

Erlend Sundøy  
Snorre Strand  
Simen Tengs

## Utvikling av mobilapplikasjonen Televindu

En plattform for sakte-TV tilpasset eldre og personer med kognitiv svikt

**Mai 2020**

### **NTNU**

Norges teknisk-naturvitenskapelige universitet.  
Fakultet for informasjonsteknologi og elektroteknikk  
Institutt for datateknologi og informatikk



## Forord

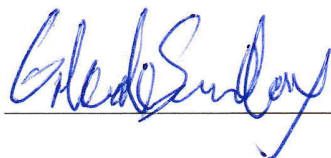
Alle i prosjektgruppen begynte å lære seg utvikling for Android i semesteret før dette prosjektet, og har blitt interessert i mobilutvikling. Ved å velge denne oppgaven fikk vi mulighet til å bedre vår kunnskap innen dette feltet. Oppgaven presenterte muligheten til å fritt velge teknologi selv, slik at vi kunne lære oss nye og moderne teknologier.

Vi fikk også mulighet til å løse et relevant problem i dagens samfunn hvor det ikke finnes mange teknologiske hjelpemidler fra før. Dette var spennende fordi vi kunne fordype oss i en problemstilling som vi anser som viktig, og som førte til mye ny kunnskap utover det ingeniørfaglige arbeidet.

Vi vil gjerne takke Øystein Hansen og resten av Televindu for å ha gitt oss muligheten til å gjennomføre dette prosjektet, og for å ha hjulpet oss med tilgang til eksisterende forskning og testing av systemet. Vi vil også takke vår veileder Alexander Holt for å ha gitt oss veiledning gjennom prosjektet.

Norges teknisk-naturvitenskaplige universitet

Trondheim, mai 2020



Erlend Sundøy



Snorre K. Strand



Simen S. Tengs

## Oppgavetekst

Den opprinnelige oppgaveteksten fra oppdragsgiver besto av å utvikle Flutter applikasjonen Televindu, en videoplattform for sakte-tv som var i tidlig utvikling. Applikasjonen er rettet mot personer med demens og andre former for kognitiv svikt. Et av fokusene var at vi skulle utvikle en livefunksjon, men av finansielle grunner gikk oppdragsgiver tidlig vekk fra dette. Oppdragsgiver bestemte dermed at vi skulle flytte fokuset til andre funksjonaliteter som bidrar til å gjøre applikasjonen mer personlig fokusert mot enkeltbrukere. Dette inkluderer blant annet å lage støtte for spillelister, videoer i nærheten av brukeren og anbefalte videoer.

Visjonsdokument finnes i vedlegg *A*, og kravdokument med kravene til systemet finnes i vedlegg *B*.

## Sammendrag

I denne oppgaven har vi tatt for oss et økende problem hos en sårbar demografi som ofte blir oversett. Eldre mennesker på helseinstitusjoner sliter med å gjøre noe sosialt, bli underholdt og holde kroppen i gang. Denne demografien blir bare større og større, med eldrebølgen på vei. I løpet av prosjektet har vi utviklet Televindu; en cross-platform mobilapplikasjon med hovedmål å øke livskvaliteten på sykehjem og andre lignende helseinstitusjoner som huser eldre personer med demens og andre kognitive problemer. Vi tar for oss hvordan sansestimuli og reminisensterapi påvirker livskvalitet, og hvordan vi kan utvikle et system som utnytter effekten av dette.

Vi har utviklet alle funksjonene som ble definert i den oppdaterte oppgaveteksten. Noen funksjoner måtte endres på grunn av omprioriteringer eller at teknologien ikke var kompatibel med systemet. Gjennomgående i prosjektet har vi fått nye krav fra oppdragsgiver som vi har tatt med i betraktning med tanke på tidsbruk og sprintplanlegging. Med noen omprioriteringer, har også alle disse funksjonene blitt implementert.

Det er gjort mange studier på både sansestimuli og reminisensterapi i ulike former. Av ulike grunner kan det likevel være vanskelig å anvende denne teorien i praksis av årsaker som for eksempel mangel på ressurser og terapeuter. Det er heller ikke noen velferdsteknologier på markedet som fokuserer på sansestimuli i den grad Televindu gjør. Televindu vil dermed tilby en ny og lett måte å få tilgang til sansestimuli og reminisensterapi.

# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>1</b>
1.1	Behov, Effektmål og Problemstilling . . . . .	1
1.1.1	Behov . . . . .	1
1.1.2	Effektmål . . . . .	1
1.1.3	Problemstilling . . . . .	2
1.2	Televindu plattformens tilstand ved prosjektstart . . . . .	2
<b>2</b>	<b>Teori</b>	<b>3</b>
2.1	Tidligere studier og bakgrunnsinformasjon . . . . .	3
2.1.1	Kognitiv svikt . . . . .	3
2.1.2	Sansestimuli hos demensrammede . . . . .	3
2.1.2.1	Over- og understimulering . . . . .	4
2.1.3	Tilpasset sansestimuli . . . . .	4
2.1.3.1	Eksisterende tiltak . . . . .	5
2.1.3.2	Bruk av velferdsteknologier . . . . .	5
2.1.3.3	Reminisensterapi . . . . .	7
2.2	Teknologi . . . . .	8
2.2.1	Utvikling på tvers av plattformer . . . . .	8
2.2.1.1	WebView Platform . . . . .	8
2.2.1.2	Bridge Platform . . . . .	9
2.2.2	MVVM arkitektur . . . . .	9
2.2.3	NoSQL database . . . . .	10
2.2.3.1	Dokument orientert database arkitektur . . . . .	11
2.2.4	Universell utforming mht. eldre personer . . . . .	11
<b>3</b>	<b>Valg av teknologi og metode</b>	<b>13</b>
3.1	Fremgangsmåte . . . . .	13
3.1.1	Fremgangsmåte for forskning . . . . .	13
3.1.2	Utviklingsprosess . . . . .	13
3.1.2.1	Arbeids- og rollefordeling . . . . .	13
3.1.2.2	Scrum . . . . .	14
3.2	Valg av teknologi . . . . .	14
3.2.1	Firestore . . . . .	14
3.2.2	Flutter . . . . .	16
3.2.2.1	Provider (State management) . . . . .	18
3.2.3	Fastlane . . . . .	18
3.2.4	Github Actions . . . . .	18
<b>4</b>	<b>Resultater</b>	<b>20</b>
4.1	Vitenskapelige resultater . . . . .	20
4.1.1	Begrensninger . . . . .	20
4.1.2	Produkt og Design . . . . .	20
4.2	Ingeniørfaglige resultater . . . . .	21
4.2.1	Produkttegenskaper og status på mål fra visjonsdokumentet . . . . .	21
4.2.1.1	ViewModel kode . . . . .	24
4.2.1.2	Platform channel . . . . .	25

---

4.2.2	System som helhet . . . . .	26
4.2.2.1	Design . . . . .	26
4.2.2.2	Navigasjon . . . . .	27
4.3	Administrative resultater . . . . .	28
4.4	Utviklingsprosessen . . . . .	28
<b>5</b>	<b>Diskusjon</b>	<b>29</b>
5.1	Vitenskapelig diskusjon . . . . .	29
5.1.1	Televindu som sansestimuli og reminisensterapi . . . . .	29
5.2	Forbedringspotensialer . . . . .	31
5.3	Helhetlig systemperspektiv . . . . .	32
5.3.1	Etiske problemstillinger . . . . .	34
5.4	Gruppearbeid refleksjon . . . . .	34
5.5	Diskusjon av Ingeniørfaglige og Administrative resultater . . . . .	35
5.5.1	Generelle valg av prosjektstruktur . . . . .	36
5.5.1.1	State management . . . . .	36
5.5.1.2	Unit testing . . . . .	37
5.5.1.3	View . . . . .	37
<b>6</b>	<b>Konklusjon</b>	<b>39</b>
	<b>Referanser</b>	<b>40</b>
<b>A</b>	<b>Visjonsdokument</b>	<b>41</b>
<b>B</b>	<b>Kravdokumentasjon</b>	<b>54</b>
<b>C</b>	<b>Scrum</b>	<b>69</b>
<b>D</b>	<b>Prosjekthåndbok</b>	<b>78</b>

# Akronymer og begreper

## Akronymer

- **ML-kit:** Machine Learning Kit
- **OEM:** Original Equipment Manufacturer
- **MVC:** Model-View-Controller
- **MVP:** Model-View-Presenter
- **MVVM:** Model-View-ViewModel
- **JIT:** Just-In-Time (Compilation)
- **AOT:** Ahead-Of-Time (Compilation)

## Begreper

- **Material design:** Et tilpasningsdyktig system med retningslinjer, komponenter og verktøy som støtter den beste fremgangsmåten for design av brukergrensesnitt.
- **WebView:** Et systemkomponent som gjør at apper kan vise nettinhold.
- **Widget:** Element brukt for å strukturere og oppbygge brukergrensesnittet.
- **Bridge Platforms:** bro som brukes for tverrplattformutvikling som kommuniserer med native tjenester, OEM widgets og programmeringsspråket man koder i.
- **Hot Reloading:** Oppdaterer filer som har blir endret under utvikling uten å miste appens tilstand.
- **Pub Package:** Pub er pakkebehandleren for programmeringsspråket Dart, som inneholder gjenbrukbare biblioteker og pakker for Flutter, Angular Dart og generelle Dart-programmer.
- **Carousel:** En slider for visning av en rekke elementer.
- **Tickets/Issues:** Oppdeling av user-stories brukt i Scrum tavler.
- **Trello:** Teknologisk samhandlingsverktøy for smidige utviklingsmetoder.



## Figurliste

2.1	Webview arkitektur . . . . .	8
2.2	Bridge arkitektur . . . . .	9
2.3	MVVM arkitektur . . . . .	10
2.4	Dokument orientert database arkitektur . . . . .	11
3.1	Flutter arkitektur . . . . .	17
4.1	Hjem-skjerm . . . . .	21
4.2	Onboarding . . . . .	22
4.3	Video-skjerm . . . . .	26
5.1	VPC Pleiesenter . . . . .	33
5.2	VPC Beboer . . . . .	33
5.3	Service-providers i applikasjonen . . . . .	37

## Forord

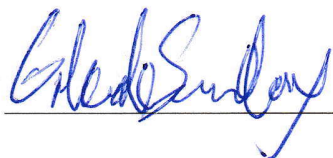
Alle i prosjektgruppen begynte å lære seg utvikling for Android i semesteret før dette prosjektet, og har blitt interessert i mobilutvikling. Ved å velge denne oppgaven fikk vi mulighet til å bedre vår kunnskap innen dette feltet. Oppgaven presenterte muligheten til å fritt velge teknologi selv, slik at vi kunne lære oss nye og moderne teknologier.

Vi fikk også mulighet til å løse et relevant problem i dagens samfunn hvor det ikke finnes mange teknologiske hjelpemidler fra før. Dette var spennende fordi vi kunne fordype oss i en problemstilling som vi anser som viktig, og som førte til mye ny kunnskap utover det ingeniørfaglige arbeidet.

Vi vil gjerne takke Øystein Hansen og resten av Televindu for å ha gitt oss muligheten til å gjennomføre dette prosjektet, og for å ha hjulpet oss med tilgang til eksisterende forskning og testing av systemet. Vi vil også takke vår veileder Alexander Holt for å ha gitt oss veiledning gjennom prosjektet.

Norges teknisk-naturvitenskaplige universitet

Trondheim, mai 2020



Erlend Sundøy



Snorre K. Strand



Simen S. Tengs

# 1 Introduksjon

## 1.1 Behov, Effektmål og Problemstilling

For mer utfyllende informasjon om visjonen for prosjektet og mål, se visjonsdokumentet i vedlegg A og kravdokumentet i vedlegg B.

### 1.1.1 Behov

Televindu er en videoplattform for sakte-TV hvor målgruppen personer med demens og/eller annen kognitiv svikt.

*“Televindu er tilpasset for personer med kognitiv svikt da teknologien bruker en form for audiovisuell stimulering som ikke avhenger av konsentrasjonsevnen og korttidshukommelsen. Formålet med Televindu er å muliggjøre målrettet audiovisuell stimulering til den enkelte basert på personens atferdsmessige og psykologiske symptomer og livshistorie.”*

<https://televindu.com/>

Siden vi skulle videreutvikle Televindu sin plattform, var det deres visjon og langsiktige mål for virksomheten som underbygge våre mål i prosjektet. Plattformen brukes i dag på flere eldre- og sykehjem, blant annet i Oslo og Danmark. Det er mange utfordringer som følger med slike pasientgrupper. Personer med demens sliter ofte med utagerende adferd, blant annet på grunn av forvirring eller at de over- eller understimulert. Effektmålene og problemstilling vår baserer seg på disse utfordringene. Et av våre mål med prosjektet er å utvikle plattformen og tilby diverse nye funksjoner som skal være med på å dekke disse behovene på best mulig måte.

