn fargeblindhet), protanomali (rød/grønn fargeblindhet), tritanopi (blå/gul fargeblindhet) og tritanomali (blå/gul fargeblindhet)

### **Blinde**

Blinde er en målgruppe som casen spesifiserer at appen skal ta høyde for. Mennesker som lider av total blindhet vil ha større avhengighet av hørsel, tale og berøring. Men på grunnlag av en begrenset tidsramme og manglende kompetanse, ble fokuset på funksjonalitet tiltenkt blinde delvis manglende. Vi ønsker imidlertid å understreke at vi er bevisste på betydningen av tilgjengelighet og inkludering, og at funksjoner ment for å dekke behovene til blinde er noe vi ville inkludert i senere iterasjoner av appen.

## **2.2 Applikasjonens funksjonalitet**

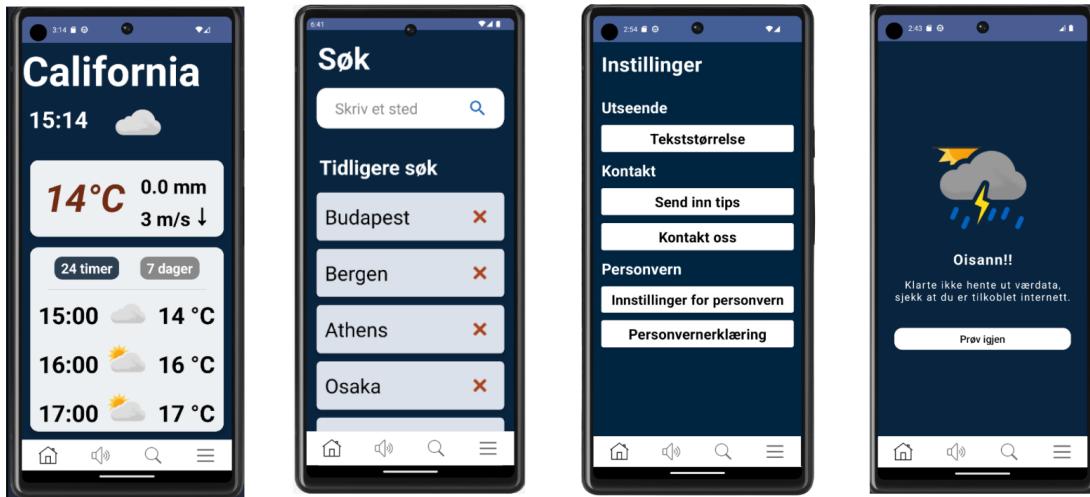
Applikasjonen er tredelt; vi har en hjemskjerm, en skjerm for søking, og en skjerm for innstillinger. Nederst på skjermen finnes det en meny der man kan velge mellom de ulike skjermene, samt text-to-speech funksjonalitet som leser opp det nåværende været i form av antall grader, antall mm nedbør, vindstyrke og vindretning.

I hjemskjermen får brukeren informasjon om det gjeldende været ved antall grader, antall mm nedbør, vindstyrke og vindretning, samt et vær-symbol som viser om det er sol, skyet, regn osv. I tillegg kan brukeren få oversikt over været de kommende 24 timene og de syv neste dagene. Ved oppstart av appen eller ved trykk på hjem-knappen, vil appen vise været fra brukerens nåværende posisjon.

I skjermen for søking kan brukeren søke etter værprognosenter et valgt sted. Hvert søk vil lagres, slik at brukeren enkelt kan finne frem til gamle søk i stedet for å søke opp på nytt. Tidligere søk kan også slettes ved å trykke på kryss-symbolet.

Appen inneholder en innstillinger-skjerm der det er planlagt funksjonalitet for endring av tekststørrelse, og informasjon om personvern og kontakt.

Dersom brukeren starter appen uten at data er lastet ned, eller forsøker å navigere til hjemskjermen når den ikke inneholder værdata, vil det komme en skjerm for feilmeldinger.



Figur 1: Skjermbilder av appen

## 2.3 Aksessering av applikasjon

Applikasjonen er kun tilgjengelig på Android plattformen. Videre krever den minst API nivå 30. Applikasjonen vår ligger tilgjengelig i gruppens git repository "Gruppe 22". Første gang man kjører applikasjonen må man godkjenne bruk av posisjon for at applikasjonen skal ha full funksjonalitet. Man kan likevel søke opp andre steder, men man vil ikke få værdata for sin posisjon. Man trenger også tilgang til internett for at applikasjonen skal fungere.

### 3. Brukerundersøkelser

#### 3.1 Overordnet metodikk for datainnsamling

En av våre viktigste kilder til data var intervjuene som ble gjennomført gjennom prosjektets livssyklus. Disse sikret at appens funksjonalitet var rettet mot målgruppens behov.

Intervjuene var spesielt nyttige i å kartlegge og evaluere krav ved å samle inn kvalitativ data knyttet til applikasjonens funksjonalitet.

I forkant av alle brukerundersøkelsene signerte deltakerne et samtykkeskjema som ga deltakeren informasjon om formålet med brukerundersøkelsen, hvordan personopplysninger ble behandlet og når dataen som er samlet inn blir slettet. Hver deltaker ble intervjuet individuelt, og ble bedt om å fjerne alle hjelpeemidler under intervjuet (briller, linser osv.). Vi så for oss at løsningen kunne være nyttig i brukskontekster der brukere ikke har tilgang til hjelpeemidler de vanligvis benytter.

#### 3.2 Første brukerundersøkelse

I den første brukerundersøkelsen ønsket vi å evaluere tre lavoppløselige prototyper vi hadde laget for å undersøke om disse kunne fylle behov hos svaksynte. Vi gjennomførte brukerundersøkelsen på familiemedlemmer til et av medlemmene i gruppen, da flere av disse var svaksynte. Ved å inkludere kandidater som hadde et nært forhold til den som utførte intervjuet, risikerte vi bias i resultatene. For å dette anonymiserte vi de ulike prototypene slik at deltakerne ikke kunne knytte dem til noen.

##### 3.2.1 Metode

Vi utførte et semistrukturert intervju der vi hadde skrevet ned noen spørsmål på forhånd slik at intervjuet fikk et rammeverk. Ellers fikk deltakerne mulighet til å styre intervjuet selv, og kunne derfor gi mer frie og utdypende svar. Slik skaffet vi oss detaljert data på kort tid.

Først ble deltakerne presentert med kjente vær-applikasjoner, som iPhone sin vær-app og Yr. Ved å teste disse appene først ville vi få en indikator på hvilke deltakere vi skulle introdusere prototypene våres til. De som klarte å lese iPhone og Yr sin værmelding uten problemer tok vi ikke med videre i brukerundersøkelsen. De som ikke klarte å lese disse værmeldingene,

slet med å lese værmeldingen, eller holdt mobilen nærmere eller lengre unna enn “normalt”, tok vi med videre i brukerundersøkelsen.

Vi hadde to deltakere med videre i brukerundersøkelsen som begge var langsynte. Deltakerne ble først spurta om å oppgi sin alder og synsstyrke, hvordan teknologi er brukt i deres hverdag og til slutt hvordan dere synshemning har påvirkning på dette. Deretter ble de presentert med de tre prototypene, og stilt spørsmål rundt hva de synes om designet; hva de mener mangler eller kan forbedres med prototypene, og om de har ideer til hvordan en ideell vær-app burde se ut.

### 3.2.2 Presentasjon av kandidatene

I undersøkelsen mener vi det var viktig og nyttig å ta hensyn til synsstyrke og bruk av hjelpemedidler, i tillegg til deltakernes erfaringer og teknologibruk. Dette ga oss en bedre forståelse av svarene og muliggjorde en mer definert målgruppe, samtidig som vi kunne forutse mulige løsninger.

Kandidat 1 var i en alder av 53 og hadde en synsstyrke på +1.5 og +2, men brukte sjeldent briller. Kandidaten bruker kun teknologi når det er nødvendig, og mener selv at dette skyldes mangel på tilpasset teknologi som imøtekommers hans behov. Kandidaten sa at han foretrekker telefonsamtaler fremfor å sende meldinger, ettersom det å bruke tastaturet på telefonen kan være utfordrende, blant annet pga synshemning.

Kandidat 2 var i en alder av 50 år og var usikker på synsstyrken sin og bruker ikke briller til vanlig. Kandidaten har støtt på problemer med teknologi, og må ofte zoome inn der det er mulig, eller ta et skjermbilde for å zoome inn på det.

### 3.2.3 Presentasjon av funn

Prototype 1 (Se vedlegg: *Prototype I*):

Deltakerne kunne holde telefonen på en “normal” avstand og lese værinformasjonen enkelt. Alle symboler valgt var synlige. Til tross for dette var deltakerne enige om at designet var veldig abstrakt, og de ville sannsynligvis ikke ha brukt appen på grunn av utseendet.

Fargekombinasjonen ga heller ikke inntrykk av en vær-app. Kandidat 1 mente spesielt at bakgrunnsfargen var ubehagelig og forstyrrende for øynene.

Prototype 2 (Se vedlegg: *Prototype 2*):

Da deltakerne ble representert med den andre prototypen, var de enige om at en analog klokke ville gjøre appen mer kompleks enn nødvendig, og at en enkel og oversiktlig løsning var å foretrekke. Kandidat 1 mente imidlertid at det ville være nyttig å inkludere en klokke i appen som angir nåværende tidspunkt. Til vanlig må vedkommende låse skjermen for å lese klokken, ettersom klokkeslettet i toppfeltet på telefonen er for lite til å leses. Deltakerne hadde ingen innvendinger mot ideen om animasjon, så lenge det samsvarer med designet og ikke blir distraherende. De mente likevel at animasjonen i prototypen var abstrakt og så barnslig ut.

Prototype 3 (Se vedlegg: *Prototype 3*)

Deltakerne ga tilbakemelding på den tredje prototypen og var enige om at designet var bra på grunn av sin minimalistiske stil og høy kontrast. Dette gjorde all informasjon lettleselig både på hovedskjermen og søker skjermen. Kandidat 1 uttrykte at funksjonen for lagrede søker ville være hjelpsom da man kunne slippe å bruke tastaturet som ofte er for lite til å skrive inn samme søker gjentatte ganger. Imidlertid var begge kandidatene enige om at fargevalget og kombinasjonene av farger virket barnslig.

Etter å ha gått gjennom tilbakemeldingene fra brukerundersøkelsen i plenum, innså vi at vår største utfordring ville være å skape en vær-app som både var attraktiv og appellerte til brukerne. Basert på tilbakemeldingene vi mottok fra deltakerne var det tydelig at fargevalg og fargekombinasjoner var en avgjørende faktor for om appen ville bli brukbar. Vår prioritet ble derfor å jobbe med å finne en fargepalett som ville tiltrekke seg brukerne, samtidig som vi ville opprettholde enkelhet og brukervennlighet i appen. Vi tok med oss deltakernes ønsker og tilbakemeldinger om funksjoner som kunne forbedre brukeropplevelsen videre, som for eksempel en lagring av søker og klokke.

### 3.3 Andre brukerundersøkelse

Etter å ha tatt deltakernes tilbakemeldinger i betrakting fra den første brukerundersøkelsen, designet vi en ny prototype som vi utviklet videre i Android Studio (Se vedlegg: *Prototype 4*). Nå som vi hadde en mer høyoppløselig prototype ønsket vi å ha en ny brukerundersøkelse der brukerne kunne interagere med appen ved at vi hadde en Samsung telefon koblet til emulatoren.

#### 3.3.1 Metode

I denne brukerundersøkelsen ble et semistrukturert intervju gjennomført. I forkant av hovedintervjuet utførte vi et pilotintervju for å sikre en best mulig undersøkelse. Vi utarbeidet en intervjuguide som inkluderte en liste over spørsmål som var ønsket å stille, samt en beskrivelse av hvordan vi skulle gjennomføre intervjuet (Se vedlegg: *Pilotundersøkelse*). Pilotintervjuet ble utført på et av medlemmene i gruppen.

Etter pilotundersøkelsen analyserte vi resultatene og identifiserte feil og mangler i intervjuguiden. Vi innså at spørsmålet om brukervennlighet var for åpent og lite konkret, og det ville derfor være vanskelig å sammenlikne svarene. Basert på denne informasjonen justerte vi spørsmålene, slik at våre brukerundersøkelser var effektive og produserte pålitelige resultater (Se vedlegg: *Andre brukerundersøkelse*).

#### 3.3.2 Presentasjon av kandidatene

I den andre brukerundersøkelsen inkluderte vi kandidat 1 fra første runde i tillegg til tre nye deltakere. Blant de nye deltakerne var det tre som var nærsynte. I tillegg var en av deltakerne blind på venstre øye.

Kandidat 2 var 24 år og hadde en synsstyrke på -8 på det høyre øyet og -5,75 på det venstre øyet. Kandidaten forteller at teknologi spiller en stor rolle i deres liv, og at teknologi brukes både i privatlivet og i jobbsammenheng for å dokumentere ting som har med pasienter å gjøre. Kandidaten har aldri opplevd utfordringer med teknologi som svaksynt.

Kandidat 3 var 23 år og hadde en synsstyrke på -6 på begge øynene. Vedkommende bruker teknologi i både privatlivet og jobbsammenheng. Kandidaten jobber som yrkessjåfør og

bruker teknologi i arbeidslivet, som for eksempel skannere for pakker og GPS-systemer. Til tross for synshemningen, har personen ikke opplevd noen utfordringer med teknologi.

Kandidat 4 var 41 år og hadde en synsstyrke på -7,5 på det høyre øyet og var blind på det venstre øyet. Kandidaten forteller at teknologi forenkler hverdagen, spesielt med bruken av nettbanker, digital netthandel og Helsenorge. Imidlertid har kandidaten støtt på utfordringer knyttet til teknologi med dårlig oppløsning.

### 3.3.3. Presentasjon av funn

Kandidat 1 påpekte at fargekombinasjonen av gult og rødt ikke var optimal, da det gjorde det vanskelig å lese klokken på skjermen. Kandidaten kunne imidlertid lese det som var plassert på den hvite bakrunnen uten problemer. Kandidaten foreslo å bruke en mørk bakgrunn med lyse detaljer eller en lys bakgrunn med mørke detaljer. Kandidaten viste til TV2 sin vær-app som et eksempel på vellykket fargebruk.

Videre forklarte Kandidat 1 at han hadde ingen problemer med å få med seg nåværende værinformasjon, men at han måtte få forklart 5-dagers menyen. For kandidaten var det viktigste at appen var tiltalende og samtidig lett å bruke for å få værinformasjon. I tillegg mente vedkommende at andre vær-apper ofte har overveldende mye informasjon om været, som kan føre til forvirring. Appen vår har imidlertid tilstrekkelig informasjon om været. Siden hovedpunktet av appen var på forsiden, var det ikke så mye navigasjon som skulle til, noe deltakeren likte. Deltakeren mente at appen ikke trengte flere funksjoner, da det ville gjøre den overkomplisert.

Kandidat 2 mente at en mørk-modus kunne være nyttig for personer med nedsatt syn, da de ofte har sensitive øyne. Dette kan bidra til å redusere øyeplager og utmattelse i øynene når de bruker appen.

Kandidat 3 hadde problemer med å lese informasjonen på telefonen og måtte holde den veldig nær ansiktet sitt for å se det. Dette førte til øyetretthet og ubehag, så testingen ble avsluttet. Det kan være nyttig å undersøke om større tekststørrelse eller endringer i kontrast kan hjelpe personer som opplever lignende problemer når de bruker appen, eller funksjoner

som text-to-speech. Kandidatene mente at navigasjonsstrukturen var naturlig og enkel, og at det minnet dem om andre apper de hadde brukt. Alle deltakerne mente at designet var på god vei til å være en app som de kunne forestille seg å bruke.

Kandidat 4 hadde ingen problemer med fargekombinasjonene, men påpekte at værsymbolet kunne ha hatt en annen farge. Personen mente også at det var vanskelig å skille ut menyen nederst på skjermen, og at dette var noe som burde tydeliggjøres i appen. Kandidaten holdt telefonen tett opptil det høyre øyet. Vi diskuterte mulige løsninger for hvordan personer som kandidat nummer fire kan bruke appen vår uten hjelpemidler. En mulig løsning var å forstørre teksten under baren og integrere den bedre i designet, kanskje ved å endre farge. Dette ville gjøre det lettere å se text-to-speech-knappen, som kan brukes til å få værinformasjon.

### 3.4 Tredje brukerundersøkelse

Ved slutten av prosjektet ønsket vi å ha en brukerundersøkelse av den ferdigstilte appen slik at vi fikk evaluert brukervennligheten samt det ferdige designet av appen. I denne omgangen benyttet vi oss av 2 Android telefoner: Samsung Galaxy S21+(6,7") og Samsung Galaxy S10e(5,8").

#### 3.4.1 Metode

Et semistrukturert intervju var utført med kandidat 1 fra de andre brukerundersøkelsene. Vi benyttet concurrent probing og retrospective probing som metoder for å kartlegge appens brukervennlighet (*Running a Usability Test*, n.d.). Brukeren ble bedt om å utføre følgende oppgaver:

1. Søking av valgfritt sted
2. Navigering til nåværende posisjon
3. Opplesing av værdata

Underveis ble kandidaten bedt om å verbalisere tankene sine for å få innsikt i hvordan de interagerte med appen. For å måle brukervennligheten observerte vi metrikker som hvor lang tid det tok å lære systemet, antall brukerfeil som oppstår, og responstid fra systemet. Etter at kandidaten hadde utført oppgavene, stilte vi spesifikke spørsmål for å få mer detaljert informasjon om brukerens opplevelse og eventuelle problemer som oppstod:

1. Hva kunne ha gjort din opplevelse med å bruke appen bedre? Hvilken funksjonalitet savner du i appen?
2. Ville du anbefale denne appen til andre som er visuelt utfordret? Hvorfor eller hvorfor ikke?

### 3.4.3 Presentasjon av funn

Ved søking på valgfritt sted likte brukeren søkefunksjonen fordi det gjorde det enklere å søke etter spesifikke steder ved hjelp av søkerfeltet. Kandidaten satte pris på at de fikk opp en liste med steder basert på deres søk. Dette eliminerte behovet for å skrive inn hele stedsnavnet.

Ved navigering til nåværende posisjon hadde kandidaten problemer med å returnere til sin nåværende posisjon. Kandidaten trodde man måtte søke opp posisjonen på nytt for å komme seg tilbake. I ettertid ble brukeren fortalt at det var mulig å trykke på “hjem”-ikonet for å navigere seg til sin nåværende posisjon. Vedkommende presiserte at dette var noe han ikke hadde vært borti før, men at det krevde mindre interaksjon med appen, noe som var positivt.

Ved opplesing av værdata, var kandidaten fornøyd med at teksten i appen kunne bli opplest. Allikevel var det misnøye rundt tiden det tok for text-to-speech å starte. Kandidaten mente likevel at dette var en nyttig funksjon, ettersom den gjorde appen tilgjengelig, selv om det var rom for bedring.

Kandidaten mente at appen hadde det viktigste; stor tekststørrelse, god fargekontrast og bra fargevalg. Kandidaten mente likevel at det ville være gunstig å ha full funksjonalitet i innstillinger-skjermen, blant annet endring av tekststørrelse. Videre likte han muligheten til å bytte mellom timevarsel og 7-dagers varslel på en enkel måte ved at det var tilgjengelig på samme sted. Brukeren sa han ville anbefalt appen til andre svaksynte fordi den inneholder tilstrekkelig værinformasjon uten forstyrrelser slik at den er lesbar og enkel å bruke.

## 4. Kravspesifikasjon og modellering

### 4.1 Kravhåndteringsprosessen

Under prosjektet tok vi i bruk en kravhåndteringsprosess på tre steg. Denne stod sentralt under utarbeidelsen av våre opprinnelige og endelige krav. Formålet med prosessen var å systematisk forsikre oss om at kravspesifikasjonen stemte overens med målgruppens behov. Å tydeliggjøre retningslinjer på hvordan vi håndterer krav bidro også til å gi hele teamet en bedre forståelse for kravene og deres betydning for prosjektet.

#### **Steg 1: Forstudie/målanalyse**

Det første steget består av å tydeliggjøre hvem målgruppen er, og deretter utarbeide noen grunnleggende tanker rundt deres behov. Vi skaffet oss innsikt gjennom dokumentasjon på nett, intervjuer og egne erfaringer teammedlemmer hadde med svaksynhet. Vi utformet brukerhistorier (Se vedlegg: *Brukerhistorier*), og basert på dette identifiserte vi noen fundamentale funksjoner og designelementer som vi videre kunne ta utgangspunkt i.

#### **Steg 2: Kravinnsamling og kravanalyse**

Steg to involverer innsamling og prioritering av første runde med krav. Her legges fundamentet for kravspesifikasjonen. Vi identifiserte kravene gjennom intervjuer, WCAG-krav og gransking av lignende apper. Vi benyttet også numerisk prioritering fra 1 til 3, for å rangere kravene etter deres viktighet. Utgangspunkt for tidsestimater og testingsprosesser av kravene kartlegges også i dette steget.

#### **Steg 3: Validering og endring**

Steg tre dreier som å finne eventuelle avvik i kravspesifikasjonen, slik at de kan fjernes eller justeres. Dette er en kontinuerlig evaluatingsprosess som i hovedsak skjedde gjennom intervjuene, hvor vi spurte målgruppen om deres mening knyttet til funksjonelle krav vi hadde utarbeidet. I tillegg evaluerte vi kravene i forhold til teamets kompetanse og tilgjengelighet. Steg tre var dermed en fleksibel og iterativ prosess som sikret relevante krav for prosjektet.

## 4.2 Endelige krav

Under er en oversikt over de viktigste endelige kravene våre. I vedlegget vil du finne de resterende kravene, samt de som ble fjernet. Kravene ble utformet gjennom kravhåndteringsprosessen.

### 4.2.1 Endelige funksjonelle krav

Under er en oversikt over de viktigste funksjonelle kravene våre, disse ble tildelt prioritet en, basert på en skala på fra en til tre. For fullstendig oversikt, se vedlegg: *Funksjonelle krav som ble implementert*.

- Systemet skal på forsiden oppgi temperaturen til brukerens nåværende posisjon.
- Appen skal tilby et talestyringsalternativ som leser opp temperatur og nedbør
- Appen skal inneholde en hjem knapp som får appen til å vise været til brukerens nåværende posisjon
- Appen skal fremstille værutviklingen i tabeller
- Appen skal gi brukeren muligheten til å bytte mellom 24-timer værvarsel og 7-dagers værvarsel.
- Appen skal ikke kræsje ved søker etter steder som ikke har værdata.

### 4.2.2 Endelige ikke-funksjonelle krav

De ikke-funksjonelle kravene er delt opp i tre kategorier; produktkrav, organisatoriske krav og eksterne krav. For fullstendig oversikt, se vedlegg: *Ikke-funksjonelle krav*. Disse følger det samme prioriteringssystemet som de funksjonelle kravene.

#### 1. Produktkrav:

- Appen skal struktureres etter MVVM
- Appen skal kunne kjøre på API-level 30 til 33
- Appen skal hente værdata fra Location Forecast API
- Appen skal hente data asynkront
- Appen skal følge WCAG kravet om konsekvent navigering(Kriterium 3.2.3 )
- Appen må være tilpasset flere skjermstørrelser og enheter
- Appen skal være utformet i henhold til WCAG krav om høykontrast
- Appen skal ikke kreve unødvendige eller overflødige brukerinteraksjoner for å vise værdata.

## 2. Organisatoriske krav:

- Teamet skal sørge for at relevant dokumentasjon er tilgjengelig og oppdatert

## 3. Eksterne krav:

- Appen skal følge 50 av 78 WCAG-retningslinjer
- Appen skal ikke krenke brukerens personvern eller samle inn data uten deres samtykke.
- Appen skal være i samsvar med personvernloven

## 4.3 Endringer og implementering av krav

Gjennom bruk av kravhåndteringsprosessen identifiserte vi ved flere anledninger et behov for å endre de funksjonelle kravene. Dette ble gjort for å bedre tilpassa applikasjonen til målgruppens behov, eller på grunn av mangel på tid og kompetanse. I denne delen skal vi se nærmere på hvorfor noen av kravene ikke ble implementert. Disse kravene ble opprinnelig antatt til å være sentrale funksjonaliteter, og deres fjerning var av stor relevans til prosjektet. Det var kun funksjonelle krav som gjennomgikk endringer eller ble fjernet. For oversikt over krav som ikke ble implementert, se vedlegg: *Funksjonelle krav som ikke ble implementert*.

### Innzooming

En av funksjonene vi lenge hadde planer om å implementere var muligheten til å kunne zoome inn på elementer i appen. Dette endte vi opp med å ikke implementere, grunnet manglende interesse fra målgruppen for en slik funksjon. Dette ble kartlagt i etterkant av en dialog med deltakerne av første brukerundersøkelse. Funksjonen ble oppfattet som kronglete og lite praktisk, spesielt hvis man sjekket appen i farta, noe som er svært vanlig for mobile applikasjoner.

### Analog klokke

I tillegg til å tilby viktig funksjonalitet, ønsket vi å gjøre appen vår kreativ og nyskapende. Etter å ha diskutert ulike alternativ, bestemte vi oss for å implementere en analog klokke. Tanken var at brukere skulle kunne velge klokkeslett de ville se været for, ved å dra rundt på klokken. Imidlertid, etter å ha mottatt tilbakemeldinger fra den første brukerundersøkelsen, innså vi at denne funksjonen ikke ville være lønnsom. Vi fikk et større bilde av hvilke mennesker som ville bruke appen vår, og lagde scenarier basert på dette. (Se målgruppe). Vi

så for oss at eldre mennesker var blant disse. Eldre mennesker sliter ikke bare med svaksynhet, men har også ofte skjelvende hender. Vi innså at dersom vi inkluderte denne funksjonen, ville en stor del av mennesker som ofte er svaksynte ikke kunne ta i bruk appen vår, på grunn av kompleks design.

## **Animasjon**

Ved gransking av andre vær-apper så vi at animasjon var et repeterende designvalg. Vi innså at implementasjon av dette ville være en måte imøtekommne brukerhistorie 10 og brukerhistorie 14. I tillegg var dette også noe som ville oppfylle WCAG kravet om familiaritet. Etter å ha representert kandidater fra målgruppen med et designutkast basert på dette, innså vi at det egentlig gikk i strid med brukerhistorie 14. På grunn av at vi ikke hadde klart å optimalisere det i henhold til de standardene kandidatene var vant med fra tidligere vær-apper, var dette noe vi måtte re-vurdere i henhold til kunnskapen og tiden vi hadde. Vi konkluderte også med at designvalget kunne gi motsatt virkning, og faktisk være distraherende i stedet for å gjøre det enklere å oppfatte værfordoldene. Vi evaluerte ulike tilnærninger og kom frem til at det var andre passende metoder å imøtekommme kravspesifikasjonen på, og valgte derfor å ikke inkludere animasjon i appen vår.

## **Endring av skriftstørrelse**

Dette var en funksjon vi egentlig så et ganske stort behov for. Planen for å implementere en skriftstørrelse “slider” i innstillinger-menyen. Denne skulle gi brukeren av applikasjonen muligheten til å justere på skriftstørrelsen rundt appens mest sentrale elementer, som blant annet temperaturen på forsiden, bottom bar og fremtidsvarslet på forsiden. I undersøkelse 2 fikk vi tillegg til tilbakemelding om at funksjonen hadde vært velkommen, selv om deltakerne i utgangspunktet mente at tekststørrelsen vi hadde satt fungerte fint. WCAG retningslinje 1.4.4 understreker også nyttigheten av å kunne justere teksten.

Grunnen til at vi ikke implementerte denne funksjonalitet var at vi undervurderte tiden det ville ta å implementere denne funksjonaliteten samtidig som det dukket opp viktigere prioriteringer i applikasjonen, noe som førte til at dette kravet ble sendt nedover i backlog prioriteringen.

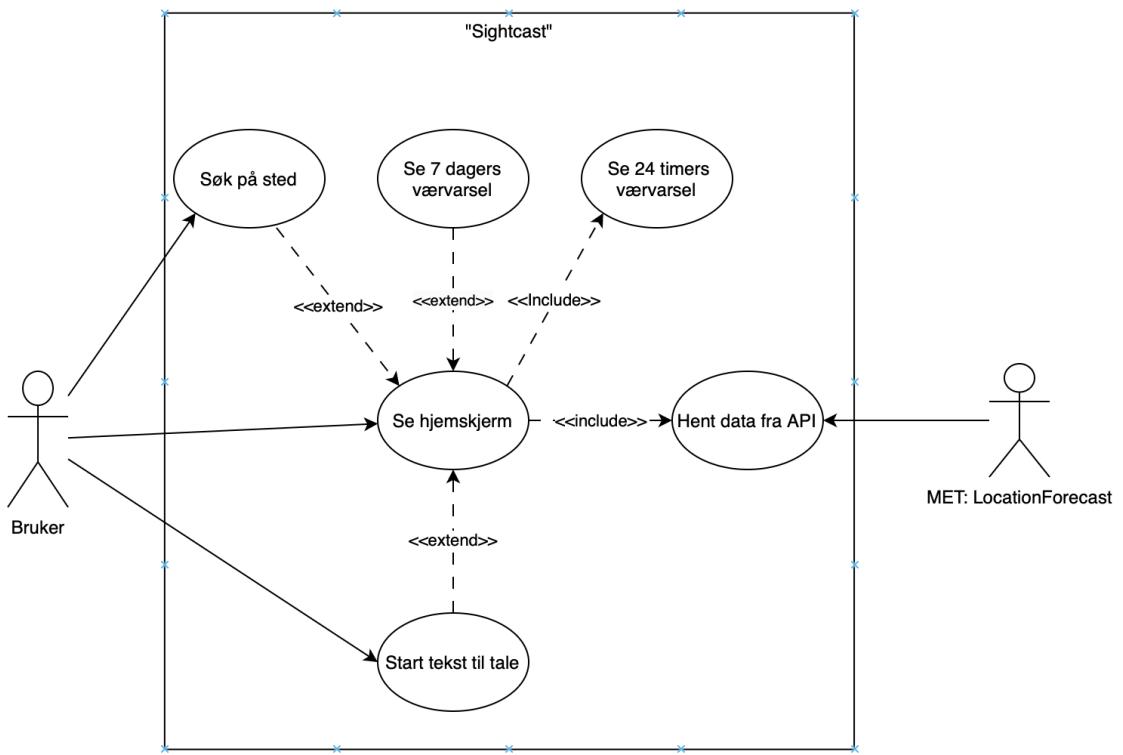
## 4.4 Modellering

Under prosjektet tok vi til bruk tre diagrammer, disse var use-case, klasse- og sekvensdiagram. Hovedpoenget med modelleringen var å skape en felles forståelse for de aller viktigste kravene i applikasjonen.