### 1.1.2 Effektmål

Televindu sine effektmål omhandler først og fremst tilpasset audiovisuell stimulering for brukerne.

*“Både under- og overstimulering av sansene kan lede til endret atferd og sykdom som kompensasjon for sansemessige utfordringer for personer med kognitiv svikt. Det kan vise seg i form av apatisk atferd, fysisk uro og angst, plutselige sinneutbrudd, og generell*

*mistrivsel.” <https://televindu.com/>*

Vi skal i denne rapporten drøfte tilpasset sansestimuli for demenspasienter. Vi skulle videreutvikle plattformen og har utviklet nye funksjoner som skal forbedre brukeropplevelsen og tilpasse applikasjonen til enkeltpersoner.

### 1.1.3 Problemstilling

I tillegg til å videreutvikle applikasjonen skal vi forske videre på om tilpasset sansestimuli har en positiv effekt på livskvalitet hos personer rammet av demens. Vi skal bruke denne kunnskapen og utforske i hvilken grad tilpasset sansestimuli kan øke demenspasienters livskvalitet gjennom Televindu sin plattform. Fokuset vårt i systemutviklingsdelen av prosjektet vil ligge i å utvikle plattformen på en måte som samsvarer med det vi kommer frem til i forskningendelen. Vi vil til slutt diskutere styrker og svakheter med løsningen vår.

*“I hvilken grad kan vi gjennom utviklingen av en plattform for sakte-TV fasilitere sansestimuli og dermed bedre trivselen for personer med kognitiv svikt på helseinstitusjoner”*

## 1.2 Televindu plattformens tilstand ved prosjektstart

Det var allerede en minimal outsourced iOS versjon av Televindu appen da vi begynte på prosjektet. Den brukte ikke Firebase, og fikk videoene direkte fra Vimeo.

Før prosjektet offisielt startet hadde vi allerede begynt på Flutter applikasjonen med alle hovedskjermene, støtte for flere språk, videospillere og kommunikasjon med Firestore. Applikasjonen var fortsatt i et veldig tidlig stadie med mange bugs og kun noe av bare bonesfunksjonalitetene da vi hadde offisiell prosjektoppstart i Februar.

## 2 Teori

I dette kapittelet skal vi skrive om teorigrunnet for vår oppgave. Vi har valgt å dele kapittelet inn i to hoveddeler. Den første delen omhandler teori som er relevant for brukergruppen til platformen. I den andre delen skal vi ta for oss teknologispesifikk teori som kan være ukjent for leseren.

### 2.1 Tidligere studier og bakgrunnsinformasjon

#### 2.1.1 Kognitiv svikt

Eldrebeholdningen er stigende, og på grunn av det ser vi en økende utvikling av sykdommer som demens. Mange funksjoner i hjernen påvirkes, og det fører ofte til at personen som rammes kan ende på sykehjem. I dag er det omkring 50 millioner mennesker som har demens, med nesten 10 millioner nye rammede hvert år. WHO - Dementia (2019). Demens er en fellesbetegnelse på flere hjernesykdommer som medfører kognitiv svikt og andre funksjonstap. De aller fleste demenstilfellene er knyttet til høy alder. FHI (2014) Vi skal her ta for oss behov og utfordringer som følger med demenssykdom, og hva som kan gjøres for å bedre de rammedes livskvalitet.

#### 2.1.2 Sansestimuli hos demensrammede

Personer med kognitiv svikt har mange utfordringer. Noen av disse utfordringene er hukommelse, orientering, språk og forståelse. WHO - Dementia (2019) Å forstå seg selv og omverden kan derfor bli ekstra utfordrende for personer med demenssykdom. Menneskets sanser spiller en sentral rolle når det kommer til å forstå hvor og hvem man er, og evnen til å kunne respondere på en fordelaktig måte i henhold til situasjonen. Bearbeidingen og organiseringen av sansene på denne måten kalles sanseintegrasjon, og handler om å bearbeide sansene på en slik måte at man reagerer på en hensiktsmessig måte på sansestimuli. Sundhedsstyrelsen (2019)

### 2.1.2.1 Over- og understimulering

Det er viktig at personer med demenssykdom får rett mengde sansestimuli. For mye sanseinntrykk kan bli forvirrende og det kan bli vanskelig for personen å foreta normal sanseintegrasjon. Dersom personer med demens får alt for lite inntrykk, kan det føre til endret adferd som kompensasjon for at sansebehovet ikke blir nådd. Det er derfor viktig at det blir gjort en innsats fra personalet på sykehjem for gjøre sansestimuli hos demenspasienter mer tilpasset. I følge sanseintegrasjonsteori kan det å påvirke sansene hos en person med utagerende atferd på en bestemt måte fremme trivsel og trygghet og begrense den gjeldendes opplevelse av stress, uro og angst. Det er mange forskjellige måter å forebygge slik uro, og skape ro og trivsel hos personer med utagerende atferd. Når den rammede viser tegn på slik urolig adferd brukes sansestimulasjon for å unngå forverring eller tap av funksjoner som kommer av under- eller overstimulering. *NFA (2018)*

Metoden av innsats for å skape ro og trivsel hos personen med demens avhenger derfor om han er understimulert og trenger å “vekkes”, eller om han er overstimulert og må beroliges. Det er mange typer sanseunderholdning som er forsøk på å vekke og underholde demensrammede. Man ser dog at vanlig sanseunderholdning kan ofte skape overstimuli hos den demensrammede. Dette skjer fordi personen får for mye inntrykk, i tillegg til at aktivitetene kan oppleves ustrukturerte og tilfeldige. Når man skal bestemme den beste behandlingen til demenspasienter avhenger det blant annet av personens funksjonsnedsettelse, sansemessige utfordringer og livshistorie. *Sundhedsstyrelsen (2019)*

Det er derfor viktig at sansestimuli tilpasses til den enkelte personen.

### 2.1.3 Tilpasset sansestimuli

Dokumentert effekt av tilpasset sansestimuli fra *Sundhedsstyrelsen (2019)*:

- Oppkvikkende og sosialiserende effekt
- Forebygger og reduserer utfordrende atferd
- Reduksjon i maktbruk og beroligende medikamenter
- Bedre arbeidsmiljø og redusert pleiebehov
- Økt appetitt

### 2.1.3.1 Eksisterende tiltak

Musikkterapi er en eksisterende løsning for tilpasset sansestimuli. Musikk kan bidra til å fremme fysisk stimulering eller avslapning. Den største effekten sees når innbyggeren synger, klapper, leser tekstene, nikker og lignende, men også for borgere som er på slutten av demenssykdommen kan ha en beroligende effekt. *Ridder, H.O. (2017), Myskja (2005)* Musikk stimulerer hjernen slik at personer med alvorlig demens kan gjenkjenne og “nyte/forholde seg til” musikk, som kan vekke positive følelser. Det er også mulig å vekke minner gjennom musikkterapeutene aktiviteter, f.eks å synge i felleskap, synge kjente sanger til pasienten eller lytte til gamle sanger. Sangene som har blitt lært i ungdomsårene kan ofte knyttes til tidligere liv.

*Et systematisk review fastslår den gavnlige kliniske effekt ved bruk af audiovisuel stimulering, der ses blandt andet klinisk effekt i form af forbedret kognitive funktioner og reduceret hovedpine hos borgerne. Myskja (2005)*

Sanserom er en form for terapi for personer med kognitiv svikt, hvor man plasserer personer i et beroligende og stimulerende miljø. Disse rommene påvirker personene ved å bruke de ulike sansene, lys, farger, lyd, lukt osv. Sansene kan vekkes, sti mulreses eller berolige. *Snoezel Netværket i Danmark (2017), Chung J.C (2002)* Sanserom kan øke trivsel og livskvalitet og forhindre understimulering.

Andre eksisterende sansestimuli-relaterte tiltak som kan bli tatt i bruk er for eksempel: smak, vibrasjonsredskaper, massasje, berøring, krembehandling, turer i naturen og kontakt med dyr. Det som er felles med mange av disse eksisterende løsningene er at de ofte krever en terapeut eller veileder som kan gi tilsyn for å få full effekt. Gjerne ved at terapeuten bruker sansestimuli som et redskap ved siden av “vanlig terapi”. For at man skal kunne få maksimal uttelling så vil det være et fokus på å koble innholdet i sansestimulien.

### 2.1.3.2 Bruk av velferdsteknologier

De ulike tiltakene vi har tatt for oss fører med seg sine utfordringer, både i forhold til gjennomførelse og å unngå feilstimulering av pasientene. I mange studier på velværen hos disse pasientene er derfor velferdsteknologi et fokus. En av grunnene til dette er at problemene som skaper redusert livskvalitet hos denne demografien omhandler mangel på eller utfordringer med sysselsetting. Dette er i mange tilfeller med på å skape for lav

og ensidig sansestimuli hos pasientene. Pleiere på slike institusjoner har gjerne andre ansvar som de må ty til, og da er ofte TVen en grei måte å sysselsette pasientene på. *de Medeiros et al. (2009)* Bruk av film, bilder og lyd er gode metoder for å fasilitere sansestimuli, og det åpner opp muligheten for potensiell tilpasning av innhold gjennom valg av applikasjoner eller ulike programmer. I mange studier på velværen hos disse pasientene er derfor velferdsteknologi et fokus.

Problemet med å “sette” demenspasienten foran TV'en er ofte at de som sliter med demens ikke får noe særlig ut av å se på TV. Det var klart at evnen til å se på TV forsvant veldig tidlig i demens-utviklingen. Det virket som demensen påvirket evnen til å se på TV; det var uklart om det var fordi de ikke klarte å følge med på skjermen, eller om innholdet var meningsløst for dem og/eller de ikke forsto det. *Is Watching Television a Realistic Leisure Option for People with Dementia? (2015)* Det kan også ha noe å si om personen ser på TV i en gruppe, for eksempel i en fellesstue, eller alene. Dersom personalet ikke er tilstede kan det bli vanskelig for demensrammede å utnytte effekt av speiling. Om man er i en større gruppe kan det bli vanskelig å tilpasse sansestimulinivået til hver enkelt person. *Sundhedsstyrelsen (2019)* I følge studien *Is Watching Television a Realistic Leisure Option for People with Dementia? (2015)* så personene i studien mellom 1-6 timer TV til dagen før dem ble rammet med demens, hvor det etter sykdommen inntraff ble redusert til 1-2 timer, og 74% av respondentene sa at TV ikke lenger var en signifikant del av livet deres. De konkluderer med at det er det sosiale aspektet som er den relevante delen med å se på TV etter at demensen utviklet seg. Det kan skape en kvalitetstid for demenspasienten, selv om det generelt kan være vanskelig å evaluere livskvalitet hos demenspasienter. Det viktigste av alt er at man ikke forlater personen med demens alene forran TV'en.

I studien *de Medeiros et al. (2009)* testet de ut TV-vanene til 22 personer i et langtids sykehjem der alle hadde moderat til stor utvikling av demenssykdom. De observerte oppførselen deres før, i løpet av og etter TV-tittingen. De kom frem til at den aller mest vanlige aktiviteten var å ta seg en lur. I tillegg var det ofte mange som satt igjen etter programmet var ferdig og så på en blank skjerm uten programmer i opp til 30 minutter etterpå. Dette underbygger det ene funnet til *Is Watching Television a Realistic Leisure Option for People with Dementia? (2015)*, som omhandler at demenspasienter enten ikke klarer å få med seg det som skjer på skjermen, eller om innholdet er meningsløst for dem.



De testet også ut om det var forskjeller i adferd i forhold til hva slags type programmer som ble vist. Det var ett program som skilte seg ut, kalt "Venice". Dette programmet holdt beboernes interesse mer enn de andre. Det som skilte dette programmet ut fra de andre var at det ikke hadde et sammenhengende plott, men isteden inneholdt det stillbilder som ikke hang sammen kronologisk. Dette kan tyde på at de andre programmene hadde for høy grad av sansestimuli, og derfor var lettere å følge med på.

Det finnes også andre former for velferdsteknologier enn TV som kan bli brukt for å tilpasse sansestimuli, for eksempel applikasjoner på mobilenheter. Disse er ikke så utbredt, men det har blitt gjort studier på noen av disse, ofte i forhold til reminisensterapi.