### 4.4.1 Use-case modellering

I use-case diagrammet har vi modellert tre use-case:

1. Se værdata ved nåværende posisjon
2. Søk på lokasjon for å se værdata
3. Les opp værdata gjennom text to speech



Figur 2: Use case diagram

### 4.4.2 Use-case tekstlig beskrivelse

#### Use case 1 - Se værdata ved nåværende posisjon

Primæraktør: Bruker av app

Sekundæraktør: *LocationForecast* (API fra MET)

Prebetingelse: Brukeren har brukt appen tidligere

Postbetingelse: Værinformasjon knyttet til oppgitt posisjon vises

Hovedflyt:

1. Bruker trykker på hjemskjerm knapp
2. Appen henter ut brukerens posisjon
3. Appen sender forespørsel til sekundær aktør om å hente værdata på posisjonen
4. Appen oppdaterer hjemskjermen med værdata

Alternativ flyt:

2.1. Systemet får ikke hentet ut posisjon

2.2 Systemer viser gammel data hvis det eksisterer, hvis ikke viser den feilmelding skjerm

3.2. Appen får ikke respons fra sekundæraktør

3.2 Systemer viser gammel data hvis det eksisterer, hvis ikke viser den feilmelding skjerm

## **Use case 2 - Søk på lokasjon for å se værdata**

Primæraktør: Bruker av app

Sekundæraktør: *LocationForecast* (API fra MET)

Prebetingelse: Brukeren har brukt appen tidligere

Postbetingelse: Værinformasjon knyttet til oppgitt posisjon vises

Hovedflyt:

1. Bruker nавигerer til søkeskjerm.
2. Bruker søker etter ønsket sted.
3. Appen returnerer en liste med steder som samsvarer med brukerens input.
4. Bruker trykker på et alternativ.
5. Appen sender forespørsel til sekundæraktør om å hente værdata på det oppgitte stedet
6. Brukeren sendes til en oppdatert hjemskjerm, som nå viser været ved det oppgitte stedet.

Alternativ flyt:

- 4.1 Lokasjonsdata finnes ikke
- 4.2 Systemer viser gammel data hvis det eksisterer, hvis ikke viser den feilmelding skjerm

### **Use case 3 - Les opp været i nåværende posisjon gjennom text-to-speech**

Primæraktør: Bruker av app

Sekundæraktør: Ingen

Prebetingelse: Hjem skjermen har værdata

Postbetingelse: Værdata på hjemskjermen leses opp

Hovedflyt:

1. Bruker trykker på tekst til tale
2. Appen leser opp værdataen på hjemskjermen

Alternativ flyt:

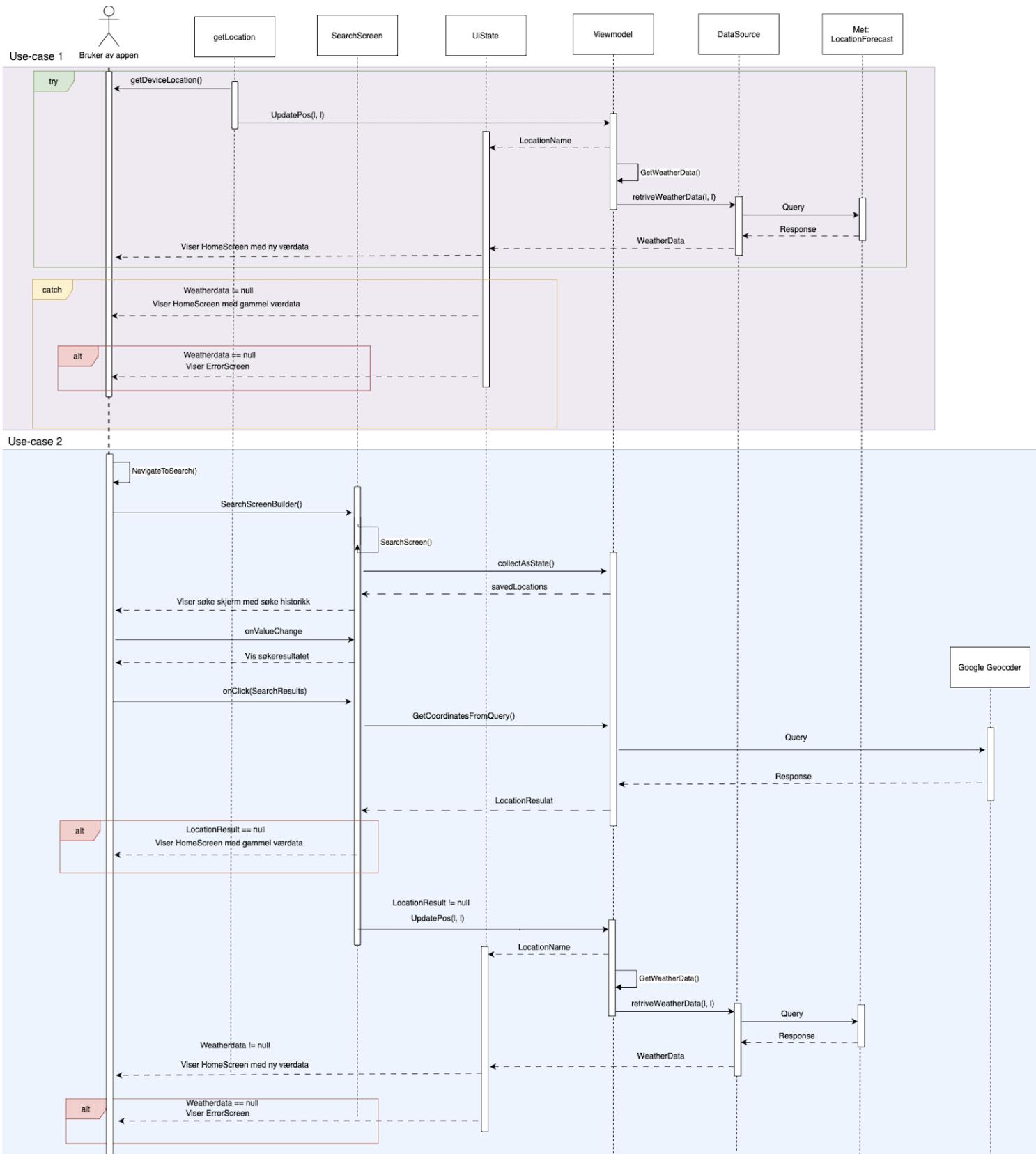
- 2.1 Google sin text-to-speech engine er treg og klarer ikke starte opp
- 2.2 Brukeren forsøker å trykke flere ganger
- 2.3 Dette skaper kø i loggen

#### **4.4.3 Sekvensdiagram**

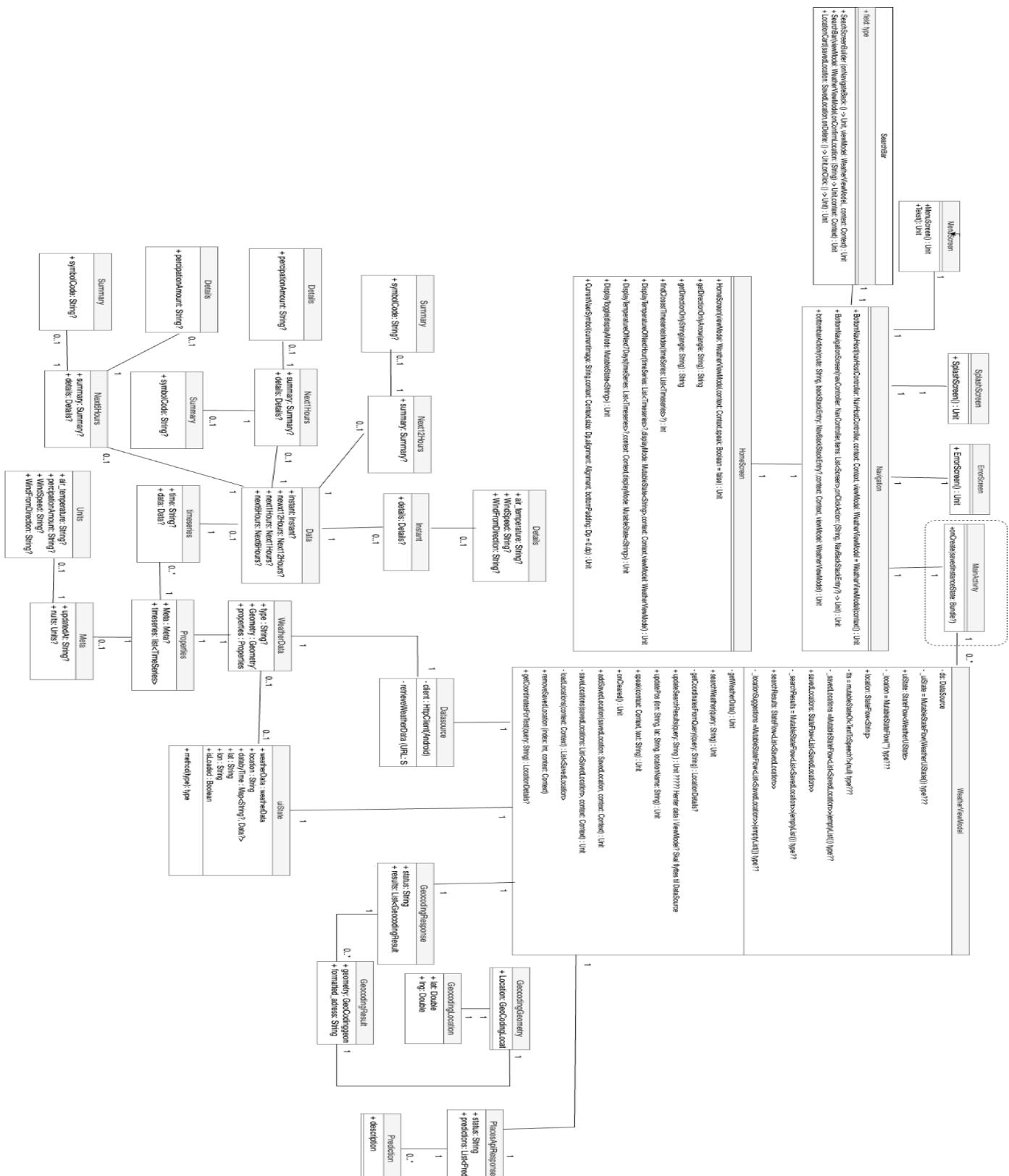
Sekvensdiagrammet inneholder use-case 1 og use-case 2 (Se Figur 3). Under utformingen av diagrammet valgte vi å illustrere en try-catch, ettersom det er en ganske unik flyt i programmet, som ville vært utfordrende å visualisere med alternative flyter.

#### **4.4.4 Klassediagram**

Klassediagrammet illustrerer hvordan systemet er bygd opp og relasjonene mellom de ulike komponentene for å kartlegge objektene implementert (Vatne, 2022, s14). Man ser også multiplisiteten, som illustrerer hvor mange referanser hvert objekt har (Se Figur 4).



Figur 3: Sekvensdiagram



Figur 4: Klassediagramm

## 4.5 Universell utforming

Universell utforming spilte en aktiv rolle i utarbeidelsen av kravspesifikasjonen vår. Det er en tilnærming til design som skal tilgjengeliggjøre produkter, tjenester og programmer for alle mennesker, uavhengig av deres funksjonsnivå og evner (uutilsynet, n.d.). Et design basert på prinsipper innenfor universell utforming vil derfor bidra til å tilrettelegge appen for svaksynte.

### 4.5.1 WCAG

Mange av de funksjonelle kravene våre har sitt opphav i WCAG krav, mens de ikke-funksjonelle kravene ofte består av konkrete WCAG krav. Under er kriteriene som på direkte vis påvirket utformingen av applikasjonen. Vi følger til sammen 58 av 78 WCAG krav, men alle er ikke av like stor relevans. Se vedlegg: *Oppfylte kriterier i WCAG* for oversikt over WCAG kravene vi følger.

#### **Retningslinje 1.4. Mulig å skille fra hverandre**

Denne retningslinjen omhandler tilgjengelighet av tekstinnhold og inneholder to suksesskriterium av spesiell relevans til appen:

- **Kriterium 1.4.6 (Kontrast (Avansert))** - Mobiltelefoner har en høyere sannsynlighet for å bli brukt i forstyrrende omgivelser preget av elementer som sollys eller andre eksterne lyskilder. Scenarioer som dette tilrettelegger for et større fokus på kontraster i appen. Lesbarhet er av øverste prioritering for oss, og vi valgte derfor det mer avanserte 1.4.6 kriteriet som etterspør en kontrast på 7:1, fremfor 1.4.5 kriteriet som etterspør 5:1. Dette er implementert rundt de mest sentrale elementene i appen, blant annet temperatur og klokkeslett på hovedskjermen.

#### **Retningslinje 3.2. Forutsigbar**

Denne retningslinjen handler om at appens elementer skal presenteres på en konsekvent og forutsigbar måte. De viktigste suksesskriteriene:

- **Kriterium 3.2.1(Fokus)** - Kriteriet om fokus handler om å unngå store endringer når en ny komponent brukeren kan interagere med kommer i fokus. Et eksempel på hvordan vi har tatt hensyn til disse kriteriene er i vår implementasjon av tabeller for å representer værutviklingen over de neste 24 timene og de neste 7 dagene. Når

brukeren trykker på et av alternativene, bevares den samme formateringen, og det er kun værdata og den forklarende teksten som endres. Ved å opprettholde det samme formatet i tabellene, unngår vi store og uforutsette endringer når brukeren navigerer mellom tabellene.

- **Kriterium 3.2.3 (Konsekvent navigering)** - Navigasjonslenker som befinner seg på flere sider bør plasseres likt og dermed på forutsigbart vis. Dette skal tilrettelegge for intuitiv navigering som igjen skal redusere behovet for å orientere seg ved å lese innholdet på skjermen. For å imøtekommne dette kriteriet, utviklet vi en navigasjonsbar som er tilgjengelig på bunnen av alle skjermer.
- **Kriterium 3.2.4 (Konsekvent identifikasjon)** - Kriteriet omhandler at komponenter som har samme funksjonalitet presenteres på en konsekvent måte. Dette skal tilrettelegge for mye av det samme som kriterium 3.2.3., ved at det skal være enklere å forstå funksjonen til ulike komponenter uten å nødvendigvis måtte orientere seg rundt hva de ulike komponentene representerer. Et eksempel på hvordan vi har tatt hensyn til dette kriteriet, er at informasjon om været alltid vil formateres på lik måte. Været på brukerens posisjon formateres likt som når brukeren søker opp et sted.

#### 4.5.2 Mangler ved WCAG og rollen til brukertesting

WCAG-standarden er ikke tilstrekkelig spesifikk til at den fungerer som den eneste kilden til krav. (Tilsynet for universell utforming, n.d.). I tillegg er det heller ikke slik at alle WCAG kravene er relevante til mobile applikasjoner, siden retningslinjene i utgangspunktet er rettet mot web-baserte teknologier som nettsider. W3C har av denne grunn lagt ut anbefalinger knyttet til tolkingen av WCAG kravene for bruk i mobiltelefoner (World Wide Web Consortium, 2018). Disse forsøkte vi å følge når vi så etter relevante WCAG retningslinjer når vi utarbeidet vår egen kravspesifikasjon.

Av disse grunnene spilte brukerundersøkelsene også en stor rolle i å passe på at kravene våre utformet en applikasjon som er universelt utformet. ”Brukertester må til for at nettsider fungerer godt” (Tilsynet for universell utforming, n.d.). Dette mener vi gjelder applikasjonen vår også. Flere av våre krav har derfor sitt opphav i intervjuer vi har gjennomført med målgruppen.

En annen av stor relevans i appens brukervennlighet er familiaritet. Målgruppen vår har erfaring med vær-apper tidligere, noe som betyr at de har en viss kjennskap til hvordan slike apper er utformet. Vi tenkte derfor at det var lurt å følge noen av de generelle trekkene vi finner i apper som yr og storm når vi utformet kravene våre.

## 5. Prosessdokumentasjon

### 5.1 Metode og utviklingsmetodikk

Vi har benyttet oss av Scrumban som rammeverk for organisering i dette prosjektet. Ved å velge en smidig metodikk i utviklingen sikret vi fleksibilitet, god prosjektoversikt, og fokus på brukeren gjennom kontinuerlig datainnsamling og feedback (Sommerville, 2019, 24).

Innenfor Scrum har vi benyttet oss av én rolle, Scrum Master. Scrum Master har vært ansvarlig for å bygge og følge agenda for møter og å være ordstyrer under møter. Vi har hatt en rullerende Scrum Master slik at alle har hatt mulighet til å prøve ut rollen og få erfaring med å organisere og planlegge møter. I tillegg har vi hatt en sekretær på møtene, som dokumenterte prosesser og noterte ned fordelte arbeidsoppgaver. Vi bestemte oss for å ikke inkludere rollen Product Owner fordi vi ikke anså dette som relevant i vårt prosjekt, siden vi ville at alle skulle ha et ansvar og et eierskap til prosjektet og applikasjonen.

For å holde oversikt over arbeidsoppgaver og mål, har vi hatt en felles Kanban tavle. Tavlen er delt inn i kategorier etter hvilke oppgaver som skal gjøres (backlog), hvilke oppgaver vi holder på med, hva som skal ses over av andre, og hvilke oppgaver som er ferdigstilt. De ulike oppgavene er igjen knyttet til hvilke medlemmer av gruppen som har ansvar for å utføre de. Kanban tavlen har vært nyttig i prosjektet for at alle skal kunne ha oversikt, og for å kunne delegere oppgaver og ha en god koordinasjon av arbeidet. Valg av arbeidsoppgaver gikk som regel på egen motivasjon og hva man hadde lyst til å jobbe med.

I prosjektet vårt har vi hatt sprinter over én arbeidsuke med faste fellesmøter på mandager og torsdager. På mandager har vi hatt et sprint planning meeting der vi kartlegger mål for sprinten og delegerer oppgaver. På torsdager har vi hatt stand up meeting der vi har gått gjennom hva vi har gjort, hvordan det har gått, og hva vi eventuelt trenger hjelp til. Vi valgte å ha møter på torsdager slik at det fremdeles er tid og mulighet til å få hjelp, eller til å avdekke for stor arbeidsmengde på enkelte sammenliknet med andre. Ved slutten av hver

sprint har vi hatt et sprint review møte der vi har diskutert hvordan sprinten gikk, hva som eventuelt kan forbedres til neste sprint, og hvordan vi skal prioritere videre. I tillegg hadde vi product backlog reviews der vi analyserte backlogen og fant ut av hvilke elementer i backlogen som skulle prioriteres eller forbedres (Sommerville, 2019, 40).

Vi benyttet oss av et Gantt-diagram (Figur 5) for å illustrere de individuelle oppgavene i prosjektet og for å gi alle en oversikt over hele prosjektet og over tidsforløpet til oppgavene. Gjennom prosjektet ble det oppdaget at ikke alt gikk etter planen, og at noen arbeidsoppgaver tok lengre tid enn beregnet. Gantt-diagrammet var imidlertid nyttig for å kunne visualisere avhengigheten mellom oppgaver, slik at man ikke begynte på oppgaver som var avhengige av at andre oppgaver var fullført først.

## IN2000: Gantt diagram



Figur 5: Gantt diagram

## 5.2 Risikovurdering

For å sikre prosjektets gang valgte vi å kartlegge ulike risikofaktorer av relevans for prosjektets suksess. Risikofaktorene ble senere evaluert i forhold til sannsynligheten for at de fremtrer, og faktorenes konsekvensnivå. Under er oversikten over risikofaktorene, og tiltakene som medfølger deres eventuelle tiltreden:

Risiko	Sannsynlighet	Konsekvens	Tiltak
Identifikasjon av feil i appen	Høy	Mindre alvorlig	Omprioriteringer i sprinten
Vanskeligheter med å implementere funksjonalitet	Middels	Alvorlig	Allokere flere teammedlemmer. Se etter andre alternativer
Mangel på tid	Middels	Katastrofal	Omprioriteringer og økning i teamets tidsbruk

### 5.3 Beskrivelse av oppstartsprosessen

Vi brukte lang tid på å diskutere de ulike casene og hvilken vi ville gå for, før vi til slutt bestemte oss for case 3. Alle på teamet syntes denne casen virket interessant og givende, og at vi kunne være kreative samtidig som vi hadde klare retningslinjer og krav vi måtte forholde oss til i utviklingen.

Det var lite struktur på de første møtene, og gjennom en retrospeksjon kom vi frem til at vi måtte ha en tydeligere agenda for møtene. Vi bestemte oss derfor for å gå for Scrumban som metode for prosjektet fordi dette ville gi en bedre arbeidsflyt og oversikt over arbeidsoppgaver. Vi lagde en Kanban tavle i Trello og bestemte oss for å jobbe i sprinter med en scrum master slik at møtene ble mer strukturert.

I oppstartsprosessen brukte vi også tid på designet av applikasjonen, og flere kom med ideer til design. Vi ble i fellesskap enige om hvilke farger og funksjoner applikasjonen skulle inneholde. Samtidig kom vi frem til at vi ikke hadde nok kunnskap om universell utforming, og vi var usikre på om designet og funksjonaliteten ville være optimal for svaksynte. Vi tok derfor kontakt med ulike organisasjoner for råd, men det viste seg fort vanskelig å få hjelp fra eksterne organisasjoner. Vi bestemte oss derfor for å bruke bekjente for å evaluere design og funksjonalitet.

## 5.4 Sprinter

### Sprint 1

I den første sprinten vår opprettet vi prosjekt i GitHub og lagde en grunnleggende arkitektur for prosjektet. Vi bestemte oss også for hvilke API-er vi ville bruke for å oppfylle krav om funksjonalitet. Etter den første sprinten hadde vi hentet ut værdata fra *LocationForecast*. Ved hjelp av designverktøy som Figma og Canva utviklet vi ulike lavoppløselige prototyper til design, og diskuterte fordelene og ulempene ved de ulike prototypene. Vi innså fort at vi måtte prioritere lesbarhet ovenfor utseende, og satt derfor igjen med tre prototyper. I dette stadiet besluttet vi at det ville være hjelpsomt med tilbakemeldinger fra mennesker som er svaksynte for å se om våre tanker og ideer var i tråd med deres behov. Vi utførte derfor en brukerundersøkelse.

### Sprint 2

I den andre sprinten brukte vi tid på å lese oss opp på universell utforming og retningslinjer innenfor WCAG. Resultatene fra den første brukerundersøkelsen ga oss mye nyttig informasjon, som førte til nye diskusjoner både rundt design og funksjonalitet. Vi ville gjerne at designet skulle være lesbart for svaksynte og samtidig pent, men dette viste seg vanskeligere enn vi hadde tenkt oss. Det var krevende å skape et bestemt design for hele applikasjonen der alle målgrupper vi ønsket å nå ble inkludert. Vi bestemte oss derfor for at vi ville inkludere egne moduser i applikasjonen for fargeblindhet, funksjonalitet for å endre tekstuørrelse slik at svaksynte kunne tilpasse appen etter sin grad av svaksynhet, og text-to-speech funksjonalitet slik at blinde også kunne bruke appen. Vi utviklet en høyoppløselig prototype i Android Studio som vi brukte i en ny brukerundersøkelse. Et viktig mål for denne sprinten var å få på plass MVP for applikasjonen, men dette ble forsinket fordi den ble overskrevet da det kom en ny oppdatering i Android Studio som førte til at filene ikke ville åpne seg.

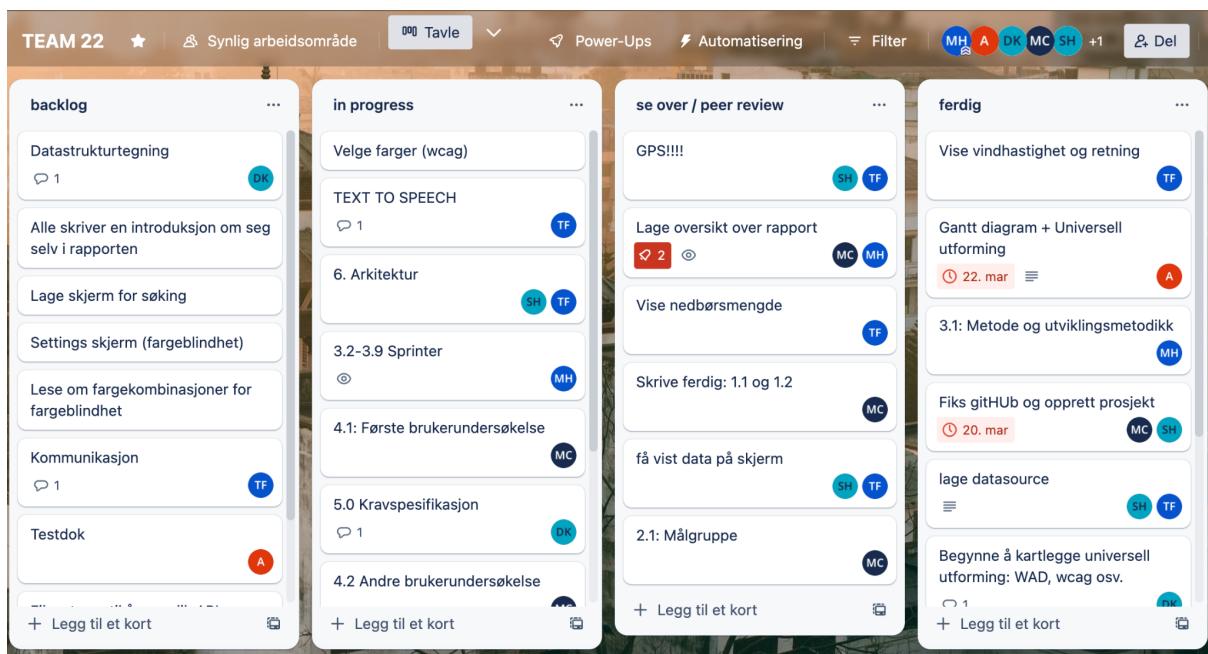
### Sprint 3

I den tredje sprinten hadde vi som mål å implementere funksjonalitet for GPS og timevarsel. Etter flere brukerundersøkelser hadde vi landet på et design for applikasjonen, og vi følte at vi hadde lyktes i å lage et design som både var lesbart for svaksynte og samtidig pent. I denne sprinten oppdaget vi at vi lå litt bak i forhold til hva vi ønsket å inkludere av funksjonalitet i applikasjonen, og vi måtte finne ut av hva vi ønsket å prioritere. Vi så også på mulige navn til

applikasjonen. Denne uken deltok noen av medlemmene av prosjektgruppen på et faglig arrangement om universell utforming, der vi fikk mye lærerik informasjon som vi kunne ta med oss videre i prosjektet.

## Sprint 4

I den fjerde sprinten implementerte vi text-to-speech funksjonalitet. Vi tok også endelige valg rundt hvordan funksjonaliteten til knappene nederst på skermen skulle være. I tillegg implementerte vi timevarsel i applikasjonen. Denne uken oppdaget vi at vi måtte rydde opp i koden slik at den fulgte prinsippene om MVVM og var mer lesbar. Vi startet også på design av logo for applikasjonen.



Figur 6: Kanban-tavle fra Sprint 4

## Sprint 5

I den femte sprinten brukte vi tid på å sette all funksjonaliteten sammen, da mange av funksjonalitetene hadde blitt laget isolert på egne branches i Git. Funksjonalitet for GPS ble ferdig denne uken, så da kunne vi begynne på implementasjonen av søk etter værinformasjon ved et valgt sted. Skjermen for innstillinger ble laget, men den manglet fremdeles litt funksjonalitet i tillegg til at den ikke var lesbar for svaksynte, så designet måtte endres videre. Denne uken bestemte vi oss for logo og navn til applikasjonen.

## Sprint 6

I den sjette sprinten hadde vi et mål om å få ferdig all ny utvikling av appen, slik at vi kunne bruke resten av tiden til å finpusse funksjonalitet og design. I denne sprinten fant vi ut at vi måtte bytte oppsettet av hovedskjermen for at all teksten på skjermen skulle være lesbar og for at applikasjonen skulle passe for ulike typer skjermer og emulatorer. Søke-skjermen ble ferdig, og vi snakket også om å legge til funksjonalitet som lagrer tidligere søk. Denne uken fant vi ut at det var nødvendig å lage egne moduser for fargeblindhet. Etter testing av kontrastforhold og simulering av fargeblindhet oppdaget vi at designet likevel var tilpasset fargeblinde, og at de ulike modusene for fargeblindhet ikke var nødvendige for å oppnå sluttresultatet vi ønsket.

### **Sprint 7**

I den syvende sprinten hadde vi som mål å få ferdig appen. Denne uken kom søker-skjermen på plass med funksjonalitet som lagret tidligere søk i tillegg til forslag som dukker opp når man begynner å skrive inn. Ved slutten av sprinten innså vi at det var mange feil og mangler i koden. Dette førte til at vi måtte nedprioritere ny funksjonalitet og heller bruke tid på å rette opp feilene vi hadde. Mange av disse feilene ble til slutt teknisk gjeld.

### **Sprint 8**

I den åttende sprinten hadde vi som mål å skrive ferdig rapporten, slik at det bare var renskriving igjen. Det var fremdeles noen ting som gjenstod på appen ved slutten av sprinten; menyen nederst på skjermen hadde ikke riktig farge, vi manglet en skjerm for feilmeldinger, og vi måtte legge til funksjonalitet for vertikal scrolling. Vi manglet også å legge inn logo og navn på applikasjonen.

### **Sprint 9**

I den niende og siste sprinten gjorde vi vårt siste arbeid på appen og rapporten. Ved slutten av prosjektet måtte vi gjøre flere omprioriteringer i backlogen, slik at vi oppnådde de viktigste kravene. Vi satte all koden sammen, og fikk endret fargene på menyen nederst på skjermen. Vi fikk også lagt til skjerm for feilmeldinger og splash-screen.

## **5.5 Teknisk gjeld**

Ved søk på ulike steder vil ikke klokkeslettene i timevarselen endre seg etter hvilken tidssone valgt sted befinner seg i. Dersom vi hadde hatt mer tid ville vi ha fikset opp i dette, men vi så

ikke på det som en viktig prioritering i ferdigstillingen av applikasjonen. Klokkeslettet vil heller ikke endre seg når man søker opp ulike steder. Dette var fordi vi beregnet bruk i Norden.

Da vi utformet designet for appen, brukte vi Oslo som stedsnavn. Dette førte til at vi fikk problemer senere med lengre stedsnavn, da det havnet bak vær-symbolet. Vi måtte derfor endre størrelsen på vær-symbolet slik at designet ville vise lengre stedsnavn, noe som førte til at symbolet ikke var like synlig. Dersom vi skulle ha et større symbol, måtte vi ha endret mye på designet, noe vi ikke fikk tid til da vi oppdaget dette sent i utviklingen.

Vi startet med å bygge HomeScreen med en Box, noe som førte til dårlig skalerbarhet slik at den ikke er optimalisert for telefoner av mindre skermstørrelse (Under 5"). I ettertid ser vi at vi burde ha startet annerledes i utviklingen av hjemskjermen.

API-ene fra Google blir kalt på fra ViewModel når de egentlig burde blitt kalt på av Datasource for å følge MVVM arkitekturen. Vi trodde det ville være en enkel justering, og prioriterte dette for sent. Men da vi forsøkte å endre dette, førte det til komplikasjoner som vi ikke hadde nok tid til å fikse. Derfor lot vi det være.