### 2.1.3.3 Reminisensterapi

Reminisensterapi har som formål å fremkalle minner og erindringer hos pasienten ved å ta i bruk media som bilder, video, musikk og kjente gjenstander, eller at man bare fører samtaler om emner som er kjent i pasientens livshistorie. Reminisensterapi som metode bruker en form for sansestimuli som ikke bare er tilpasset til sansebehovet hos pasienten, men som fokuserer mest på å være spesifikk rettet mot pasientens livshistorie. Reminisensterapi kan foregå på gruppe- eller individnivå. Studier har vist at reminisensterapi hos demenspasienter kan forbedre kognisjon, depresjon, apatinivå, livskvalitet, oppførsel, psykososiale symptomer av demens og kommunikasjon mellom pasienten, familiemedlemmer og helsepersonell. *Fang Yu (2019)*

Reminisensterapi er en utbredt behandlingsform for demenspasienter. Selv om det er blitt gjort mye forskning på reminisensterapi, kan det være en utfordring å anvende teorien i praksis på grunn av at en terapeut ofte trengs for å gjennomføre behandlingen. Det har i nyere tid begynt å bli mer vanlig å bruke teknologiske hjelpemidler (TV, iPad og diverse andre avspillingsenheter) som en måte å levere reminisensterapi på en mer autonomisk måte.

I en pilotstudie hvor det ble forsket på effektiviteten av et interaktivt mimrespill *Fang Yu (2019)* fant forskerne ut at det var mulig å oppnå positive effekter på sinnsstemning og humør gjennom autonom reminisensterapi. Det var derimot ikke lett å få pasientene til å konsistent spille spillet selv, men med hjelp fra en pleier eller et familiemedlem så kunne det bli gjennomført uten en terapeut tilstede. Et annet interessant funn i

denne studien var at noen pasienter reagerte veldig positivt på gruppesesjoner med reminisensterapi, mens andre krevde en mer individuell, tilpasset versjon. I studien om TV som fritidsaktivitet for demente *Is Watching Television a Realistic Leisure Option for People with Dementia?* (2015), nevner de også i konklusjonen at TV generelt kan fungere som reminisensterapi dersom det er under rette omstendigheter og program som kan vekke minner hos demenspasientene.

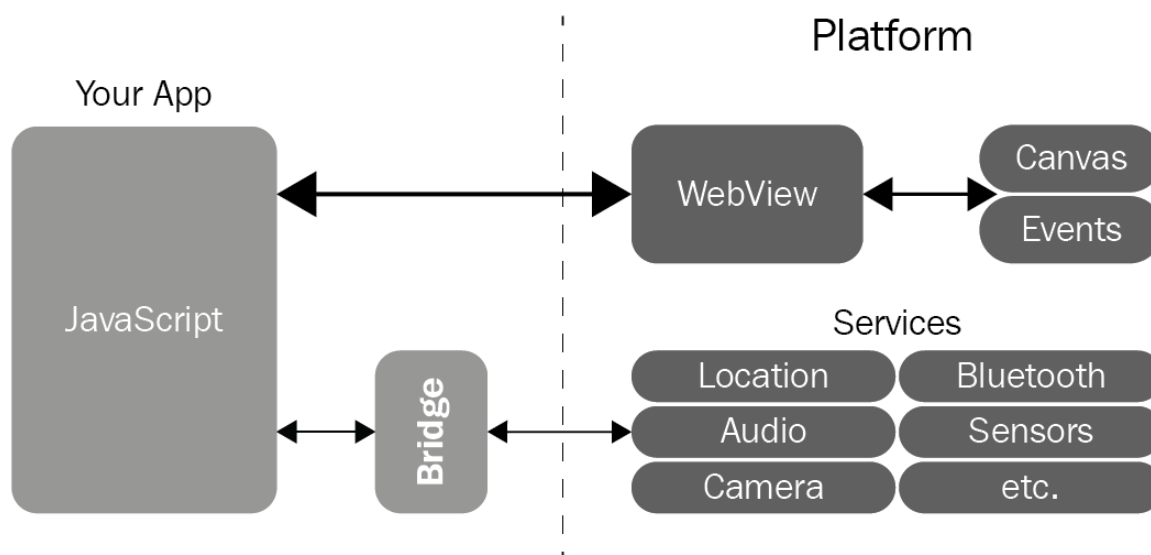
## 2.2 Teknologi

### 2.2.1 Utvikling på tvers av plattformer

Tverrplattformutvikling betyr å utvikle programvareprodukter eller tjenester for flere plattformer. Utviklere bruker forskjellige metoder for å imøtekomme forskjellige operativsystemer for en applikasjon. De to mest kjente metodene for å utvikle på tvers av operativsystemer for mobil er WebView og Bridge platforms.

#### 2.2.1.1 WebView Platform

Figur 2.1: Webview arkitektur

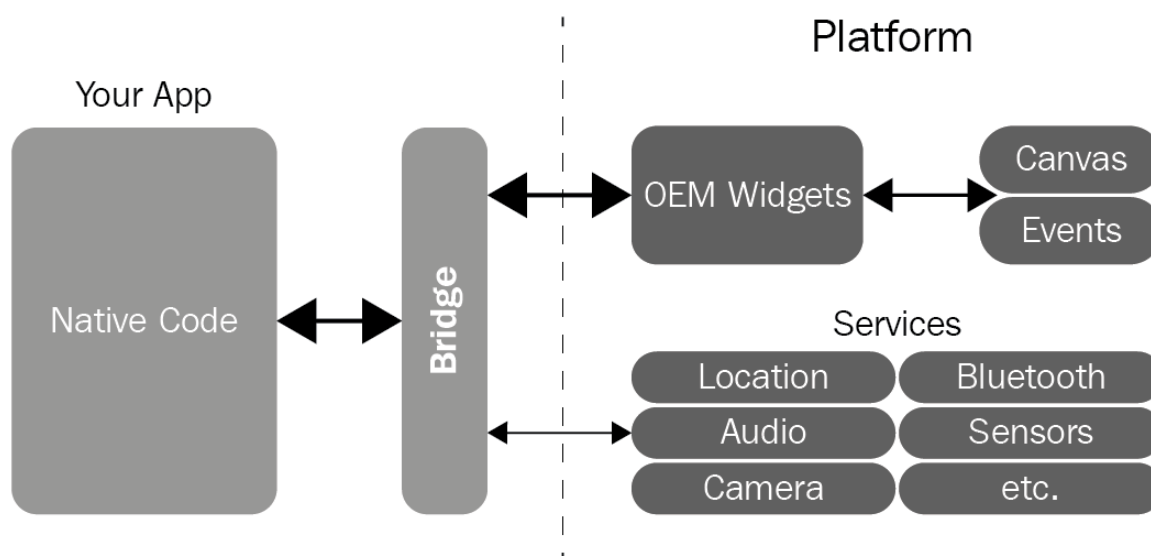


Ved å bruke WebView, også kalt hybrid cross-platform approach, kan man vise netttinnhold (HTML5) i en native enhet. Denne løsningen passer bra for frontend utviklere som har

erfaring med HTML5, CSS og Javascript. Systemet bruker en “bridge” for å bytte mellom JavaScript og det opprinnelige systemet. Effektiviteten av denne løsningen vil være avhengig av internett hurtigheten til brukeren. I tillegg bruker WebView lang tid på å aksessere funksjonalitetene i en native enhet, for eksempel bruk av kamera, GPS og Bluetooth.

### 2.2.1.2 Bridge Platform

Figur 2.2: Bridge arkitektur



En annen metode for utvikling på tvers av plattformer er å ha en bro som kommuniserer med native tjenester, OEM widgets og programmeringsspråket man koder i. Denne broen påvirker ytelsen av applikasjonen dersom brukeren krever tilgang til flere tjenester (Services) fra mobilenheten i løpet av kort tid.

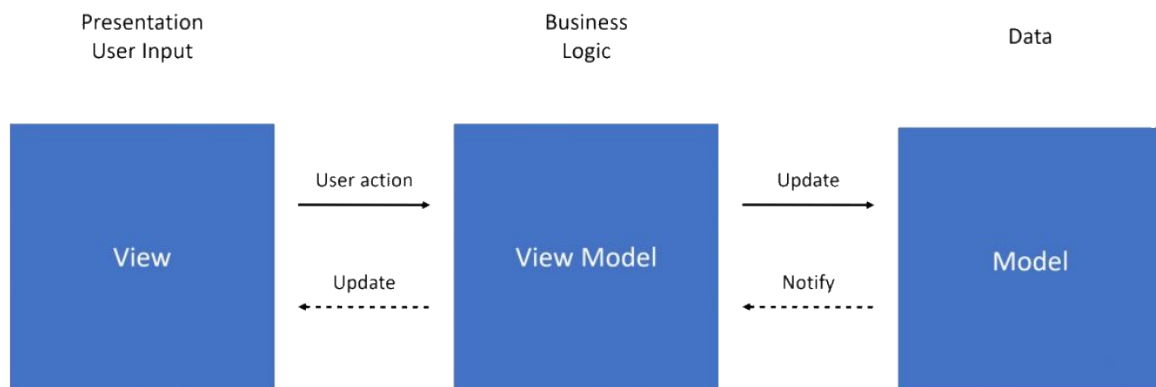
## 2.2.2 MVVM arkitektur

“Model–View–ViewModel” (MVVM) er en arkitektur som separerer det grafiske brukergrensesnittet (View) fra utviklingen av virksomhetslogikken (Model). Denne arkitekturen fører til at en View-klasse ikke er avhengig av en spesifikk Model-klasse ved at brukergrensesnittet går gjennom en ViewModel. En ViewModel styrer kommunikasjonene mellom brukergrensesnittet og logikk.

MVVM skiller seg fra kjente arkitekturer som “Model View Presenter” (MVP)

og “Model View Controller” (MVC). Presenter og Controller bestemmer direkte hva brukergrensesnittet skal vise. Siden ViewModel ikke har direkte kontakt med brukergrensesnittet, sender den istedet ut events/streams som brukergrensesnittet kan binde seg til. Dette gjør at ViewModel kan letter gjenbrukes og unit testes.

**Figur 2.3:** MVVM arkitektur



- View
  - Reagerer på nye endringer i ViewModel og oppdatere brukergrensesnittet
  - Fortell ViewModel om nye brukerinnganger (f.eks. knappeklikk, tekstendringer, skjermberøringer)
- Viewmodel
  - reagerer på brukerinnganger (f.eks. ved å endre modellen, starte nettverksforespørsler eller navigere til forskjellige skjermer)
  - Tilbyr data som View kan abonnere på
- Model
  - Representerer/styrer databasen
  - Representerer/styrer applikasjonen tilstand (state)

### 2.2.3 NoSQL database

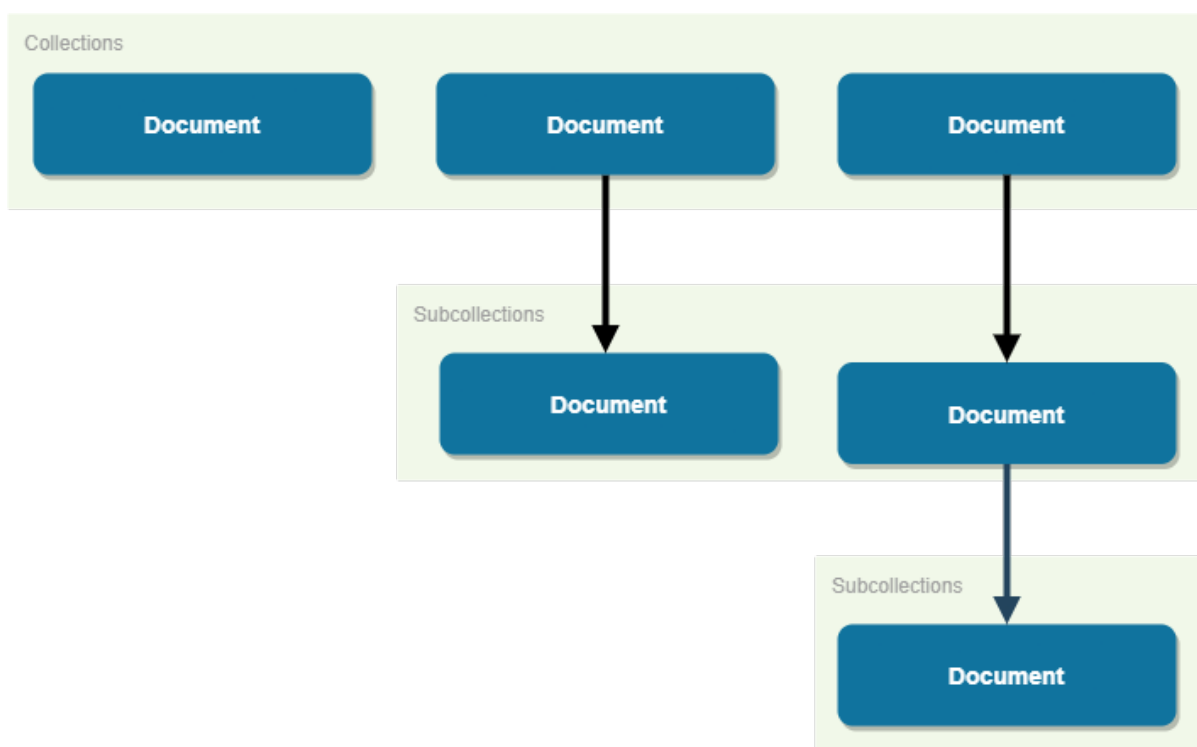
NoSQL databaser håndterer ustrukturerte data, ulikt fra en SQL relasjonsdatabase som bygger på matematiske relasjoner og bruker tabeller med rader og kolonner. NoSQL-

databaser gjør det lettere for utviklere å raskt opprette databasesystemer og skalere databasen. NoSQL kan lett skalere horisontalt, som betyr at man bruker flere maskiner til å lagre dataen, mens SQL databaser hovedsakelig må skalere vertikalt, som betyr å gjøre en maskin kraftigere (CPU, RAM).

### 2.2.3.1 Dokument orientert database arkitektur

En type NoSql-database er dokumentdatabaser. De organiserer dokumenter i grupper / samlinger. Dokumentene støtter nøkkel-verdipar og tilatter spørringer på alle attributter i et dokument.

**Figur 2.4:** Dokument orientert database arkitektur



### 2.2.4 Universell utforming mht. eldre personer

Når folk blir eldre, skjer det visse fysiologiske og kognitive endringer. Spesielt for de som blir rammet av demens og andre kognitive problemer, er det viktig at teknologien blir tilrettelagt på en slik måte at de kan ta den i bruk uten for store problemer.

Det er viktig at applikasjoner som bruker berørings skjermteknologi ikke har kompliserte finger bevegelser (gestures). Man må altså unngå å ha interaksjoner som krever to fingre.

Stadig flere eldre begynner å bruke nettbrett *Monica Anderson (2017)*, og det er derfor viktig at applikasjoner er responsive og tilpasset til nettbrett.

En annen viktig funksjon av applikasjonen er at navigasjonen er enkel. Navigasjonen bør være godt organisert og innholdet bør bli gradvis blir mer spesifikt jo dypere inn brukeren navigerer i strukturen. Brukeren skal alltid vite hvor de er i applikasjonen og hvordan man kommer til en destinasjon og tilbake/hjem navigasjon bør være lett tilgjengelig på alle skjermer av applikasjonen.

Hvis applikasjonen inneholder funksjonaliteter som kan være vanskelig å forstå, bør man ha en Onboarding som introduserer nye brukere med opplæringen av applikasjonen. Onboardingen bør inneholde detaljerte beskrivelser av applikasjonen. Brukere som har mer erfaring med moderneteknologi burde kunne hoppe over introduksjonen. I tillegg til Onboarding bør en instruksjonsguide være tilgjengelig senere i applikasjonen.

Når det kommer til farger i applikasjonen så kan nyanser av blått virke bleknet for eldre, og potensielt redusere kontrasten når blå elementer brukes i ett design. *POLYUK (2019)* Det er derfor viktig å unngå å bruke fargen blå for viktige elementer i brukergrensesnittet og ellers ha høy kontrast mellom fargene.

## 3 Valg av teknologi og metode

Dette kapitlet er delt inn i to deler; fremgangsmåte og valg av teknologi. I første delen blir de valgene vi har tatt som omhandler forskings og utviklingsprosessen beskrevet. I den andre delen er rettet spesifikt mot valg av teknologier.

### 3.1 Fremgangsmåte

#### 3.1.1 Fremgangsmåte for forskning

Som en sentral del av prosjektet var det opprinnelig planer om å gjennomgå A/B brukertester hos ulike sykehjem. Vi hadde blant annet en avtale med Melløsparken og Orkerød Sykehjem i Moss om å brukerteste applikasjonen. Vi skulle også drive kontinuerlig datainnsamling hos de institusjonene som tok i bruk tidlige versjoner av plattformen. Da korona-viruset inntraff, hadde pleiesenterne imidlertid ikke nok ressurser til å ha korrespondens med oss angående Televindu. Dermed bestemte vi oss for å basere arbeidet vårt på tidligere forskning istedenfor å samle inn data i felten selv.

Løsningen Televindu tidligere brukte for videoplattformen på iOS baserte seg mye på de samme prinsippene, og vi har dermed hatt muligheten til å dra nytte av og implementere resultatene fra brukertestene som har blitt utført på denne.

#### 3.1.2 Utviklingsprosess

##### 3.1.2.1 Arbeids- og rollefordeling

Scrum- og møterollene har blitt inndelt som følger:

- Scrumleder: Snorre Strand
- Ordstyrer: Erlend Sundøy
- Referent: Simen Tengs
- Veileder: Alexander Holt
- Produkteier og Oppdragsgiver: Øystein Hansen

- Prioriterer backlog
- Ansvarlig fra kundesiden
- Kontinuerlig kontakt med utviklingsteamet
- Sentral i godkjenningsprosessen

### 3.1.2.2 Scrum

Det har vært naturlig å ta i bruk en smidig utviklingsprosess med hensyn til arbeidsmengde og tidsramme. Dette passte også bra til antallet studenter på denne oppgaven. Vi valgte å ta i bruk Scrum. Det presenterte seg dog ikke nødvendig å inkorporere alle artefaktene og metodene som en Scrum prosess vanligvis har. Vi har å valgt å ta med de delene vi mente var mest nødvendige. Dette inkluderer sprinter, sprint-reviews og standup møter.

Det ble gjennomført sprinter på 2 uker hver. Det ble lagd en generell backlog som sammen med sprintene ble lagd ved hjelp av teknologiske verktøy. Dette finnes i utfyllende scrum dokumentasjon i vedlegg *D*. Vi delte user-storyene inn i mindre deler og dette ble issueene i backloggen. Ut ifra disse ble det lagd generelle milepæler for hver sprint der vi tok med de issueene vi trengte for hver sprint. Ved å dele arbeidet inn på denne måten oppnår vi god oversikt på både kort- og lang sikt angående tidsrammene for milepælene, og dermed oppnådde vi også god arbeidsflyt.

## 3.2 Valg av teknologi

### 3.2.1 Firebase

Firebase er en BaaS (Backend as a Service) som gir utviklere en plattform for å utvikle mobil og web-applikasjoner. Plattformen er en del av Google Cloud Platform (GCP) som tillater automatisk skalering og integrering av andre GCP-tjenester. Plattformen er “pay as you go/scale”. Det gjør plattformen passende for et oppstartsselskap som Televindu. Samtidig gjør det utviklingen enklere siden Firebase har 19 forskjellige tjenester i økosystemet sitt for utvikling, distribusjon og analyse av applikasjonen.

Vi har tatt i bruk disse tjenestene fra Firebase i prosjektet:

- Firebase Authentication



Vi bruker Firebase Authentication til å autorisere brukere i systemet. Firebase Authentication SDK'en tilbyr metoder til å lage og administrere brukere som bruker email og passord til å logge inn. Firebase authentication integrerer lett med de andre Firebase servicene vi bruker i applikasjonen.

- Cloud Storage for Firebase

Gjør det mulig for brukere å lagre innhold som musikk, bilder og videoer. Cloud Storage med Firebase har integrert Firebase Authentication, slik at man kan legge til ekstra sikkerhet for opplasting og lesing av filer. Dette er nyttig for oss siden vi kan garantere at kun administratorer kan laste opp bilder.

- Cloud Functions

Istedenfor at vi må vedlikeholde og ekspandere backendfunksjonalitet og serverkraft selv så fikser Firebase alt slikt for oss etter hvert som behovene vokser. Ved hjelp av denne teknologien kan vi enkelt integrere forskjellige systemer fra Firebase sitt økosystem og la dem spille av hverandre uten å bekymre oss for administrasjon av serverne. Funksjonene som ligger bak kulissene her har vi skrevet i Typescript og blir bare kjørt når spesifikke hendelser trigger dem.

- Google Analytics

Google Analytics er Google sitt system for analyse av bruk av applikasjonen. Ved å bruke dette så kan Google gi oss nyttig og egendefinert statistikk om brukerdata og -oppførsel. F.eks: hvilken videoer som blir sett, hva brukere søker etter og hvordan brukerspillelister blir brukt.

- Firebase Performance Monitoring

Tjenesten har innebygde funksjonaliteter som måler oppstartstid og hvor lang tid det tar å laste inn hver scene. I tillegg kan man måle tidsbruk på egendefinerte steder i koden. Dette hjelper oss med å finne flaskehalser i programmet, så vi kan optimalisere de delene.

- Firebase App Distribution

Firebase App Distribution gjør at utviklere kjapt kan distribuere betaversjoner av applikasjonen til testbrukere gjennom Firebase sin egen applikasjon. Dette gjaldt når

vi ved behov sendte betaversjoner til oppdragsgiver og diverse personer i Televindu, slik at de kunne teste applikasjonen og gi feedback. Dette var spesielt hjelpsomt i oppstartsfasen av prosjektet.

- Firestore

Firestore er en NoSQL dokumentorientert database. Grunnen til at vi har valgt å ta i bruk en NoSQL database istedenfor en tradisjonell relasjonell database, er at det er mye sjeldnere at man legger inn ny data eller endrer eksisterende data i databasen enn at applikasjonen må hente det. Firestore håndterer horisontal skalering i databasen automatisk, som gjør at det er en billig løsning spesielt på startfasen og når det begynner å koste penger for å opprettholde databasen, over 10GiB reads/writes per måned, vil Televindu allerede tjene mer enn nok.

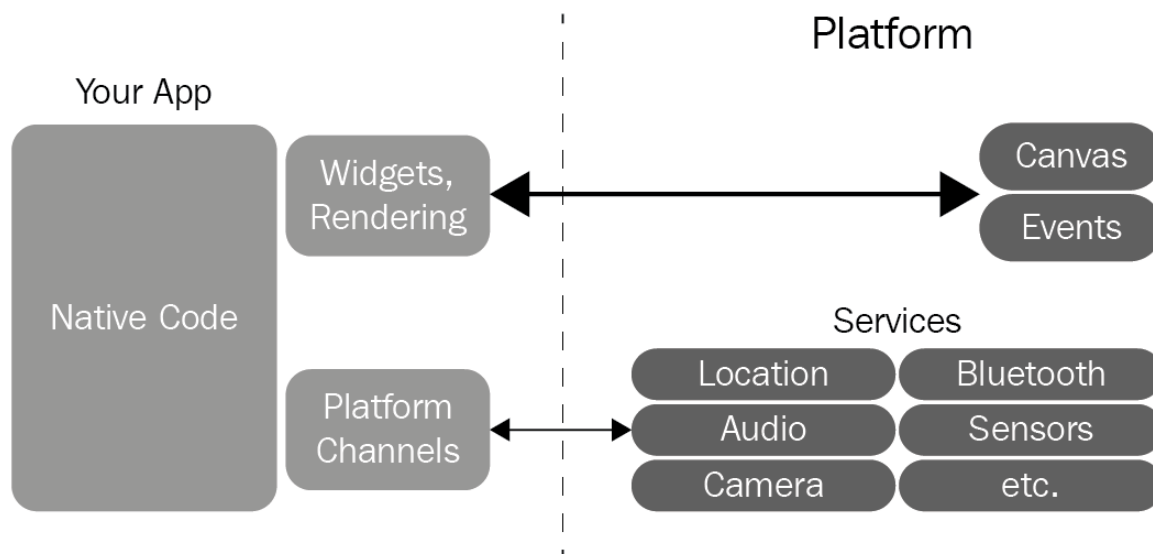
- Cloud Firestore Security Rules

Firestore Security Rules er en tjeneste som kontrollerer hvem som leser og skriver i Firestore. Man bruker et simpelt programmeringsspråk for å skrive regler på create/read/update/delte i databasen. I programmeringsspråket har man også tilgang til Firebase Authentication.

### 3.2.2 Flutter

*Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. <https://flutter.dev/>*

Vi har valgt å utvikle applikasjonen i Flutter, et tverrplattform rammeverk for programmeringsspråket Dart. Det kompilerer til lokal kode for både Android og iOS, som er de operativsystemene Televindu skal støtte. Den største fordelen med å bruke Flutter kontra Kotlin/Java og Swift er at det sparer mye tid, siden vi kun må lære oss ett språk og ikke må lage applikasjonen i to forskjellige programmeringsspråk.