Innstillinger-skjermen er ikke scrollable. Vi innså dette litt for sent, ettersom vi ikke utviklet noe særlig funksjonalitet her. Dette fører til at innstillinger-menyen ikke er skalerbar på alle skermstørrelser.

## 6. Produktdokumentasjon

### 6.1 Arkitektur

#### 6.1.1 Model-View-ViewModel (MVVM)

MVVM er et arkitektonisk designmønster som i stor grad benyttes til utvikling av moderne apper. Appens logikk deles inn i tre hoveddeler: Model representerer dataene og logikken i systemet, ViewModel som representerer det grafiske grensesnittet brukeren kan interagere med, og i View håndteres kommunikasjonen og dataflyten mellom modellen og ViewModel

(Martin, 2023). En slik arkitektur var relevant for å bidra til strukturert utvikling av apper ved å separere ansvar til ulike modeller.

En av fordelene med MVVM-arkitekturen er at separasjonen av de ulike komponentene skaper høy kohesjon og lav kobling. Høy kohesjon betyr at hver enkelt komponent kun er ansvarlig for en spesifikt definert oppgave. Lav kobling betyr at komponentene er uavhengige av hverandre, slik at endringer i en komponent ikke har noen uforutsette konsekvenser i andre komponenter (Montiel, 2018). For appen skaper dette fleksibilitet som forenkler vedlikehold og testing av koden.

Tidlig i utviklingen valgte vi å bruke dette designmønsteret basert på tidligere erfaring fra emnet. Etterhvert i utviklingen ble det krevende å opprettholde denne standarden med tanke på at det var mange som jobbet med utviklingen samtidig. Vi implementerte ny funksjonalitet raskt og effektivt, og dermed forsvant fokuset på MVVM. I ettertid måtte vi reformatere og gjøre endringer for å oppnå kravet om høy kohesjon og lav kobling.

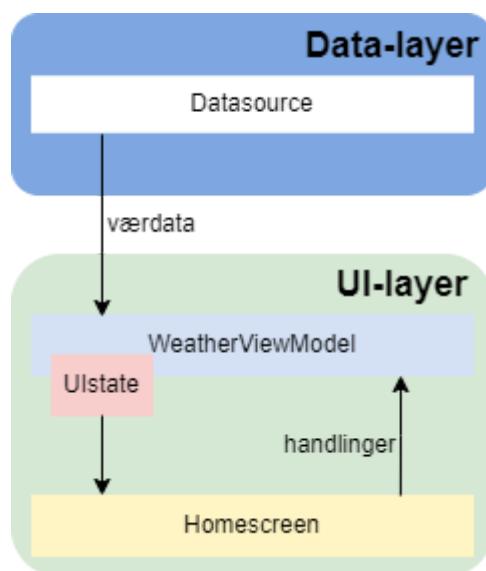
### 6.1.2 Front-end- og back-end-utvikling i *SightCast*

Jetpack Compose ble valgt som vårt primære UI-verktøy for brukergrensesnittet i applikasjonen. Dette verktøyet skiller seg fra det tradisjonelle XML-baserte layoutsystemet. Dette er fordi det også indirekte samhandler med applikasjonens back-end. Denne samhandlingen ser en via mekanismer som hendelseshåndtering og tilstandsbehandling, som gir muligheten til å oppdatere og lagre værdata mens appen kjører (DeLaGrana, 2023). Dette er synlig på søkeskjermen, hvor brukerne kan enkelt lagre og få tilgang til værdata for nylige søker på steder med ett tastetrykk.

I videre utviklingen av appen ble det forsøkt å forholde seg til "Android-developers guide to architecture". Gitt denne tilnærmingen er appen delt inn i 2 deler: UI-layer som representerer front-end og Data-Layer som representerer back-end (*Guide to App Architecture*, 2023). Vårt UI-lag består av ulike Screen.kt-filer som brukes for å generere det visuelle i appen. ViewModel som fungerer som en slags SSOT (Single source of truth), der all data som vises på skjermen kommer direkte fra UiState gjennom ViewModel. De forskjellige skjermene håndteres av en BottomNavigationBar som er definert i Navigation.kt. Data-Laget er representeret av Data Source, GPS og data-klassene for håndtering av værdata. Det er Data

Source som har ansvar for å hente ut data fra *locationforecast* APIet, der den oppretter et objekt av typen WeatherData, som videre oppdaterer alt fra temperatur til vindhastighet. De 3 resterende Google API-ene kalles på fra ViewModel. ViewModel sender en HTTP GET-forespørsel til disse API-ene for å hente ut koordinater og stedsnavn basert på lokalitet.

Vi kan se sammenhengen mellom datalagene når Main Activity gjør ett funksjonskall på ViewModel, og når den da videre oppdateres vil den kalle på Data Source (vårt repository), og bruker så returverdiene for å oppdatere værdataene i UIstate. Disse kan da direkte aksesseres i HomeScreen, og slik blir front- og back-end i applikasjonen mindre separert.



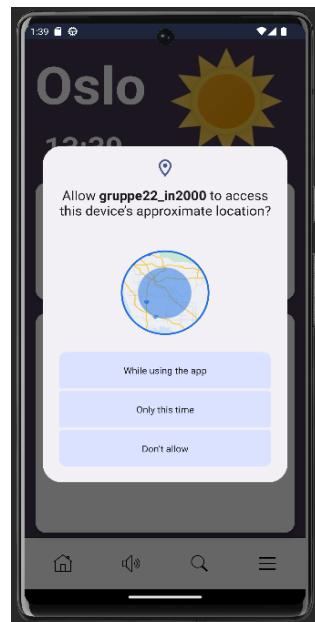
Figur 7: Viser interaksjonen mellom Data- og UI-layer

## 6.2 API

### 6.2.1 Valg av API nivå

API-nivå 33 ble benyttet som targetSdk, da dette er Androids mest moderne nivå og versjon, *Android 13 Tiramisu*. Slik fikk vi tilgang til mange nye funksjoner, oppgraderinger og optimaliseringer, sammenlignet med tidligere nivåer. Nyere API-nivåer kommer ofte med ytelsesforbedringer som kan hjelpe appen å kjøre mer jevnt og effektivt. Samtidig har forbedrede sikkerhetsfunksjoner som kan beskytte applikasjonen mot potensielle trusler. Ved bruk av API-nivå 33 kan man dra nytte av disse optimaliseringene, som videre kan gi bedre brukeropplevelse.

En av nye funksjonene vi brukte, var Androids oppdaterte *permissions* (*Request Location Permissions*, n.d.). I Tiramisu kan brukerne få valget "choose approximate location" (se figur 8), fordi dette sikrer bedre personvern, da det her brukes telefonens tilkobling til internett for å finne ut hvor brukeren er i en generell nærhet. Dette krever også færre bakgrunnsressurser, enn det gjør ved søk av mobilens presise lokasjon, da det der brukes GPS og sensorer.



Figur 8: Appen spør om tillatelse til å aksessere brukerens *omtrentlige lokasjon*

I henhold til Googles nye retningslinjer må nye apper som publiseres på Google Play ha et targetSd-nivå på 33. (*Target API Level Requirements for Google Play Apps*, n.d.) Dette kravet ga oss enda en grunn til å velge Androids nyeste API-nivå, da vi vurderte muligheten for å publisere appen i fremtiden. Dermed var det viktig at appen vår opprettholdt samme kvalitet og stabilitet som andre apper på Google Play, både når det gjaldt front-end og back-end.

### 6.2.2 Implementerte APIer

En rekke APIer ble brukt for å kunne fremstille den korrekten temperaturen for en gitt lokasjon. Det ble benyttet APIer fra to forskjellige kilder. Det ene er fra Meteorologisk Institutt (MET) med navn *LocactionForecast* (Weather Forecast API 2022), og tre fra Google Maps; *Places API*, *Geocoding* og *Place Autocomplete API*.

Hensikten med å bruke *LocationForecast* var å hente værdata, og dette var også et anbefalt API fra Meteorologisk institutt. Dette APIet tilbyr brukeren oppdatert værdata for de valgte lengde- og breddegradene, både for nå og flere dager fremover (*Locationforecast*, n.d.).

*Geocoding* ble brukt for å hente ut lokasjoner sammen med *Places API*, slik at brukerne kan søke etter et valgfritt sted. *Geocoding* konverterer koordinater fra *Places API* for å vise stedsnavn som en tekststreng (Geocoding API Overview, 2023). Ved å integrere disse API-ene, kunne applikasjonen tilby en søkefunksjon som var presis, rask og brukervennlig. Samtidig ble *Autocomplete API* brukt med hensyn til brukerundersøkelsen, da det ble oppfattet at færre tastetrykk ville gjøre appen enklere å bruke. Ved bruk av dette APIet vil ulike stedsvalg dukke opp når brukeren begynner å skrive i søkerfeltet, noe som forbedrer ytelsen og forholder seg til appens formål.

Å lage en fungerende GPS tok lengre tid enn planlagt. da den i starten ikke alltid ville fungere. Vi fant ut at problemet lå i *LocactionRequest*, da typen forespørsel ikke samkjørte bra med vår type lokasjonshenting (*ACCESS\_COARSE\_LOCATION*). Dette fikk vi løst opp i og hadde videre i utviklingen ingen problemer med henting av mobilens koordinater.

## 6.3 Teknologier og verktøy

### 6.3.1 Kommunikasjon

Messenger ble valgt til å overholde hovedkommunikasjon, da dette er en applikasjon alle i teamet var godt kjent med. Dette ble brukt til rask kommunikasjon angående planlegging av møter og korte oppdateringer. For app-relatert kommunikasjon og samkoding ble Discord benyttet. I tillegg brukte vi Microsofts Teams for å kommunisere med veilederne våre for mer formelle samtaler. Gjennom en kombinasjon av disse kommunikasjonsverktøyene kunne vi effektivt samarbeide.

### 6.3.2 Utvikling

For utvikling har vi benyttet oss av Google sin IDE, Android Studio. Android Studio tilbyr en mengde funksjonalitet, inkludert en kodeeditor med tilrettelegging for avansert feilsøking og testing, emulering av Android-enheter, Git integrering for versjonshåndtering og støtte for flere språk.

For deling og lagring av kode opprettet vi et felles prosjekt på GitHub. Ved å bruke GitHub sin versjonshåndtering ga det muligheten til å spore endringer i kildekoden, og returnere til en fungerende versjon dersom noe skulle gå galt. Dette gir oss muligheten til å eksperimentere og teste funksjonalitet i egne grener.

Google Drive ble brukt til lagring av all data som ikke er koderelatert, som for eksempel møtereferater, figurer og diverse notater. Siden Google Drive er skybasert sikret det at alle medlemmene av gruppen alltid hadde tilgang til den nyeste versjonen av alle dokumentene.

## 6.4 Kvalitetsegenskaper ved appen

Vi evaluerte kvalitets egenskapene til appen gjennom å bruke ISO 25010 standarden. Den inkluderer ulike evalueringskriterier egnet til å vurdere ulike aspekter av appens kvalitet. Alle kriteriene var ikke av relevans til oss, vi valgte derfor ut de som vi mente var viktigst. Valget av hva vi mente var viktigst var i stor grad basert på de ikke funksjonelle kravene våre, siden disse spesifiserer kvalitetsattributtene til appen (Visuresolutions, n.d.).

### 6.4.1 Brukbarhet

Kriteriet om brukbarhet var det viktigste vi identifiserte. Vurderingen på dette kriteriet skjedde gjennom den siste brukerundersøkelsen vår. Den generelle opplevelsen var at appen var lett å lære, og samsvarte godt med behovene til svaksynte. Det bør bemerknes at brukervennlighet i stor grad baserte seg på størrelsen til skjermen. Skjermer på 5 tommer og oppover var god brukeropplevelse, men skjermer under denne størrelsen gjorde at applikasjonene var vanskelige å ta til bruk.

### 6.4.2 Ytelse

Ytelse identifiserte vi også til å være et viktig vurderingskriterium. Dette ble også i hovedsak vurdert utfra tilbakemeldinger fra intervju 3. Oppfattelsen fra intervjuet var at applikasjonen var på den litt tregere siden, men ikke til en grad som var avgjørende. Dette var da kun kvalitativ data, og vi mangler tallfestet kvantitativ data på appens ytelse, mangelen på dette forekom av tidsmangel på vår side, siden evalueringen av appen skjedde på et senere tidspunkt i prosjektets livssyklus enn ønsket.

### 5.4.3 Sikkerhet

Sikkerhet er et viktig aspekt innenfor ISO 25010 som innebærer å tilfredsstille konfidensialitet, integritet, og tilgjengelighet til appen. Ved å modellere truslene innenfor appen kan man oppdage potensielle sikkerhetstrusler og evaluere truslene for kravene til appen. Ved å knytte noen av sikkerhetsmålene til truslene kan man få en generell oversikt over mulige mottiltak. I tillegg er det verdt å merke seg at appen ikke har mange trusler fordi den ikke har funksjonalitet til å søke etter brukerens personopplysninger med unntak av stedstjenester.

Kvalitetskrav (ISO 25010)	Trusler	Tiltak
Konfidensialitet	Uautorisert tilgang til Google API nøkkel	Krypter API nøkkel og ikke push til github
Integritet	Feil data for brukerens gitte posisjon	Valider med testing at appen viser korrekt data for brukerens posisjon
Tilgjengelighet	Overbelastning av tjenesten ved misbruk av API nøkkel	Sørg for at API nøkkel brukt ikke overskriver kontoens grense

## 6.5 Videreutvikling

Vi hadde et ønske om å implementere en mer avansert splash-screen som ville kommet opp mens appen og tilhørende data ble lastet inn, men dette fikk vi ikke tid til. Splash-screen skulle inneholde en animasjon som inneholdt appens logo, og skulle vises mens en egen “loading state” ble initialisert i ViewModel. Denne loading-staten skulle laste inn data i bakgrunnen.

Vi hadde også et ønske om at brukeren kunne få mer informasjon om været ved langtidsvarsel ved å trykke på en dag fremover i tid i 7-dagers værvarselet. Brukeren ville da ha fått opp en ny skjerm eller en utvidet meny der de også kunne se timevarsel for den dagen.

I tillegg hadde vi lyst til å gi brukeren mulighet til å endre tekststørrelsen. Vi startet implementasjonen av dette i innstillinger-menyen gjennom en slider som justerer

tekststørrelsen til en setning etter brukerens preferanse. Implementasjonen av dette for hver tekststreng i applikasjonen var mer krevende enn antatt, og vi hadde heller ikke nok tid. Vi fikk lagt inn knapper for innsending av tips, kontaktinformasjon, innstillinger for personvern og personvernerklæring. Disse har ingen funksjon enda, så dette er noe vi ville ha utviklet videre. Vi ville også at brukeren kunne lagre steder som sine favoritter.

## 6.6 Warnings

Vi hadde to warnings i programkoden vår, som begge kom fra funksjoner. Den ene var `getIdentifier()` som henter ID til drawable basert på strengnavn, fremfor å aksessere den direkte med identifikatorer. Vi måtte bruke den for å finne navn på bilderessurser basert på en streng da vi i dataene fra APIet kun har filnavn, og vi vurderte dette til en bedre måte enn å legge inn 83 variabler og benytte et “when- statement” for å sjekke disse.

Den andre warningen gikk ut på at funksjonen `getFromLocation()` ikke lenger blir gitt offisiell støtte, men at den fremdeles er tilgjengelig for bruk. Det finnes en nyere versjon, men vi valgte å holde oss til den forrige da vi allerede hadde implementert den.