**Figur 3.1:** Flutter arkitektur

Flutter bruker Platform Channels til å aksessere platform spesifikk kode, som Android og iOS. I den platform spesifikke koden har man da bruke enhetens services direkte. Man trenger dermed ikke å bruke en bro for å kommunikasjon med programmeringsspråket og services, som vil gi en bedre ytelsen av applikasjonen.

I tillegg til å være et tverrplattform rammeverk tilbyr Flutter både “Just in time” (JIT) og “Ahead of time” (AOT) -kompilering. Under utvikling bruker Flutter JIT-kompilering for å kunne utnytte hot-reload, der filene som har blitt endret blir raskt oppdaterert uten at applikasjonen mister sin tilstand. I produksjon bruker Flutter AOT-kompilering for bedre ytelse og for å redusere størrelsen på applikasjonen. I motsetning til Flutter bruker de fleste andre tverrplattform-løsninger bare JIT og er dermed avhengig av en bro med programmeringsspråket og plattformen.

Flutter er lagd av Google så Flutter har offisielle Google packages til å støtte ulike Firebase services. I tillegg er Flutter samfunnet aktive og har mange packages tilgjengelig. Dette gjør utviklingen av applikasjonen enda lettere. Et viktig eksempel på dette er “video\_player” og “flutter\_downloader”, som ga oss muligheten til å spille av og laste ned videoer uten å fokusere på native kode på iOS og Android.

Flutter sitt brukergrensesnitt er bygd opp av et tree av widgets, der hver widget beskriver hvordan det tilsvarende UI-elementet skal se ut ut i fra hvordan widgeten er konfigurert

og hva slags tilstand den er i. Hver gang tilstanden til en widget blir oppdatert, så vil widgeten bygge beskrivelsen sin opp på nytt og sammenligne den nye beskrivelsen med den gamle for å identifisere den minste mulige forandringen som kreves i “render tree” for å gå til den nye tilstanden.

Ved å bruke widgets kan man lett dele opp kode basert på innkapsling som vil øke ytelsen og utviklingen av applikasjonen. Ettersom en bra strukturert Flutter applikasjon vil ha små widgets der bare små deler av UI'en blir oppdatert der det trengs, istedenfor at hele skjermen blir oppdatert.

### 3.2.2.1 Provider (State management)

Provider er en package som gjør det lettere å aksessere tjenesteklasser og kontrollere data i applikasjonen. Provider passer bra i en MVVM arkitektur siden Provider inneholder klassen `ChangeNotifierProvider`. Hvis et View lytter etter en klasse som arver fra `ChangeNotifierProvider`, vil Viewet automatisk bli oppdatert hvis `notifyListeners()` blir kalt. Dette gjør prosjektet mer oversiktlig og leselig, fordi man lett vilt kunne se når brukergrensesnittet skal bli oppdatert. I tillegg vil Provider automatisk “rydde opp” klasser som ikke lenger er i bruk i Flutter treet.

### 3.2.3 Fastlane

Fastlane er en plattform som gjør prosessen med å distribuere Android og iOS applikasjoner lettere og kjappere. Ved å bruke plattformen kan vi automatisere byggingen og distribuering til Google Play Store, Apple App Store og Firebase app distribution. Dette var spesielt hjelpsomt ettersom prosessen med å bygge og distribuere apper i iOS er tidkrevende.

### 3.2.4 Github Actions

Vi har valgt å bruke Github Actions for Continuous Integration (CI). Med Github Actions kan vi automatisere spesifikke “jobber” som kjører fra skyen regelmessig istedenfor at vi må gjøre det manuelt hver gang. Det er et stort samfunn rundt Github Actions, som betyr at det er mange tilgjengelige ferdiglagde Actions fra andre utviklere som vi kan ta i bruk.

Vi bruker for eksempel Flutter building og deployment som Github Actions. Når noen av oss for eksempel lager en pull request eller pusher til master, bygges appen i skyen og så

---

sjekkes det om alt fungerer og testene er vellykkede før det eventuelt kan bli automatisk distribuert gjennom Firebase / Google Play Store / Apple App Store.

## 4 Resultater

xcolor

### 4.1 Vitenskapelige resultater

#### 4.1.1 Begrensninger

På grunn av korona-situasjonen har vi ikke hatt mulighet til å utføre datainnsamling og de planlagte brukertestene vi skulle utføre. Data og Empiri vil av denne grunn ikke bli en sentral del av dette underkapittelet. Fokuset her er produktets funksjoner og designvalg som konkret svarer på problemstillingen.

#### 4.1.2 Produkt og Design

Se kravdokument i vedlegg *B* for mer detaljert informasjon om de enkelte funksjonalitetene.

- Televindu offisielle spillelister

Som administrator har man mulighet til å opprette offisielle Televindu spillelister, som vises på hjem-skjermen og spilleliste-skjermen.

- Egne brukerspillelister

Sluttbrukere har mulighet til å lage, endre og slette egendefinerte spillelister. Administratorer kan også lage spillelister for brukere som ønsker det.

- Lokasjonsbaserte videoer (Videoer fra nærområdet)

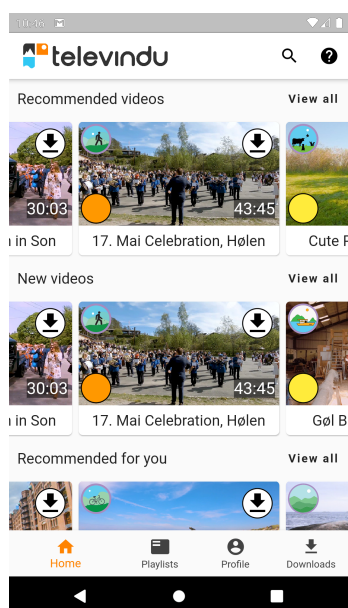
I hjem-skjermen vises det en videokarusell for for videoer som har blitt filmet i nærområdet, hvis man ikke har tillatt stedsproing, eller det ikke er noen videoer i nærheten.

- Anbefalte videoer

En av karusellene i hjem-skjermen inneholder videoer anbefalt til hver enkelt bruker. Disse blir anbefalt basert på hva slags videoer du har sett tidligere.

- Fargekode

Figur 4.1: Hjem-skjerm



Vi har designet fargekode til hver videoer, skalert på stimuli den produserer. grønn, gul og oransje. Alle videoer har en fargeidentifikasjon som forteller noe om hvilken grad med stimuli den tilbyr. Grønn - rolig, gul - middels, oransje - høy.

- Televindu egne ikoner etter Video types

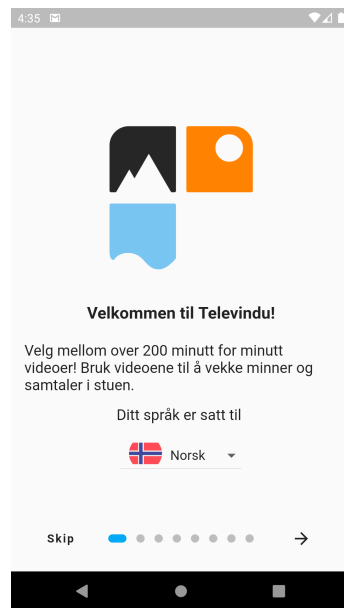
Televindu har supplert oss med design til ikonene for hver video type. Disse blir brukt slik at brukeren lett kan se hvilken type (Walk, Scenery, Animals etc.) hver video er.

## 4.2 Ingeniørfaglige resultater

### 4.2.1 Produktegenskaper og status på mål fra visjonsdokumentet

Alle målene som ble satt i starten av prosjektet i visjonsdokumentet i vedlegg A ble implementert. Det ble satt nye krav underveis i prosjektet som også er implementert. Disse funksjonalitetene og kravene til dem er beskrevet i kravdokumentet i vedlegg B. I dette kapitlet vil vi beskrive produktegenskapene utover det som ble nevnt i vitenskapelige resultater.

- Onboarding / Hjelpesfunksjon

**Figur 4.2:** Onboarding

Vi har utviklet en onboarding prosess som vises første gang man åpner appen. Denne tar brukeren gjennom en forenklet brukermanual. Det er fortsatt mulig å få tilgang til en tilpasset versjon av onboardingen i ettertid ved å trykke på et hjelpeikon øverst til høyre i applikasjonen, i appbaren.

- Egenskaper uten internett

Det er mulig å laste ned videoer og/eller spillelister for å se dem uten internetttilkobling. Hvis man mister internettforbindelsen eller aktiverer frakoblet modus i innstillinger, så vil man ha mulighet til å se nedlastet innhold.

- Brukerhistorikk

Målet som i visjonsdokumentet er beskrevet som “historie for brukere” har blitt endret på. Det er ikke en egen side hvor brukeren har en oversikt over alle videoene han har sett, som var den opprinnelige idéen. Det er nå mulig å se om man har sett en video fra før, og hvor langt man har sett ved hjelp av en progresjonsbar på thumbnailen til videoen. Dersom en bruker åpner en video han har sett før, vil videoen gjenoppta fra siste avspillingspunkt.

- Anbefalte videoer

Hver gang en bruker ser en video blir tagsene til videoen lagret i brukerens



subcollection i Firestore. Anbefalte videoer blir plukket ut basert på hvilke tags brukeren har sett på mest.

- Tilpasset for Nettbrett

Brukergrensesnittet er tilpasset for nettbrett ved at elementene er responsive og blir skalert i forhold til størrelsen på skjermen. På nettbrett kan man også bytte mellom landskap- og portrettmodus.

- Chromecast

I “video detail” er det et Cast-ikon oppe til høyre på skjermen, i appbaren. Hvis man trykker på den vil man se alle ledige Cast-enheter. Når man caster har man tilgang til pause og play fra appen på kildeenheten.

- Firebase

Firebase Analytics, Firebase Performance, Firebase Crashalytics og Cloud Messaging har blitt implementert ved å bruke Flutter sine offisielle Firebase packages. Firebase produktene er integrert i applikasjonen ved å bruke Provider.

- Søk etter video

Et av de nye kravene fra Televindu var at brukere skulle ha mulighet til å søke på videoer. Dette ble implementert ved at klienten henter metadata om alle videoene i Firestore og søker på klientsiden. Man kan søke ved å trykke på et søkeikon plassert oppe til høyre på skjermen, i appbaren.

- iOS versjon av applikasjonen

I iOS ble designet tilpasset for Apple sin standard, Cupertino, og videospilleren tilpasset slik at man kan bruke Airplay sine funksjonaliteter.

- Airplay

Et annet nytt krav var å implementere Airplay for iOS enheter. Med Airplay kan man koble til trådløst eller bruke HDMI-kabel for å vise videoer fra en iOS enhet på en TV. Ettersom Flutter’s “video\_player” package ikke støtter Airplay, måtte vi bruke en egendefinert Platform Channel, og kode dette native i Swift. Når brukeren bytter mellom Flutter og Swift sin videospiller er posisjonen bevart.

### 4.2.1.1 ViewModel kode

Hver hoved-View sjekker hva slags skjerm enheten bruker, og gir en tilpasset View dersom enheten bruker mobil, nettbrett eller desktop.

```
class ProfileView extends StatelessWidget {
  const ProfileView({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return ViewModelBuilder<ProfileViewModel>.reactive(
      viewModelBuilder: () => ProfileViewModel(
        Provider.of<NavigationServiceModel>(context, listen: false),
        Provider.of<AuthServiceModel>(context, listen: false),
        Provider.of<PreferencesRepositoryModel>(context, listen: false),
      ),
      builder:
        (BuildContext context, ProfileViewModel viewModel, Widget child) {
          return ScreenTypeLayout(
            mobile: _ProfileMobile(viewModel),
            tablet: _ProfileTablet(viewModel),
            desktop: _ProfileDesktop(viewModel),
          );
        },
    );
  }
}
```

Her ser vi ProfileView hvor ProfileViewModel blir initiert med alle servicene ViewModelen trenger, ved å bruke *Provider.of(context)*. Builderen bruker ScreenTypeLayout til å vise en mobil, nettbrett eller desktop UI widget.

### 4.2.1.2 Platform channel

Flutter kan kommunisere med Swift gjennom å bruke en Platform Channel med et felles navn.

Flutter kode som kaller på Swift kode:

```
await _platform.invokeMethod('playVideo', <String, dynamic>{
  'path': path,
  'isDownload': isDownload,
  'progress': _currentPositionSubject.value
});
```

Her kaller vi på *playVideo*-metoden i platform channelen 'flutter/video', med argumentene: "path", "isDownload" og "progress".

Swift kode som lytter etter kall fra Flutter:

```
playerChannel.setMethodCallHandler({
[weak self] (call: FlutterMethodCall, result: @escaping FlutterResult) -> Void in
  guard let strongSelf = self else {
    return result(FlutterError.init(code: "FlutterVideoPlayer does not exist"))
  }
  switch (call.method) {
    case "playVideo":
      strongSelf.playVideo(arguments: call.arguments, result: result)
      break
    case "pauseVideo":
      strongSelf.pauseVideo(result: result)
      break
    case "stopVideo":
      strongSelf.pauseVideo(result: result)
      break
    case "getProgress":
      result(Int(strongSelf.player.currentTime().seconds))
      break
  }
})
```

```

    default:
      result(FlutterMethodNotImplemented)
    }
  })
}

```

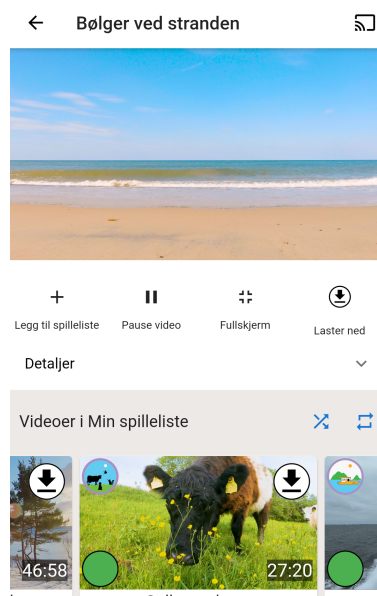
Her initierer vi en 'setMethodCallHandler' i Swift til å lytte etter kall fra 'flutter/video' platform channelen. Vi bruker en switch for å sjekke hvilken metode som blir kalt, i dette tilfelle vil metoden være 'playVideo' der vi starter en AVPlayer.

## 4.2.2 System som helhet

### 4.2.2.1 Design

Applikasjonen bruker "Video Cards" for å representere videoene i en "carousel". Video cardsene er delt opp i thumbnail og tittel. I thumbnailen er det en fargekode, et typeikon, nedlastingsstatus og lengden på videoen. I carouselene kan man sveipe horisontalt og sentralisere den valgte videoen i midten i carouselen. Det er mulig å trykke på "View All" og få en liste av video cards innenfor valgt spilleliste/kategori.

**Figur 4.3:** Video-skjerm



I "Video detail" skjermen har man mulighet til å legge en video til en spilleliste, pause/starte videoen, åpne fullskjerm og laste ned videoen. Det er ett ekspansjonspanel der man kan få mer informasjon om videoen, samt en carousel slider til spillelisten/kategorien videoen er

i.

Vi har brukt Google sitt Material Design *Material Design Design* (n.d.) for designvalg gjennom hele prosjektet. Material Design har retningslinjer for mange elementer i apputvikling, noen eksempler på hva vi har brukt: Appbar, Bottom navbar, Cards, Floating Action Buttons.

#### 4.2.2.2 Navigasjon

Hovedskjermen er delt opp i “hjem”, “spilleliste”, “profil” og “nedlastinger”. Disse er representert i en bunn navigasjonsbar.

- Hjemskjerm

I hjemskjermen vises det en rad av horisontale videocarousels, hvor anbefalte videoer, nye videoer, videoer i nærheten og utvalgte spillelister blir vist. Her kan man også søke etter videoer ved å trykke på søkeikonet i appbaren.

- Spillelisteskjerm

Spillelisteskjermen er delt opp i to tabs; Televindu sine offisielle spillelister og bruker spillelister. I bruker spilleliste taben har man mulighet til å lage, endre og slette brukerspillemlister.

- Profilskjerm

I profilskjermen har man muligheter til å endre språk, streaming- og nedlastningskvalitet, passord og aktivere frakoblet modus. Andre funksjoner i profil skjermen: navigerer til kontakt oss, brukervilkår, personvernærklaring, endre abonnement og logge ut av applikasjonen.

- Nedlastningskjerm

I nedlastningskjermen ser man videoer som blir/har blitt nedlastet, og man kan velge å avbryte, pause eller slette nedlastede videoer.

## 4.3 Administrative resultater

Detaljer om timelister, tidsbruk, møter og referat finnes prosjekthåndboka i vedlegg ???. Her finnes også prosjektmål fra visjonsdokumentet samt planlagte milepæler i framdriftsplanen. De fleste kravene i framdriftsplanen ble gjennomført innen de estimerte datoene, med unntak av onboarding og Firebase Analytics som skulle bli ferdig innen 1. mars. En del av vår oppgave var å distribuere en betaversjon av applikasjonen til Melløsparken og Orkerød Sykehjem i Moss. Dette skulle også i følge framdriftsplanen bli gjennomført innen 1. mars. Dette lot seg ikke gjøre på grunn av koronasituasjonen, som førte til store endringer i forskningsdelen av prosjektet. Televindu satt gjennom prosjektet nye krav som ble integrert i sprintene og gjennomført fortløpende. De nye kravene var å implementere fargekode og televindu typeikon for hver video, søkefunksjon, airplay støtte og mulighet for å lage egne brukerspillelister. Noen krav ble simplifisert; det gjelder anbefalte videoer, brukerhistorikk og søkefunksjon.

## 4.4 Utviklingsprosessen

Vi brukte en tilpasset Scrum metodikk, der vi hadde med de elementene vi synes var nyttige for vårt use case, i og med at vi kun var tre personer i gruppen. Vi hadde to retrospekter i løpet av prosjektet, en etter de første to sprintene og en til etter sprint fire. Vi startet med å lage en backlog med alle user stories fra førsteutkastet av kravdokumentasjonen i Trello sitt Issue Board som “tickets”. Vi delte deretter opp ticketene inn i 6 sprinter etter fremdriftsplanen. Ettersom nye krav kom fra Televindu, lagde vi nye tickets og la dem inn i sprintene underveis. I starten av prosjektet brukte vi de ledige grupperommene vi kunne finne på NTNU sine campus og biblioteker i byen til å jobbe og ha standups. Da koronarestriksjonene inntraff brukte vi Discord Voice som samhandlingsverktøy.

Sammendrag av hver sprint, issue board og retrospektiv ligger i scrum dokumentet i vedlegg *D*.

## 5 Diskusjon

Diskusjonskapittelet vil være strukturert på samme måte som resultatkapittelet. Vi vil her forklare valgene vi har tatt samt diskutere resultatene fra forrige kapittel og hvordan de relaterer til problemstillingen.

### 5.1 Vitenskapelig diskusjon

#### 5.1.1 Televindu som sansestimuli og reminisensterapi

Det er beskrevet ulike eksisterende metoder som blir brukt for å oppnå sansestimuli hos demenspasienter i teorikapitlet. Siden vi skulle utvikle en velferdsteknologi med tilpasset sansestimuli i fokus, var det viktig for oss som utviklere at vi hadde i bakhodet at mange av de eksisterende formene for tilpasset sansestimuli ofte har en utfordring med at de lett kunne over- eller understimulere pasientene. Det er noen brukere som har behov for å roe seg ned, mens andre trenger å “våkne opp”. Av denne grunn har vi utviklet fargekoder som en lett måte å spesifisere sansenivået på hver enkelt video. Vi har valgt å dele inn i høyt, middels og lavt sansenivå. Målet er at pleiere lett skal kunne finne passende videoer til situasjonen.

I studien *Is Watching Television a Realistic Leisure Option for People with Dementia? (2015)* kom forskerne frem til at mange demenspasienter mistet evnen til å få noe ut av å se på TV tidlig i sykdomsløpet. Grunnen til dette kan være at de ikke klarer å følge med på skjermen eller at de ikke forstår innholdet. I følge *de Medeiros et al. (2009)* studien, var det det også variert hvor mange pasienter som aktivt fulgte med på skjermen. Et viktig funn her, var at mange pasienter fulgte nøye med på programmet “Venice”, som var en presentasjon av stillbilder som var uavhengig av hverandre.

Selve innholdet i videoene til Televindu er av lignende struktur som programmet Venice i den forstand at videoene ikke er avhengige av hverandre og at hver enkelt video ikke har et sammenhengende plott. Vi tror at det vil det være lettere for demenspasientene å følge med på innholdet og vil dermed få nytte av den tilpassede sansestimulien som Televindu plattformen tilbyr. Vi har utviklet flere funksjoner som skal samle og tilpasse innholdet enda bedre til den enkelte brukeren.

Videoer i Televindu er delt inn i forskjellige kategorier. Det er for eksempel treningsvideoer der Televindu har filmet lange sykkelturer hvor sluttbrukeren kan sykle på spinnings sykkel mens de ser på. Andre eksempler på kategorier er “Walk”, “Scenery”, “Animals” og “Train”. Vi har implementert type-ikoner tilhørende hver enkelt kategori slik at det skal være enda mer oversiktlig. Spillelistene som er standard baserer seg på disse kategoriene. Vi har også lagt til rette for at sluttbrukere skal kunne opprette og administrere egne spillelister. Ved å lage egne spillelister får man muligheten til å gjenoppta videoer som brukeren har respondert positivt på. Dette vil gjøre det lett for helsepersonell å ha passende videoer lett tilgjengelig for spesifikke brukergrupper. Det kan for eksempel være nyttig ved at pleiere har hurtig tilgang til beroligende innhold for utagerende pasienter.

I studien *Is Watching Television a Realistic Leisure Option for People with Dementia?* (2015) nevner de at TV generelt kan fungere som reminisensterapi dersom det er under rette omstendigheter og det vises program som kan vekke minner hos demenspasientene. Den lokasjonsbaserte spillelisten vi har utviklet er lagt opp til å fungere reminiserende for brukere ved å vise innhold som er filmet i nærområdet til sykehjemmene. Televindu har også fått tilgang til norske historievideoer fra Digitalt Museum <https://digitaltmuseum.no> (Accessed: 2020-03-10) og lagt disse videoene i egne spillelister. Studier har vist at reminisensterapi har dokumentert effekt på blant annet kognisjon, apati og livskvalitet. Derfor er disse videoene viktig for å kunne bedre livene til brukerne. Det er også slik at det ofte må være en terapeut tilstede ved de fleste former for reminisensterapi *Fang Yu* (2019). Televindu skiller seg ut ved at man ikke trenger profesjonelle terapeuter for å dra nytte av innholdet, men det har større effekt dersom en helsearbeider er tilstede. Dersom helsepersonell ikke er tilstede, kan det være vanskelig for pasientene å speile følelsene sine. Det er viktig å ta i betraktning om applikasjonen blir brukt individuelt eller i en gruppe. Vi antar at størrelsen på gruppen vil variere. Dette vil ha en effekt både på tilpassing av stimulnivået og hvor godt Televindu vil fungere som reminisensterapi. Desto større gruppen som ser på er, desto større sjanse er det for at enkelte i gruppen vil bli under- eller overstimulert av innholdet. Grunnen til dette er at det må gjøres ett kompromiss når man skal velge video, dersom det er brukere i gruppen som har forskjellige sansebehov. Hovedformålet med anbefalte videoer-funksjonen er å gjøre det lettere å finne passende nytt innhold. Denne funksjonen vil fungere best dersom man bruker som enkeltperson.



Da vil man kunne få anbefalinger spesifikt for seg selv. Den kan også fungere for grupper til en viss grad. Hvis en gruppe kommer til enighet om en type video som passer best for fellesskapet, vil dermed denne funksjonen etterhvert kunne forutse gode alternativer til fellesskapet. Dette gjelder når det er de samme gruppene som går igjen. Da vil prosessen med å finne videoer som passer bra for fellesskapet bli lettere. I studien *Is Watching Television a Realistic Leisure Option for People with Dementia? (2015)* konkluderer de med at det var det sosiale aspektet ved å se på TV som var mest relevant for demenspasienter, og at det var viktig å ikke forlate demenspasienten forran TV-en. Vi antar at selv om brukerne at applikasjonen får et mindre tilpasset innhold i større grupper, vil Televindu uansett fungere som et sosialt samlingspunkt.

## 5.2 Forbedringspotensialer

Selv om vi ikke hadde mulighet til å teste applikasjonen ut i praksis, ser vi i ettertid noen svakheter og potensialer for forbedring med løsningen vår.