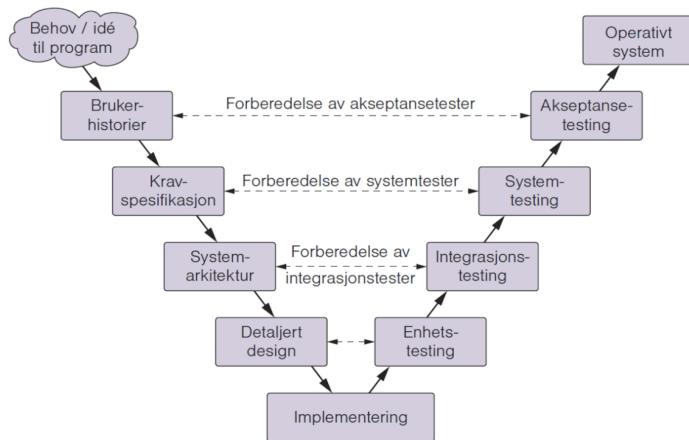
I App-Modulen sin build.gradle-fil er det 21 warnings, men disse kommer av at det eksisterer nyere versjoner av enkelte dependencies. Vi har valgt å ikke endre versjoner, med mindre det er et kjent problem knyttet til akkurat denne versjonen, for å beholde stabilitet i appen.

## 7. Testdokumentasjon

Testing er en viktig del av enhver utviklingsprosess og hjelper til med å sikre at en programvareapplikasjon fungerer som den skal. Statisk og dynamisk testing ble benyttet i prosjektet. Statisk testing involverer gjennomgang av kode uten å faktisk kjøre programvaren. Dette ble gjort gjennom reformatering av koden og gjennomgang av dokumentasjon fra Android Studio. Enhetstester og integrasjonstester er en form for dynamisk testing som omhandler kjøring av kode for å evaluere komponenter av koden (Gillis, n.d.). Dette var essensielt for å avdekke potensielle problemer underveis i utviklingsprosessen.

## 7.1. Planlegging og gjennomføring

Testing av appen ble gjennomført kontinuerlig i løpet av prosjektet, med jevnlige tester av både utseendet og koden. Ved å teste og planlegge tidlig kunne vi oppdage avvik og mangler i appen, noe som førte til kontinuerlige forbedringer. Testingen ble gjennomført i henhold til V-modellen, som er en systemutviklingsmodell der testing integreres i hver fase av produktets livssyklus (Florea, n.d.). Tidlig i utviklingen ble det klart at appen måtte oppfylle visse krav, inkludert synlighet, brukervennlighet og minimalistisk design for svaksynte. Før vi begynte å kode, lagde vi prototyper av appens utseende ved hjelp av ulike designverktøy for å visualisere ideen og teste dens realistiske funksjonalitet.



Figur 9: V-modellen hentet fra forelesningsopptak (Vihovde, 2023)

For å oppfylle kravene for MVVM ble koden sammenlignet med dokumentasjonen fra Android Studio. Under implementeringsfasen ble integrasjonstesting utført for å sikre korrekt implementering av alle komponentene. Det var også viktig å teste dataene fra API-et, derfor benyttet vi ulike verktøy for enhetstesting som JUnit for presis og grundig testing av individuelle enheter. I tillegg ble testRunner brukt som et rammeverk som tillater kjøring av tester på Android enheter. Dette verktøyet bidro til å sikre nøyaktige resultater og verifisere at kravene ble tilfredsstilt, noe som resulterte i en fungerende applikasjon i henhold til forventningene.

## 7.2. Enhetstesting

Oversikt over enhetstestene befinner seg i mappen `com.example.gruppe22_in2000 (androidTest)`. Ulike tester ble gjennomført for å teste om de enkelte delene i appen fungerer som det skal. Gjennom enhetstester verifiserte vi at temperaturen var riktig for dens tilhørende tid og sted. I utviklingen av appen var det et spesielt fokus på ViewModel, som tok seg av håndteringen av værdata og samhandling med andre filer.

Tests	Duration	Pixel_6_API_33
✓ Test Results	330 ms	10/10
✓ UnitTest	330 ms	10/10
✓ testGetDirectionOnlyArrow	29 ms	✓
✓ testGetCoordinatesForTest	258 ms	✓
✓ testWindFromDirection	3 ms	✓
✓ testNext6HoursPrecipitation	1ms	✓
✓ testUpdateUiState	31ms	✓
✓ testClearSearchResults	3 ms	✓
✓ testNext6HoursTemperature	1ms	✓
✓ testNext1HoursSummary	1ms	✓
✓ testLocationDetails	1ms	✓
✓ testFindClosestTimeseriesIndex	2 ms	✓

Figur 10: Viser ti vellykkede tester

Å teste funksjonene knyttet til værdata er kritisk for å sikre at appen presenterer nøyaktig og relevant informasjon. Dette bidrar til brukerens tillit til å tilfredsstille deres krav til appen. Disse testene sikret at riktig data ble sendt fra ViewModel til View, noe som er viktig for å opprettholde en konsistent brukeropplevelse. Andre tester inkluderte verifikasiing av initiale verdier, håndtering av søkeresultater, validering av plassering og koordinater i henhold til forventede API-data, og bekrefteelse av korrekte temperaturdata. De viktigste funksjonelle kravene som omfatter arkitekturen og funksjonaliteten ble testet i samsvar med enhetstestene. Dette er avgjørende fordi alle filene er avhengig av vær-og stedsdata. Uten testing av funksjonelle krav kan det oppstå feil i programvaren, noe som kan føre til misfornøyde brukere, og i verste fall tap av ressurser. Tabellen under forklarer testene:

Tester	Beskrivelse	Krav
getDirectionOnlyArrow	Denne testen sjekker om funksjonen returnerer riktig retningssymbol for en gitt vinkel	Appen skal på forsiden oppgi riktig vindretning og hastighet
testWindFromDirection	Testen sjekker om vindretningen går overens med vindretningen fra API	
findClosestTimeseriesIndex	Testen finner indeksen til den nærmeste tidsserien i listen	Kravet for å vise utvikling av temperatur over 24 timer og 7 dager.
testNext6HoursPrecipitation	En test som viser nedbørsmengde for de neste 6 timene	Oversikt over nedbørsmengden på forsiden.
testNext6HoursTemperature	Testen verifiserer at temperaturen viser riktig for neste 6 timer	Systemet skal på forsiden tilby oversikt over temperaturens utvikling over 24 timer.
testRemoveSavedLocation	validerer at clearSearchResults-funksjonen har evnen til å fjerne elementer.	Appen skal gi brukeren valget om å slette tidligere søk, ved trykk på symbol.
getCoordinatesForTest	Testen validerer at den plasseringen og koordinatene stemmer overens med forventede verdier fra API-data	Appen skal gi brukeren muligheten til å søke opp ulike posisjoner, slik at disse vises fremfor deres nåværende posisjon.
testLocationDetails	Bekrefter om bredde- og lengdegrad samsvarer med de opprinnelig angitte verdiene.	Systemet skal på forsiden oppgi temperaturen til brukerens nåværende posisjon.
testNext1HoursSummary	Bekrefter at værsymbolet er korrekt visualisert	Været skal visualiseres ved bruk av vær-symboler
testUpdateUiState	Testen validerer at uiStaten blir oppdatert med ny lokasjon	Appen skal hente data asynkront

### 7.3. Integrasjonstesting

Integrasjonstesting er en viktig del av utviklingsprosessen der de individuelle delene av programvaren kombineres og testes i en helhet. Denne typen testing tar hensyn til både funksjonelle og ikke-funksjonelle krav for å imøtekommе appens formål. Den valgte tilnærmingen for integrasjonstesting var MVVM, der de øverste nivåene av systemet ble først testet, nemlig datasource, og deretter ned i hierarkiet i rekkefølgen View, ViewModel, SearchBar og InstillingerMenu. Dette ble utført ved hjelp av logcat og appkjøring for å teste samspillet mellom arkitekturens komponenter.

Logcat som inneholder meldinger skrevet i appen ble brukt for å teste funksjoner knyttet til værdata for å sikre at data i UI-state samsvarer med UI-tilstanden (*Logcat Command-Line Tool | Android Studio*, 2023). Blant annet blir manglende data for et bestemt sted logget som "null", noe som hjalp med å identifisere unormal oppførsel tidlig og redusere teknisk gjeld. Deretter ble View opprettet for å gi en visuell oversikt over appen og avdekke eventuelle utfordringer knyttet til utseende og layout, for eksempel farger, størrelse og unødvendig tekst. Videre ble ViewModel koblet til View for å teste nøyaktig henting og riktig visning av data på skjermen ved hjelp av logcat og feilhåndtering.

I tillegg var gjentatte kjøringer av appen nødvendig for å teste om appens utseende var forståelig og lesbart. Med tanke på designvalget og målgruppen for appen var det essensielt å stadig sjekke om temperaturen, symbolet og annen værdata ble presentert på en lesbar måte. I sluttfasen av utviklingen ble hendelseshåndtering testet ved å simulere endringer i Datasource, mangel på nettverk, og brukerens inndata for å verifisere at riktig respons ble gitt. Samtidig ble det utført tester for å sikre at appen fungerte på ulike emulatorer uten å krasje, og responstiden mellom komponentene ble evaluert.

### 7.4. Testing av universell utforming

Gjennom brukerundersøkelsene fikk vi en bedre forståelse for hva som var nødvendig for å imøtekommе brukernes behov. Tilbakemeldingene fra brukerundersøkelsene bidro til å skape en mer brukervennlig app for svaksynte.

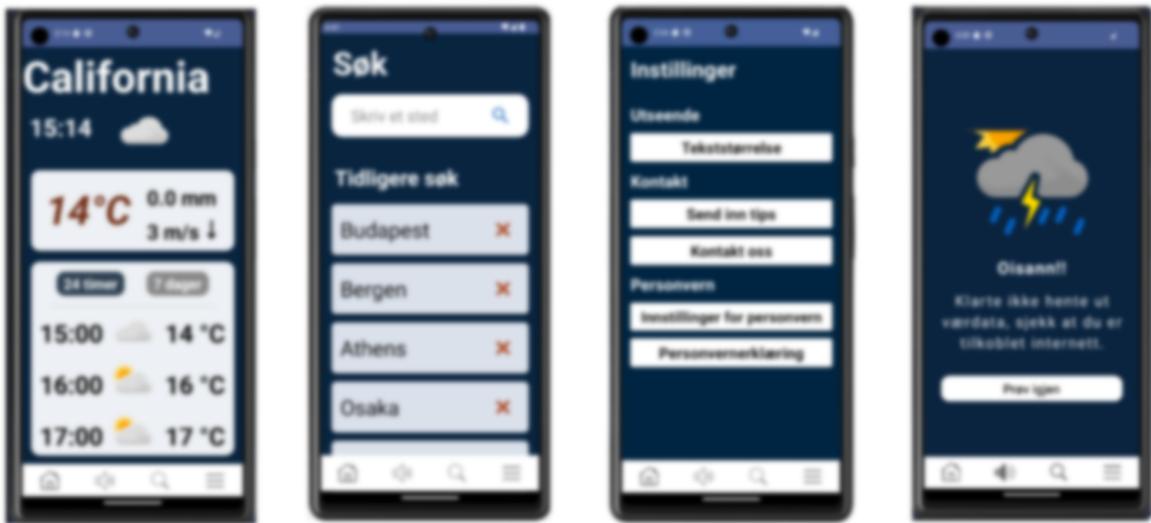
For å sikre at appen vår er så tilgjengelig som mulig, tok vi i bruk Adobe Color Contrast Analyzer-tjenesten. Dette verktøyet hjalp oss med å velge en passende fargekombinasjon som oppfylte **Suksesskriterium 1.4.6 Kontrast (avansert)**. For å sikre optimal brukervennlighet for svaksynte og fargeblinde brukere, ble et kontrastforhold på 14:1 implementert på den endelige hjemskjermen.

For å visualisere værdata eksperimenterte vi med ulike vær-symboler som kunne hjelpe brukeren å forstå værutviklingen over flere dager. Igjen var det viktig å sørge for at alle værsymbolene hadde et kontrastforholdet som egnet seg for svaksynte. Derfor var det nødvendig å utføre en siste brukerundersøkelse for å sørge for dette.

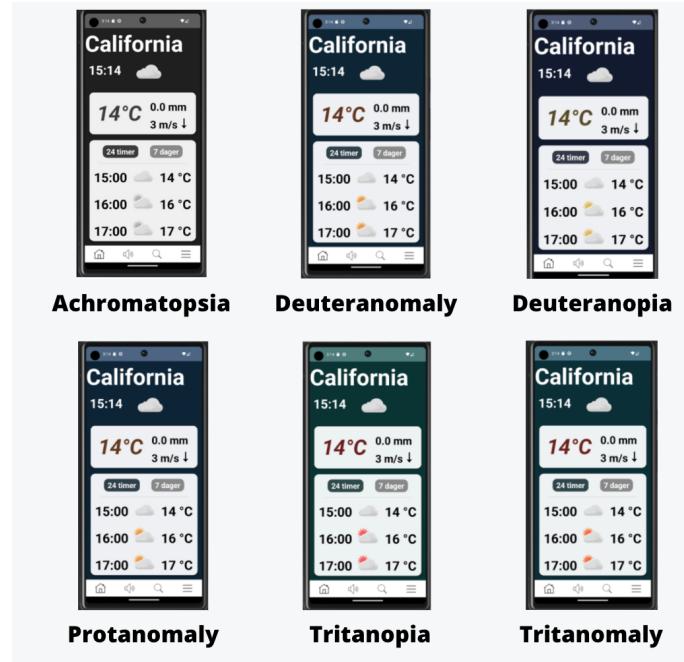


Figur 11: Viser test for de ulike kontrastforholdene som er godkjent etter WCAG

En tilgjengelighetstest med Google Chrome utvidelsen *Funkify* ble utført for å videre vurdere brukervennligheten. Ved hjelp av dette verktøyet kunne man simulere svaksynhet og fargeblindhet ved å justere graden av ulik funksjonshemming. Under testingen ble det tatt hensyn til de seks hovedvariantene av fargeblindhet. Etter testen ble det oppdaget at de endelige fargekontrastene var så høye at designet var tydelig for fargeblinde. Derfor var det ikke nødvendig å utvikle videre funksjonalitet for endring av farger i designet.



Figur 12: Viser appen med synsstyrke -2



Figur 13: Viser appen med seks ulike former for fargeblindhet

## 8. Refleksjon

### 8.1 Samarbeid

Vi hadde en fin sammensetning av kunnskaper og erfaringer på gruppen. Vi kom fra tre ulike linjer og hadde tatt ulike emner som kunne utfylle hverandre. Tidlig i utviklingen ble det derfor naturlig for oss at noen tok ansvar for programmeringen, mens andre hovedsakelig

jobbet med rapportskriving. Dette var fordi noen hadde mer erfaring med programering, mens andre hadde mer erfaring med å skrive lignende rapporter og prosjektoppgaver. I prosjektet var det hovedsakelig fire av medlemmene som programerte. Senere i utviklingen viste det seg vanskelig å inkludere alle i kodingen av applikasjonen, da det ble mye nytt å sette seg inn i, og det ville ta mye tid fra prosjektet. I ettertid ser vi at vi kunne inkludert alle i flere arbeidsoppgaver fra starten, og utført parprogrammering i større grad. Samtidig var alle inkludert i beslutninger som ble tatt både når det gjaldt design og funksjonalitet. Det var heller et skille i hvilke medlemmer av gruppen som implementerte funksjonaliteten.