Idéen med lokasjonsbaserte videoer var at demensrammede lettere skulle kunne få tilgang til videoer av sentrale steder i byen sykehjemmet befinner seg i, og på den måten kunne reminisere der de vokste opp. Vi har vært fokusert på å automatisere prosessen med å finne riktig lokasjon, og Televindu er aktivt ute og filmer inn innhold. Det vi ikke hadde tenkt så mye over før prosjektet nærmet seg slutt, er at det kanskje er mange som ikke er fra det området hvor de befinner seg når de ser på Televindu. Vi ser i ettertid at det ville vært optimalt å ha automatisk lokalisering som standard, men fortsatt kunne la brukeren velge lokasjon selv, om de skulle ønske det. På denne måten ville lokasjoner blitt en mer omfattende kategori, og beboere ville hatt et større utvalg for hvilken epoke fra livet de vil se på innhold fra.

Anbefalte videoer blir anbefalt basert på Televindukontoen, som gjerne vil bli brukt av flere individer eller i grupper. Hvert individ som bruker Televindu vil ha et forskjellig behov for sansestimuli, men én bruker vil gjerne ofte kreve den samme typen stimuli som han/hun har krevd tidligere. Med anbefalte videoer kan det dermed bli lettere å velge innhold uten at brukeren må lete etter tag-spesifikke videoer selv. Siden reminisensterapi er veldig subjektivt så vil ikke de anbefalte videoene tilføye noe der. Etter hvert tenkte vi på at det i helseinstitusjoner gjerne vil være forskjellige grupper med beboere som ser på

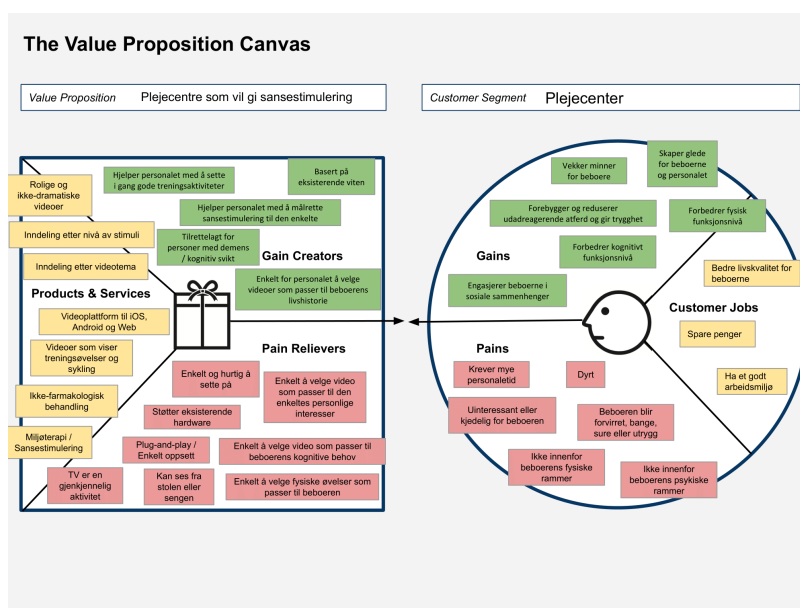
Televindu. For eksempel setter gjerne en pleier på videoer med et visst stimulinivå for en gruppe beboere, før de senere spiller av noe annet for en annen gruppe som har andre behov. Hvis dette alltid skjer på samme konto, så vil anbefalte videoer etter hvert bli helt baseløst. Dette hadde vi gjerne tenkt på mye tidligere om vi hadde fått utført brukertestene vi hadde planlagt. Et forbedringspotensiale som Televindu eventuelt kan implementere ved en senere anledning er å ha flere “profiler” per konto, slik som f. eks. Netflix gjør det. Da vil pleieren kunne velge hvilken profil som skal være aktiv til enhver tid og anbefalte videoer vil være profilavhengig istedenfor kontoavhengig. Det vil selvfølgelig avhenge av om disse kontoinnstillingene faktisk ville blitt brukt. Gjerne tvinge sluttbrukeren til å velge profil ved oppstart av Televindu og etter inaktive perioder.

Under utviklingen brukte Televindu Vimeo som VOD løsning, dette førte til at man ikke kunne ha adaptive bitrate streaming. Brukeren må derfor manuelt gå inn i innstillinger og velge kvalitet på streaming. Man kan velge mellom 360p, 540p, 720p og 1080p. Ved at Televindu brukte Vimeo for å distribuere videoer var alle videoer offentlige etter man fikk streaming URL-en fra databasen. Videoene som blir lastet ned er ikke kryptert, som fører til at Android enheter kan se videoene i andre videospillere og dele dem videre. Når Televindu i fremtiden skal håndtere video distribusjon selv, må Televindu ta hensyn til økt sikkerhet med autorisering og kryptering av videoer.

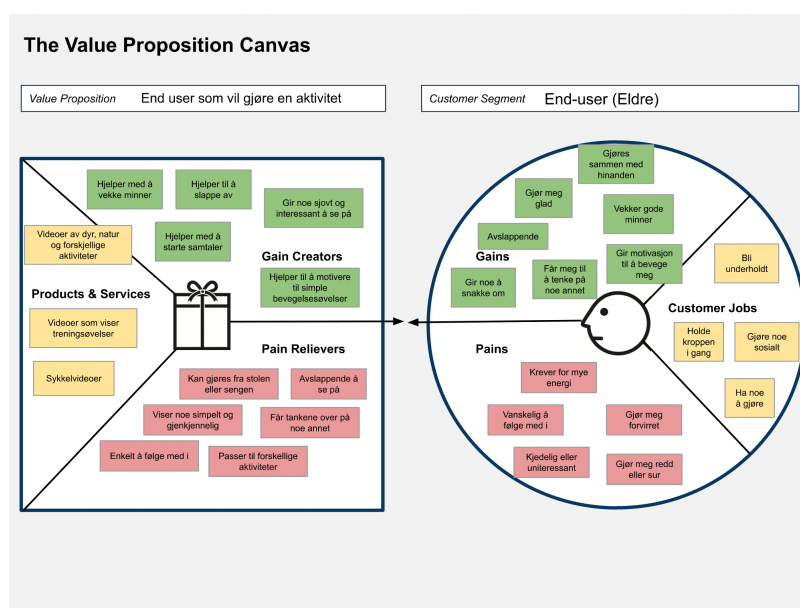
### 5.3 Helhetlig systemperspektiv

Vi mener at prosjektet skaper en god verdi for en oversett og sårbar gruppe i samfunnet. Prosjektet er et “shared value” prosjekt som vil si at det gir en økonomisk gevinst for firmaet samtidig som det gir en samfunnsmessig gevinst. Televindu har utarbeidet et “value proposition canvas” som peker på behov og verdier hos pleiesenterne og beboerne.

Figur 5.1: VPC Pleiesenter



Figur 5.2: VPC Beboer



Som vi kan se på 5.1 og 5.2 så er det mange muligheter for å skape gode verdier for både helsefagarbeidere og beboere på pleiesenterne. Likevel er det lite konkurranse innen sakte-TV for personer med demens. Det vil derimot være mulig å finne lignende innhold på mer generelle plattformer som Youtube og NRK, men ikke noe like spesialisert som det Televindu vil være.

### 5.3.1 Ethiske problemstillinger

Selv om Televindu skal bringe gode verdier på pleiesenter, er det også noen etiske diskusjoner vi vil ta stilling til.

Når det kommer til innsamling av data så er det spesielt viktig for oss at vi følger GDPR, siden en stor del av våre brukere vil inneholde en sårbar pasientgruppe. Vi samler ikke inn data på pasientene, ettersom de ikke er kjøperne av Televindu kontoen. Vi samler kun inn data om institusjonene som melder seg på. I tillegg har vi lagt vekt på sikkerhet for lagring av dataene, ved å bruke Cloud Firestore Security Rules og Server-side kryptering. I tillegg er dataene som blir samlet inn gjennom Google Analytics anonyme. Detaljer om disse tjenestene er beskrevet i metodekapittelet.

Ved at pleiesentere integrerer Televindu i sine daglige aktiviteter, vil det være mulig å redusere bruken av pleiere. Det vil bli lettere å sysselsette pasienten, som ble nevnt i teori ref. Det er derimot ikke meningen at pleiere skal parkere beboere foran TVen og ikke være til stede. Televindu har derfor et fokus på å tydeliggjøre at plattformen alltid skal brukes sammen med personalet.

Som vi har vært inne på i teorikapittelet så kan beboere på slike helseinstitusjoner være sensitive til stimuli. Hva hvis en video med for høy stimuli blir vist til en beboer som ikke tåler det? Eller en video fra havet blir vist til en person med vannskrekk? Televindu vil da ha en negativ effekt på en sårbar person. Vi gjør vårt beste ved å tydelig markere videoens innhold gjennom fargekode, symbol, thumbnail, tittel og beskrivelse, men det er personalets ansvar å benytte seg av dette hjelpemiddelet på en forsvarlig måte.

## 5.4 Gruppearbeid refleksjon

Vi startet prosjektet 3. februar, ettersom alle tre i gruppa var på utveksling i Nederland hvor høstsemesteret varer ut januar. I startfasen lette vi etter kontorplasser, men da vi ikke fant noe innenfor budsjettet til Televindu så endte vi opp med å bare bruke ledige grupperom på campus og bibliotek. Det har gått bra, vi fungerer godt sammen og spiller på hverandres styrker. Da koronasituasjonen gjorde at fysisk samhandling ble umulig, fortsatte vi hjemmefra og holdt standups og diskusjoner over Discord, som er en applikasjon for kommunikasjon over nett. Der var det litt vanskeligere å samkjøre arbeidet

i starten, men vi fant ut av det etter hvert.

## 5.5 Diskusjon av Ingeniørfaglige og Administrative resultater

I de første ukene i prosjektet brukte vi en del tid på å utforme problemstillingen vår, holde møter med veileder og oppdragsgiver og å sette oss inn i Dart, Flutter og Firebase. Vi satte estimerte overordnede mål for hver måned i framdriftsplanen.

Vi brukte mer tid enn forventet på å sette oss inn i de nye språkene og det eksisterende systemet. Det gjorde at Firebase Analytics og Onboarding, som skulle bli ferdig før 1.mars, ikke ble fullført i tide. På slutten av første sprint var alle kjent med prosjektet og det nye programmeringsspråket.

Underveis i prosjektet fikk vi nye krav fra Televindu, disse ble satt som nye “user stories” i kravdokumentet. De nye kravene var blant annet å implementere fargekoder, typeikoner, søkefunksjon, Airplay og mulighet for å lage egne brukerspillelister. Dette gjorde at vi måtte omprioritere tidsbruken i noen av sprintene, og brukerhistorikk og anbefalte videoer ble mindre omfattende enn planlagt. Den originale planen var å implementere anbefalte videoer funksjonen ved å bruke “Firebase ML-kit”, som er Firebase sitt toolkit for maskinlæring. Dette kunne hjulpet oss med å bedre forutsi hva hva brukeren vil like, men vi bestemte oss etter hvert for å prioritere andre funksjonaliteter og heller bruke en tag-basert løsning.

Vi begynte med å implementere brukerhistorikk i sprint 3. Vi så på det tidspunktet for oss at det skulle bli mulig for brukere å se en logg over hvilke videoer de har sett. Vi lagde backend-støtten i denne sprinten. Etter hvert endret Televindu dette kravet, og ville ikke lenger ha en loggside. I sprint 5 fortsatte vi på å implementere brukerhistorikk, hvor vi heller lot brukerne se om de har sett en video, og i så fall hvor langt de har sett, ved å markere det på thumbnailen til videoen.

Søkefunksjonen skulle opprinnelig bli implementert via API-et Algolia som allerede var integrert i Televindu sin backend da vi begynte på prosjektet. Etttersom Algolia ikke støtter Flutter enda, og det ville vært for tidkrevende å manuelt sette opp HTTPS forespørsler, simplifiserte vi denne funksjonen. Nå henter applikasjonen informasjon om alle videoer og

søker på klientsiden, men siden antall videoer i databasen ligger på ca 200, fungerer dette som en midlertidig løsning frem til Algolia støtter Flutter.

De fleste sykehjemmene Televindu har vært i kontakt med har ikke Chromecast, og bruker heller et Lightning Digital AV Adapter fra iOS-enhet til TV. Derfor var ett av de nye kravene fra Televindu å implementere støtte for Airplay, slik at de kan bruke det oppsettet for å se på Televindu. Dette var en så viktig funksjonalitet at distribusjonen av iOS-applikasjonen ble satt på vent inntil støtte for Airplay var ferdig. Å implementere dette var tidkrevende og komplisert, ettersom Flutter sin `video_player` package ikke støttet Airplay. Vi måtte sette oss inn i et iOS native programmeringsspråk, Swift, og lage en egendefinert Flutter Platform Channel for å la Flutter og Swift kommunisere med hverandre.