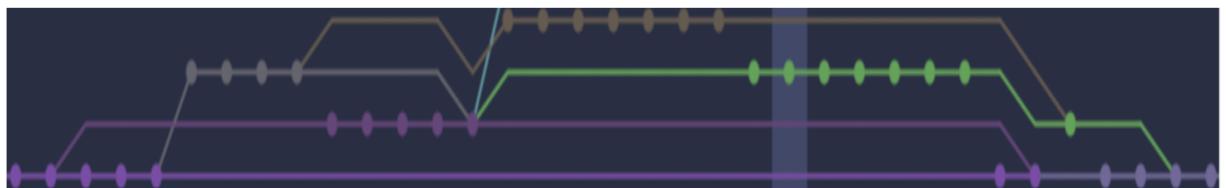
Vi jobbet mye i fellesskap på møtene med alle sammen til stede. Slik sikret vi enighet og en felles forståelse for funksjonalitet og design. Ellers arbeidet alle med sine arbeidsoppgaver hver for seg eller i mindre grupper. Innad i gruppen var det rom for ærlighet og å spørre om hjelp. Vi var også en sosial gruppe og vi ble godt kjent i løpet av prosjektet. Alle tok selv initiativ ved fordeling av oppgaver slik at alle fikk en oppgave de trivdes med eller som samsvarer med deres kunnskap. Dersom noen slet med en oppgave, diskuterte vi oppgaven i plenum før så å eventuelt fordele oppgaven slik at ingen satt med for mye arbeid alene.

Vi opplevde få interne konflikter i løpet av prosjektet. I oppstartfasen opplevde vi at mange gjerne ville snakke og hadde mange ideer, men at det fort førte til mye tidsbruk og lite struktur. Dette løste seg fort ved å ha en bedre struktur på møtene. I tillegg hadde vi ulike meninger om design og funksjonalitet, men dette ble heller berikende og førte til at applikasjonen ble forbedret. Vi kommer fra ulike fagfelt og har ulikt fokus på hva som er essensielt, og dette førte til at vi fikk en applikasjon som oppfylte krav fra flere fagområder.

Vi opplevde noen kommunikasjonsproblemer i utviklingen da ikke alle hadde like god kjennskap til GitHub og om hvordan man jobber sammen med å programmere. Dette førte til mer arbeid underveis der vi brukte mye tid på å rydde opp i koden slik at alt fungerte sammen på en god måte. Alle jobbet på hver sin branch over litt for lange tidsperioder, og merging ble derfor en større jobb enn det hadde trengt å være. Etter figur 15 har én på gruppen jobbet direkte på en klone av Main med det siste resterende av funksjonalitet.



Figur 14: Starten av prosjektet



Figur 15: Slutten av prosjektet

## 8.2 Utvikling

Vi hadde mange mål for de ulike sprintene som ikke var realistiske. Gjennom prosjektet oppdaget vi at noen arbeidsoppgaver var mer krevende eller at de tok lengre tid enn først antatt. Dette førte til flere forsinkelser i arbeidet. Blant annet brukte vi lang tid på GPS-funksjonalitet. Ved slike tilfeller valgte vi å heller fortsette der vi kunne ved å for eksempel lage skjeletter til skjermer eller design, for så å legge inn verdiene vi trengte senere. Samtidig var det også noen arbeidsoppgaver som var enklere eller tok mindre tid enn antatt, som for eksempel text-to-speech-funksjonalitet.

Ved slutten av prosjektet oppdaget vi flere feil i appen som ble for tidkrevende å ordne opp i. I ettertid ser vi at vi burde ha testet appen på flere emulatorer og med ulike stedsnavn tidligere, slik at vi hadde oppdaget disse feilene før. MVVM prinsippene stod sentralt i starten av utviklingen, men fokuset forsvant underveis i utviklingen. Dette skyldtes at vi gjerne ville legge til mye ny funksjonalitet samtidig som det var lite samkjøring og kommunikasjon ved implementeringen.

## 8.3 Konklusjon

Gjennom prosjektet har vi lært mye om både prosjektarbeid og apputvikling. Ingen av oss hadde erfaring med å jobbe med et så stort prosjekt over så lang tid før, og gjennom dette arbeidet har vi fått prøve å benytte smidige metoder. Vi har lært mye om prosjektplanlegging, kommunikasjon og samkoding i GitHub. I apputviklingen har vi fått tekniske kunnskaper og

ferdigheter. Dette prosjektet ga oss mulighet til både å arbeide med konkrete teknologier, og lære å finne relevant informasjon.

Vi har alle fått mulighet til å utvikle vår egen selvinnsikt, empati, evne til å se gruppens mål fremfor egne mål, og å øve på å håndtere konflikter i fellesskap. I tillegg har vi fått erfaring med å skape en brukerorientert løsning som følger lovbestemt universell utforming.

## 9. Vedlegg

### 9.1 Teamavtale

#### Tilstedeværelse

1. Kom på alle møter, og dersom du ikke har mulighet til å komme, må det gis beskjed på forhånd (helst dagen før).
2. Alle må ikke møte opp dersom det ikke passer. Det skal være mulig å møte opp digitalt (discord).

#### Tidsbruk

1. Felles faste møter mandager og torsdager. Ellers har vi møter etter behov, eller møtes i mindre grupper.
2. Målet er at oppgavene skal bli ferdig, man bruker den tiden man trenger på det. Dersom man føler på at man bruker for mye tid, skal man gi beskjed om det.
3. Ingen skal jobbe alt for mye, og vi skal forsøke så godt vi kan å fordele oppgaver slik at alle jobber omtrent like mye.

#### Forventninger til den enkeltes bidrag

1. Lage møtereferater for hvert møte slik at vi har oversikt over hvordan vi kommer frem til alle beslutninger. Dette er også lurt med tanke på at alle har en oversikt over hva som ble diskutert og besluttet.
2. Alle gjør de oppgavene de har fått utdelt og gir beskjed dersom man trenger hjelp eller står fast.
3. Notere ned alle oppgaver som skal gjøres i en log.

#### Hva som skjer ved avvik eller uenigheter

1. Dersom det skjer en konflikt eller uenighet skal man ta det først opp i gruppen og forsøke å løse det på en demokratisk måte ved avstemning. Hvis det fortsatt ikke løser seg, skal gruppen ta kontakt med veileder.

## 9.2 Samtykkeerklæring

Vi er studenter i emnet *IN2000 – Software Engineering med prosjektarbeid* ved Institutt for informatikk ved Universitetet i Oslo. Med dette skrivet ønsker vi å informere hva prosjektet vårt har som formål, spørre deg om du vil delta i prosjektet, samt berette hva deltagelse vil innebære for deg

### **Formål**

Formålet med vårt prosjekt er å forstå svaksyntes behov, slik at vi kan bedre tilrettelegge appen vår. I forbindelse med dette ønsker vi å utføre brukerundersøkelser.

### **Deltakelse**

Du blir spurta om å delta fordi du faller innenfor vår målgruppe, definert som svaksynte. Dersom du velger å delta ønsker vi å intervju deg for vår datainnsamling. Intervjuet vil vare i 10 minutter, og vi kommer til å notere underveis.

### **Frivillig deltagelse**

Det er frivillig å delta. Du kan når som helst avslutte eller trekke tilbake informasjon som er gitt. Du kan når som helst velge å trekke samtykket uten å måtte oppgi grunn. Dersom samtykket trekkes vil eventuelle personopplysninger som er innsamlet om deg slettes og det vil ikke innebære noen negative konsekvenser for deg at du velger å trekke ditt samtykke.

### **Personvern: innsamling, oppbevaring, behandling og bruk av dine opplysninger**

Ingen sensitive personopplysninger (jf. Personvernforordningens artikkel 9 og 10) vil bli innsamlet. Personlige opplysninger om deg vil kun benyttes til formålene beskrevet i dette informasjonsskrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Personlige opplysninger innsamlet vil bli anonymisert i rapporteringen. Ingen andre enn vi vil ha tilgang til dataen, og vi vil følge Universitetet i Oslo sine rutiner for sikker oppbevaring.

Data vi samler inn kan ettersendes deg ved ønske. Dataen som oppbevares, inkludert anonymisert data, vil ikke bli publisert og vil heller ikke kunne tilbakeføres til deg.

### ***Hva skjer med innsamlet data når studentprosjektet avsluttes?***

Alle notater fra intervjuet blir slettet senest 20.06.2023. Dette gjelder også anonymiserte og avidentifiserte opplysninger om deg.

### ***Rettigheter***

Vi behandler opplysninger om deg basert på ditt samtykke. Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hvis du har spørsmål til undersøkelsen, eller ønsker å benytte deg av dine rettigheter, ta kontakt med Maha eller vår universitetsansatte fagkontakt Yngve Lindsjørn på e-post [ynglin@ifi.uio.no](mailto:ynglin@ifi.uio.no).

Før intervjuet begynner, ber vi deg om å samtykke i deltagelsen ved å undertegne på at du har lest og forstått informasjonen på dette arket, og ønsker å stille opp til lydintervju.

Med vennlig hilsen

Maha Nisha Chaudhry, Thomas Aleksander Flier, Simon Tidemann Halvorsen, Miriam Schumacher Hillesund, Ayesha Suman Ishaq og Dion Kumnova

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om brukerundersøkelsen, og har fått anledning til å stille spørsmål. Jeg samtykker til:

---

Sted og dato

---

Fullt navn

---

Signatur

### 9.3 Intervjuguide

#### Første brukerundersøkelse

1. Hva er din synsstyrke?
2. Hva slags teknologi bruker du i din hverdag?
3. Har din synshemning påvirkning på din bruk av teknologi?
4. Hva synes du om designet?
5. Hva mener du mangler og kan forbedres i dette designet?
6. Hvis du skulle ha brukt en app for å se værmeldingen, har du noen ideer om hvordan den ideelt ville sett ut?

#### Pilotundersøkelse

1. Var tekststørrelsen, fargekontrasten og fargevalget tilstrekkelig for deg?
2. Var vær-symbolene og beskrivelsene klare og forståelige?
3. Var det noen aspekter av appen som var forvirrende eller vanskelig å forstå?
4. Hva kunne ha gjort din opplevelse med å bruke appen bedre? Hvilken funksjonalitet savner du i appen?
5. Hva syntes du om den overordnede brukervennligheten til appen?

#### Andre brukerundersøkelse

1. Var tekststørrelsen, fargekontrasten og fargevalget tilstrekkelig for deg?

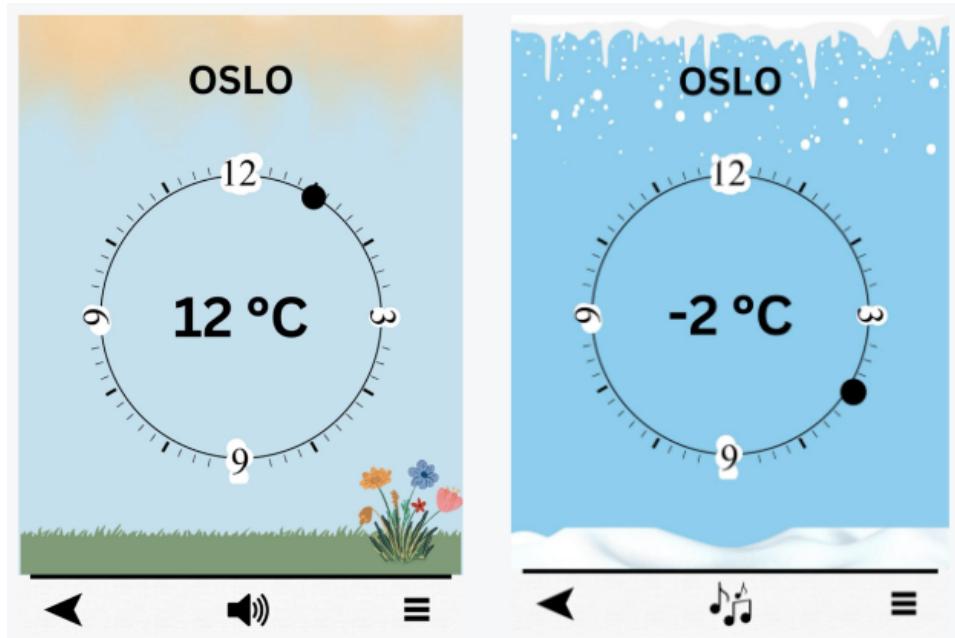
2. Var vær-symbolene og beskrivelsene klare og forståelige?
3. Var det noen aspekter av appen som var forvirrende eller vanskelig å forstå?
4. Hva kunne ha gjort din opplevelse med å bruke appen bedre? Hvilken funksjonalitet savner du i appen?
5. Fikk du tilstrekkelig informasjon om værforholdene som var relevant for deg?
6. Er appens navigasjonsstruktur enkel å forstå?
7. Er brukergrensesnittet ryddig, estetisk tiltalende og i tråd med moderne designprinsipper?

## 9.4 Prototyper

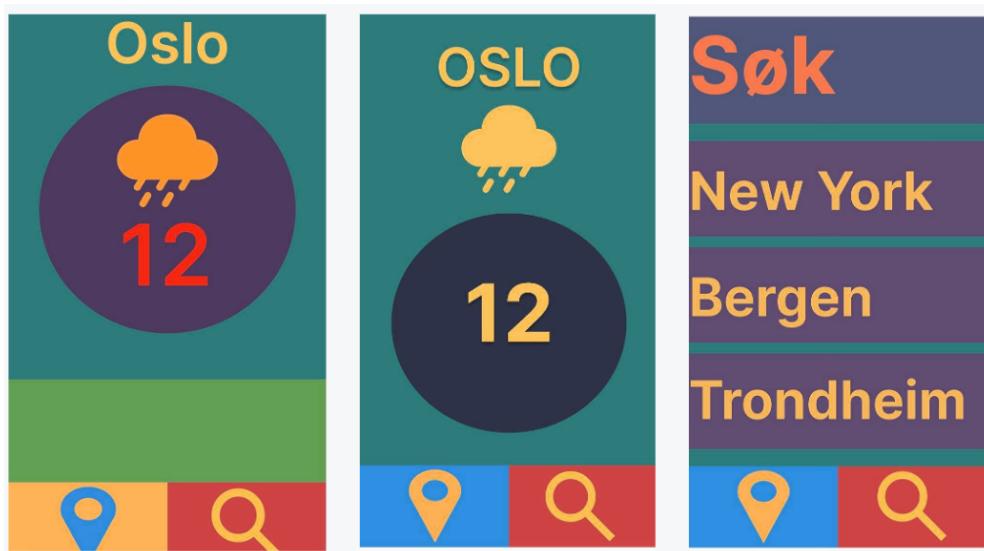
Prototype 1



Prototype 2



Prototype 3



Prototype 4



## 9.5 Fullverdig kravspesifikasjon

### 9.5.1 Brukerhistorier

1	Som bruker av appen ønsker jeg å se værmeldingen for å være informert om værforholdene på min nåværende posisjon.
2	Som en fargeblind bruker av appen ønsker jeg å tilpasse fargene til min fargeblindhet for å kunne lese vær-informasjonen.
3	Som bruker av appen ønsker jeg å se værmeldingen til et annet sted for å være informert om værforholdene der.
4	Som bruker av appen ønsker jeg å kunne få været til min posisjon opplest, slik at jeg kan bli informert om værforholdene uten å anstrengte synet mitt
5	Som bruker av appen ønsker jeg at elementer i appen allerede er tilpasset og forstørret i samsvar med min svaksynhet, slik at jeg tydelig kan lese og tolke informasjon, uten behov for zooming.
6	Som bruker av appen, ønsker jeg å få presentert værinformasjon, på en tydelig og strukturert måte, slik at jeg raskt kan få med meg værdataene.
7	Som bruker av appen, ønsker jeg tydelig skille mellom ulike elementer ved bruk av en høykontrast fargekombinasjon, slik at det er enkelt å lese og finne det jeg trenger.
8	Som bruker av appen ønsker jeg et minimalistisk design, som unngår overveldende

1	Som bruker av appen ønsker jeg å se værmeldingen for å være informert om værforholdene på min nåværende posisjon.
2	Som en fargeblind bruker av appen ønsker jeg å tilpasse fargene til min fargeblindhet for å kunne lese vær-informasjonen.
3	Som bruker av appen ønsker jeg å se værmeldingen til et annet sted for å være informert om værforholdene der.
4	Som bruker av appen ønsker jeg å kunne få været til min posisjon opplest, slik at jeg kan bli informert om værforholdene uten å anstrenges synet mitt
	informasjon, slik at jeg slipper å anstrenges synet mitt.
8	Som bruker av appen, ønsker jeg at været skal visualiseres, slik at jeg raskt kan få med meg værforholdene.
10	Som bruker av appen, ønsker jeg at appen har en intuitiv og konsistent oppbygging, slik at jeg kan forutsi hvor og hvordan ulike funksjoner og navigasjonselementer befinner seg og fungerer.
11	Som bruker av appen, ønsker jeg å se været frem i tid, slik at jeg kan planlegge dagene mine i henhold til været.
12	Som bruker av appen, ønsker jeg at søkerne mine lagres, slik at jeg slipper å søke opp stedet på nytt hver gang.
13	Som bruker av appen, ønsker jeg at appens brukergrensesnitt er i tråd med moderne design trender og standarder, slik at jeg føler meg oppdatert og tilfredsstilt av en moderne opplevelse.
14	Som bruker ønsker jeg å enkelt, kunne navigere meg frem til min nåværende posisjon, for å se værforholdene der.

### 9.5.2 Funksjonelle krav som ble implementert

Krav	Opphav	Prioritet
Systemet skal på forsiden oppgi temperaturen til brukerens nåværende posisjon.	Gransking av andre værappper og brukerundersøkelse 1	1
Systemet skal på forsiden tilby oversikt over temperaturens utvikling over 24 timer	Gransking av andre værappper, brukerundersøkelse 2	2
Systemet skal på forsiden tilby oversikt over temperaturens	Gransking av andre værappper, brukerundersøkelse 1	1

utvikling over 7 dager, med informasjon om laveste og høyeste temperatur for hver dag.		
Appen skal på forsiden oppgi riktig vindretning og hastighet	Gransking av andre væranner og brukerundersøkelse 2	2
Appen skal gi brukeren muligheten til å søke opp ulike posisjoner, slik at disse vises fremfor deres nåværende posisjon.	Gransking av andre væranner og brukerundersøkelse 1	1
Appen skal gi brukeren ulike alternativer underveis i søkerprosessen.	Gransking av andre væranner og brukerundersøkelse 3	1
Appen skal ved stavelses feil fra brukeren fremdeles greie å finne frem til det mest sannsynlige alternativet.	Gransking av andre væranner og brukerundersøkelse 2	2
Appen skal tilby et talestyringsalternativ som leser opp temperatur og nedbør	Case beskrivelse og brukerundersøkelse 1	1
Appen skal inneholde en hjem knapp som får appen til å vise været til brukerens nåværende posisjon	Gransking av andre væranner og brukerundersøkelse 1	1
Appens skal tilby en liste med tidligere søk, som brukeren kan trykke på for å få opp værdata igjen.	Brukerundersøkelse 2	3
Appen skal tilby en navigasjonsbar på alle skjermer	WCAG Kriterium 3.2.3 (konsekvent navigering)	2
Appen skal fremstille værutviklingen i tabeller som ikke endrer formatering basert på valgt tidsrom	Gransking av andre apper og WCAG Kriterium 3.2.1	1
Appen skal ha et "søk"-ikon som brukeren kan trykke på, etter å ha oppgitt ønsket sted.	Gransking av andre apper	3
Appen skal navigere brukeren til en oppdatert hjemskjerm når et alternativ fra søker-menyen blir	Egne vurderinger på applikasjonens utforming	3

valgt.		
Appen skal gi brukeren valget om å slette tidligere søk, ved trykk på symbol	Gransking av andre vær-apper	3
Appen skal gi brukeren muligheten til å bytte mellom 24-timer værvarsel og 7-dagers værvarsel.	Brukerundersøkelse 2 og gransking av andre værappeller	1
Været skal visualiseres ved bruk av vær-symboler	Gransking av andre apper	2
Appen skal ikke lagre duplikater av samme søk i listen over tidligere søk.	Egne vurderinger på applikasjonens utforming	3
Appen skal vise en error skjerm dersom det er fravær av internett tilkobling	Egne vurderinger på applikasjonens utforming	2
Appen skal legge det nyeste søker øverst i listen av tidligere søker	Gransking av andre vær-apper	3
Appen skal vise værdata til steder som ikke er inkludert i listen med alternative steder.	Egne vurderinger på applikasjonens utforming	3
Appen skal ikke kræsje ved søker etter steder som ikke har værdata.	Egne vurderinger på applikasjonens utforming	1

### 9.5.3 Funksjonelle krav som ikke ble implementert

Krav	Opphav	Prioritet	Begrunnelse for mangel på implementasjon
Appen bør gi muligheten til å lagre lokasjoner til en egen "favoritt side"	Gransking av andre værappeller og brukerundersøkelser	2	Mangel på tid og ressurser, hadde ført til komplikasjoner som vi mente ikke var verdt det.
Appen skal gi fargealternativer for de tre vanligste formene for	Egne vurderinger på applikasjonens utforming	2	Vi innså at det var enklere å velge et fargepalett som allerede var tilpasset alle former for fargeblindhet.

fargeblindhet.			
Appen skal tilby alternativer for tekststørrelse	WCAG kriteriet 1.4.4	1	Dette endte vi opp med vente for lenge med. Vi var ikke klar over vanskeligheten av det, ble derfor utelatt.
Analog klokke	Egne vurderinger på applikasjonens utforming	1	Dette var et konsept som etter undersøkelse 1 ble fjernet, siden vi identifiserte en mangel på interesse for en slik funksjon, og innså at det ville gå i strid med kravet om enkelhet.
Været skal visualiseres ved bruk av animasjon	Egne vurderinger på applikasjonens utforming	3	Basert på tilbakemeldinger fra den første brukerundersøkelsen.
Appen skal vise en snack-bar dersom brukeren forsøker å legge inn ugyldige søk, slik som tomme søker, emoji og heltall	Gransking av andre apper	3	Mangel på tid. Vi mente at det var viktigere å prioritere andre ting fremfor dette.
Appen skal vise en snackbar dersom brukeren søker etter værdata for områder eller steder som ikke er støttet eller tilgjengelig i API	Gransking av andre apper	3	Mangel på tid. Vi mente at det var viktigere å prioritere andre ting fremfor dette.

#### 9.5.4 Ikke-funksjonelle krav

Krav	Prioritet	Type
Appen skal struktureres etter MVVM	1	Produktkrav
Appen skal kunne kjøre på API-level 30 til 33	1	Produktkrav
Appen skal hente værdata fra Location Forecast API	1	Produktkrav
Appen skal hente data	1	Produktkrav

asynkront		
Appen skal følge WCAG kravet om konsekvent navigering(Kriterium 3.2.3 )	1	Produktkrav
Appen må være tilpasset flere skjermstørrelser og enheter	1	Produktkrav
Appen skal være utformet i henhold til WCAG krav om høykontrast	1	Produktkrav
Appen skal ikke kreve unødvendige eller overflødige brukerinteraksjoner for å vise værdata.	1	Produktkrav
Systemets kodehåndtering skal gjøres gjennom github	2	Organisatorisk krav
Teamet skal ta til bruk en scrumban modell	2	Organisatorisk krav
Teamet skal sørge for at relevant dokumentasjon er tilgjengelig og oppdatert	1	Organisatorisk krav
Teamet skal kommunisere gjennom messenger og discord	3	Organisatorisk krav
Appen skal etter beste evne følge WCAG-retningslinjer.	1	Eksternt krav
Appen skal ikke krenke brukerens personvern eller samle inn data uten deres samtykke.	1	Eksternt krav
Appen skal være i samsvar med personvernloven	1	Eksternt krav

## 9.6 Oppfylte kriterier i WCAG

Her er en oversikt over WCAG prinsippene, og tilhørende krav vi har fulgt:

### *Prinsipp 1: Mulig å oppfatte*

1.1.1

1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8

1.3.2, 1.3.4, 1.3.5, 1.3.6,

1.4.1, 1.4.3, 1.4.6, 1.4.7, 1.4.8, 1.4.9, 1.4.10, 1.4.11, 1.4.12, 1.4.13

### *Prinsipp 2: Mulig å betjene*

2.1.1, 2.1.2, 2.1.3

2.2.1, 2.2.5, 2.2.6

2.3.1, 2.3.2, 2.3.3

2.4.1, 2.4.2, 2.4.3, 2.4.4, 2.4.5, 2.4.6, 2.4.7, 2.4.8, 2.4.9, 2.4.10

2.5.1, 2.5.2, 2.5.4

### *Prinsipp 3: Forståelig*

3.1.1, 3.1.2, 3.1.3, 3.1.4, 3.1.5, 3.1.6

3.2.2, 3.2.3, 3.4.4, 3.2.5

3.3.2, 3.3.6

## 10. Litteraturliste

*Adobe Color Contrast Analyzer.* (n.d.). Adobe Color. Retrieved May 23, 2023, from

<https://color.adobe.com/nb/create/color-contrast-analyzer>

DeLaGrana, A. (2023, February 9). *Jetpack Compose vs XML: Android UI Development Compared - Moove It.* Blog. Retrieved May 24, 2023, from

<https://blog.moove-it.com/jetpack-compose-vs-xml-android-ui/>

Entur. (n.d.). *Brukerhistorier.* Entur Designsystem. Retrieved May 25, 2023, from

<https://design.entur.no/kom-i-gang/for-designere/bruikerhistorier>

Florea, R. (n.d.). *ETSF01 - Lecture 1*. UiO. Retrieved May 19, 2023, from

<https://www.uio.no/studier/emner/matnat/ifi/INF3121/v15/lecture-02.pdf>

*Geocoding API overview*. (2023, 05 22). Google for Developers. Retrieved May 23, 2023,

from <https://developers.google.com/maps/documentation/geocoding/overview>

*Geolocation API overview*. (2023, 05 18). Google for Developers. Retrieved May 23, 2023,

from <https://developers.google.com/maps/documentation/geolocation/overview>

Gillis, A. S. (n.d.). *What Is Static Testing?* TechTarget. Retrieved May 26, 2023, from

<https://www.techtarget.com/searchsoftwarequality/definition/static-testing>

*Guide to app architecture*. (2023, March 28). Android Developers. Retrieved May 23, 2023,

from <https://developer.android.com/topic/architecture>

Holtebekk, T. (2021, 06 27). *ISO – Store norske leksikon*. Store norske leksikon. Retrieved

May 24, 2023, from <https://snl.no/ISO>

Inteoptik. (n.d.). *Synsfeil Dette bør du vite om langsnyhet*. Inteoptik. Retrieved May 25,

2023, from <https://www.interoptik.no/oyehelse/synsfeil/langsynhet/>

ISO. (n.d.). *ISO 25010*. iso25000.com. Retrieved May 26, 2023, from

<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

Koyuncular, B. (2021, October 25). *The Population of Blind People in the World!* BlindLook.

Retrieved May 19, 2023, from

<https://www.blindlook.com/blog/detail/the-population-of-blind-people-in-the-world>

*Locationforecast*. (n.d.). the MET Norway Weather API. Retrieved May 24, 2023, from

<https://api.met.no/weatherapi/locationforecast/2.0/documentation>

*Logcat command-line tool | Android Studio*. (2023, April 12). Android Developers. Retrieved

May 26, 2023, from <https://developer.android.com/tools/logcat>

Martin, M. (2023, April 1). *MVC vs MVVM – Difference Between Them*. Guru99. Retrieved

May 25, 2023, from <https://www.guru99.com/mvc-vs-mvvm.html>

Montiel, I. (2018, September 17). *Low Coupling, High Cohesion. The key to creating maintainable code... | by Ivan Montiel | clarityhub*. Medium. Retrieved May 25, 2023, from  
<https://medium.com/clarityhub/low-coupling-high-cohesion-3610e35ac4a6>

NHI. (2018, September 13). *Langsynthet*. NHI.no. Retrieved May 25, 2023, from  
<https://nhi.no/sykdommer/barn/oye/langsynthet/?page=3>

NHI. (2023, January 20). *Svaksynthet og blindhet*. NHI.no. Retrieved May 19, 2023, from  
<https://nhi.no/sykdommer/oye/diverse/svaksynthet-og-blindhet/>

*Request location permissions*. (n.d.). Android Developers. Retrieved May 26, 2023, from  
<https://developer.android.com/training/location/permissions>

*Retningslinjer for universell utforming av nettinnhold (WCAG) 2.1*. (2018, June 5). W3C. Retrieved May 24, 2023, from  
<https://www.w3.org/Translations/WCAG21-no/#contrast-minimum>

*Running a Usability Test*. (n.d.). Usability.gov. Retrieved May 26, 2023, from  
<https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>

Sandvig, K., & Høvding, G. (2020, 2 3). *svaksynthet – Store medisinske leksikon*. Store medisinske leksikon. Retrieved May 25, 2023, from <https://sml.snl.no/svaksynthet>

Sandvig, K., & Høvding, G. (2020, 7 20). *fargeblindhet – Store medisinske leksikon*. Store medisinske leksikon. Retrieved May 25, 2023, from <https://sml.snl.no/fargeblindhet>

Sommerville, I. (2019). *Engineering Software Products: An Introduction to Modern Software Engineering*. Pearson.

*Target API level requirements for Google Play apps - Play Console Help*. (n.d.). Google Support. Retrieved May 26, 2023, from  
<https://support.google.com/googleplay/android-developer/answer/11926878>

Tilsynet for universell utforming. (n.d.). *Brukertesting med funksjonshemmede*. Tilsynet for universell utforming av ikt. Retrieved May 26, 2023, from

<https://www.uutilsynet.no/regelverk/brukertesting-med-funksjonshemmede/193>

Tilsynet for universell utforming. (n.d.). *Kvifor universell utforming av ikt?* Tilsynet for universell utforming av ikt. Retrieved May 26, 2023, from

<https://www.uutilsynet.no/veiledning/kvifor-universell-utforming-av-ikt/240>

Tilsynet for universell utforming. (n.d.). *Universell utforming i utviklingsprosessen*. Tilsynet for universell utforming av ikt. Retrieved May 26, 2023, from

<https://www.uutilsynet.no/regelverk/universell-utforming-i-utviklingsprosessen/201>

Vatne, A. A. (2022, 05 21). *Repetisjon - klassediagram og sekvensdiagram*. UiO. Retrieved May 26, 2023, from

<https://www.uio.no/studier/emner/matnat/ifi/IN1030/v22/repetisjon/Lysark%20med%20repetisjon%20/repetisjon---klassediagram-og-sekvensdiagram.pdf>

Vihovde, E. H. (2023, 02 20). ". ." - Wiktionary. Retrieved May 26, 2023, from

<https://www.uio.no/studier/emner/matnat/ifi/IN2000/v23/forelesningsvideoer/ynglin-IN2000-20230220-141552.mp4?vrtx=view-as-webpage>

Visuresolutions. (n.d.). *Funksjonelle vs ikke-funksjonelle krav - Visure-løsninger*. Visure Solutions. Retrieved May 26, 2023, from

<https://visuresolutions.com/no/requirements-management-traceability-guide/functional-vs-non-functional-requirements/>

Visuresolutions. (n.d.). *Krav Definisjon: Hva er det og hvordan bruke det? | Komplett veiledning*. Visure Solutions. Retrieved May 19, 2023, from

<https://visuresolutions.com/no/blog/requirements-definition/>

World Wide Web Consortium. (2018, Desember 11). *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*. W3C. Retrieved May 25, 2023, from <https://w3c.github.io/Mobile-A11y-TF-Note/#touchscreen-gestures>