## 5.5.1 Generelle valg av prosjektstruktur

### 5.5.1.1 State management

I starten av prosjektet brukte vi “`get_it`” som state management. Med `get_it` har man tilgang til servicene hvor som helst i koden. Det gjorde det enklere for oss på denne tiden i prosjektet, men introduserte mer usikkerhet. Etter hvert som vi mer erfarne med Flutter og prosjektet vokste, gikk vi over til å kun bruke Provider. Flutter anbefaler selv å bruke Provider til state management.

Provider trenger mer “boilerplate kode” enn `get_it` og er avhengig av “context” for å få tilgang til servicene, men vil hjelpe med å holde prosjektet strukturert når applikasjonen blir større og mer komplisert. Med Provider er alle servicene initialisert i Widget treet. Det betyr at man kun har tilgang til disse servicene og providersene ved bruk av context i widget treet. Provider vil automatisk dispose og lage objekter etter behov, og vi kan styre når brukergrensesnittet skal bli oppdatert ved å bruke `notifyListeners()`. Ved at alle providersene er under Widget treet forhindrer man sirkulær avhengighet. I tillegg kan man lettere debugge siden man har tilgang til Provider i Flutter DevTools.

**Figur 5.3:** Service-providers i applikasjonen

Her er et bilde av Flutter dev tools som illustrerer alle servicene vi bruker i prosjektet. Vi injecter alle servicene i applikasjonen i en MultiProvider før MainApplication blir initialisert. Dermed kan vi kalle på dem lenger nede i Widget treet ved å bruke *Provider.of(context)*, da vil Provider søke opp i Widget Treet etter servicen.

### 5.5.1.2 Unit testing

Alle service klassene implementerer en abstrakt klasse som definerer tilgjengeligheten (private, public) til variabler og funksjoner. Vi gjør det slik sånn at vi lett kan unit teste service klassene med *Mockito*, som er en Dart package for unit testing.

```
class MockAuthService extends Mock implements AuthServiceModel
```

Her ser vi at vi lager en ny klasse “MockAuthService” som implementerer “AuthServiceModel”. Dermed kan vi injisere denne mock-klassen i en klasse som trenger AuthServiceModel for å simulere data og funksjoner.

### 5.5.1.3 View

I dag bruker nesten hele applikasjonen “mobile view”, dvs. tablet og desktop widgetene returnerer bare brukergrensesnittet for mobil i koden. Grunnen til at det er satt opp slik er fordi Flutter vil støtte desktop- og web applikasjoner i fremtiden. Det finnes allerede en

betaversjon av Flutter som støtter web og macOS. Dette gjør at man lett kan tilpasse brukergrensenettet på forskjellige enheter, f.eks. ved å ha navigasjonen i en app-drawer i stedet for en bottom navigation bar, hvis man er på en desktop skjerm.

```
return ScreenTypeLayout(  
  mobile: _HomeMobile(viewModel),  
  tablet: _HomeTablet(viewModel),  
  desktop: _HomeDesktop(viewModel),  
);
```

Ett av kravene beskrevet i visjonsdokumentet var å tilpasse applikasjonen for nettbrett. Grunnen til at vi returnerer mobile view på nettbrett er fordi widgetene i seg selv er responsive. Med widgetene responsive ser applikasjonen allerede veldig bra ut på forskjellige skjermstørrelser på nettbrett, så det var ikke nødvendig å gjøre drastiske endringer i brukergrensesnittet.



## 6 Konklusjon

Det har blitt gjort mange studier på effekten av sansestimuli og reminisensterapi hos demenspasienter som har vist positiv effekt på livskvalitet og forbedring av utagerende adferd. Tradisjonelle metoder for reminisensterapi som er mer omfattende, og gjerne gjennomført sammen med en terapeut, vil ha en større effekt enn funksjonene Televindu tilbyr. Innhold som kan ha en reminiserende effekt antar vi uansett vil være med på å øke trivselen blant brukere dersom det er pleiere tilstede, slik at brukerne kan utnytte effekt av speiling. Vårt produkt prøver å oppnå disse effektene på en mye mer tilgjengelig måte enn eksisterende tilbud.

Siden personer med demens som får feil mengde sansestimuli ofte vil kompensere med utagerende adferd, vil det å kunne berolige eller stimulere brukere av applikasjonen være den viktigste faktoren for å øke trivsel og livskvalitet blant brukerne. Applikasjonen tilbyr en oversiktlig måte for å finne tilpasset innhold som kan stimulere brukere etter deres behov. En svakhet med løsningen er at tilpasningseffekten blir redusert når applikasjonen blir tatt i bruk av større grupper. Ved større gruppesesjoner er det stor sannsynlighet for at enkelte brukere ikke vil få dekket sine sansebehov. Da risikerer man å oppnå motsatt av ønsket effekt. Vi vil anbefale personer som skal utføre lignende prosjekter i fremtiden å utforske denne problemstillingen mer.

Det vil være demenspasienter som ikke vil dra like mye nytte av Televindu dersom de ikke liker eller klarer å følge med på TV. En stor faktor utover tilbudet om sansestimuli og reminisensterapi er det sosiale aspektet. Det spiller en viktig rolle i å øke trivsel blant demente, og applikasjonen vil kunne fungere som et samlingspunkt på helseinstitusjoner. Dersom en demenspasient ikke har spesielle behov for å beroliges eller stimuleres, kan applikasjonen også bli tatt i bruk som ren underholdning, selv om dette ikke er hovedmålet.

Ettersom vi mistet muligheten til å teste ut applikasjonen og gjennomføre brukertester på sykehjem, førte det til at vi ikke hadde konklusive data og tilbakemeldinger å basere oss på. Vi vil anbefale at Televindu samler inn data og utfører brukertester av applikasjonen når helseinstitusjoner har anledning, for å videre evaluere effekten applikasjonen kan ha på livskvaliteten til brukerne.

## Referanser

Chung J.C, Lai, C. C. P. . F. H. (2002), 'Snoezelen for dementia'.

de Medeiros, K., Beall, E., Vozzella, S. & Brandt, J. (2009), 'Television viewing and people with dementia living in long-term care'.

**URL:** [https://www.researchgate.net/publication/245771421\\_Television\\_Viewing\\_and\\_People\\_with\\_Dementia\\_Term\\_Care\\_Pilot\\_Study](https://www.researchgate.net/publication/245771421_Television_Viewing_and_People_with_Dementia_Term_Care_Pilot_Study)

Fang Yu, Michelle A. Mathiason, K. J. J. E. G. D. K. (2019), 'Memory matters in dementia'.

**URL:** <https://www.sciencedirect.com/science/article/pii/S2352873719300629>

FHI (2014), 'Demens'.

**URL:** <https://www.fhi.no/nettpub/hin/ikke-smittsomme/demens/>

<https://digitaltmuseum.no> (Accessed: 2020-03-10), <https://digitaltmuseum.no/>.

*Is Watching Television a Realistic Leisure Option for People with Dementia?* (2015).

**URL:** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4404928/>

*Material Design Design* (n.d.), <https://material.io/design>. Accessed: 2020-03-06.

Monica Anderson, A. P. (2017), 'Tech adoption climbs among older adults'.

**URL:** <https://www.pewresearch.org/internet/2017/05/17/technology-use-among-seniors/>

Myskja, A. (2005), 'Musik som medicin. lyd, musik og terapi'.

NFA (2018), 'Tal og fakta om arbejdsmiljøet'.

**URL:** <http://nfa.dk/-/media/NFA/Arbejdsmiljodata/Fakta-om-Arbejdsmiljo-og-Helbred-2018.ashx?la=da>

POLYUK, S. (2019), 'Age before beauty – a guide to interface design for older adults'.

**URL:** <https://www.toptal.com/designers/ui/ui-design-for-older-adults>

Ridder, H.O. (2017), 'Musikterapi og demens'.

*Snoezel Netværket i Danmark* (2017).

Sundhedsstyrelsen (2019), 'Viden og erfaringer om brug af sansestimuli til at forebygge og reducere udadreagerende adfærd på ældreområdet'.

**URL:** <https://www.sst.dk/-/media/Udgivelser/2019/Sansestimuli.ashx?la=dahash=923339158CFE6B>

WHO - *Dementia* (2019).

**URL:** <https://www.who.int/news-room/fact-sheets/detail/dementia>

## A Visjonsdokument

**Televindu  
Visjonsdokument**

**Versjon <0.2>**

## REVISJONSHISTORIKK

Dato	Versjon	Beskrivelse	Forfatter
<14/februar/2020>	<0.1>	Første utkast	Simen Tengs, Snorre Strand og Erlend Sundøy
<17/februar/2020>	<0.2>	Andre utkast	Simen Tengs, Snorre Strand og Erlend Sundøy

## Innholdsfortegnelse

1 INNLEDNING	4
1.1 REFERANSER	4
2 SAMMENDRAG PROBLEM OG PRODUKT	5
2.1 PROBLEMSAMMENDRAG	5
2.2 PRODUKTSAMMENDRAG	6
3 OVERORDNET BESKRIVELSE AV INTERESSENER OG BRUKERE	7
3.1 OPPSUMMERING INTERESSENER	7
3.2 OPPSUMMERING BRUKERE	8
3.4 SAMMENDRAG AV BRUKERNES BEHOV	9
3.5 ALTERNATIVER TIL VÅRT PRODUKT	10
4 PRODUKTOVERSIKT	10
4.1 PRODUKTETS ROLLE I BRUKERMILJØET	10
4.2 FORUTSETNINGER OG AVHENGIGHETER	10
6 IKKE-FUNKSJONELLE EGENSKAPER OG ANDRE KRAV	11

# 1 INNLEDNING

---

Bachelorgruppen består av tre studenter; Snorre Strand, Erlend Sundøy og Simen Tengs. Dette dokumentet skal beskrive overordnet visjon og plan for bachelorprosjektet i emnet TDAT3001.

Oppgaven går ut på å videreutvikle sakte-tvplattformen Televindu og å prøve å forstå hvordan en slik plattform kan påvirke eldre personer og andre med kognitive utfordringer.

Målet vårt er at resultatproduktet kan brukes i eldre hjem o.l. og blant annet bidra til økt trivsel og høyere aktivitetsgrad.

De tekniske kravene for produktet vil bli konkretisert i et eget kravdokument.

## 1.1 REFERANSER

- [1] Google Analytics: <https://firebase.google.com/docs/analytics>
- [2] Firebase Performance: <https://firebase.google.com/docs/perf-mon>
- [3] Cloud Messaging <https://firebase.google.com/docs/cloud-messaging>
- [4] Crash Reporting: <https://firebase.google.com/products/crashlytics/>
- [5] Flutter: <https://flutter.dev/docs>

## 2 SAMMENDRAG PROBLEM OG PRODUKT

### 2.1 PROBLEMSAMMENDRAG

<b>Problem med</b>	trivsel for beboere hos helseinstitusjoner
<b>berører</b>	eldre personer med demens og andre kognitive utfordringer, samt menneskene rundt dem
<b>som resultatet av dette</b>	kan hverdagen til disse personene bli kjedelig og monoton, og de kan føle seg ensomme.
<b>En vellykket løsning vil</b>	stimulere sansene deres etter hver enkelts behov og bidra til økt trivsel og helse.



## 2.2 PRODUKTSAMMENDRAG

<b>For</b>	helseinstitusjoner
<b>som</b>	har behov for underholdning/ målrettet sansestimulering for sine beboere.
<b>Televindu</b>	er en videoplattform for sakte-tv
<b>som</b>	tilbyr sansestimulerende innhold tilpasset målgruppen og deres varierende behov.
<b>I motsetning til</b>	eksisterende video plattformer som ikke har tilpasset innhold for eldre og personer med kognitive utfordringer
<b>Har vårt produkt</b>	et mer tilpasset og brukervennlig innhold.

## 3 OVERORDNET BESKRIVELSE AV INTERESSENER OG BRUKERE

---

### 3.1 OPPSUMMERING INTERESSEENTER

Navn	Beskrivelse	Rolle
Statlige organer	Kommuner og statlige organisasjoner som huser eldre og andre med kognitive utfordringer	Økonomisk støtte og investering. Finne investorer (sykehjem / aldershjem). Registrerte brukere.
Beboere på helseinstitusjon	Sluttbruker av systemet	Bruke systemet og gi tilbakemeldinger og forbedringsforslag
Pleier på helseinstitusjon	Sluttbruker av systemet	Administrere tilpassede spillelister for beboerne og gi tilbakemeldinger
Televindu	Administrator og eier av systemet	Administrasjon og vedlikehold

### 3.2 OPPSUMMERING BRUKERE

Navn	Utdypende beskrivende	Rolle under utviklingen	Representert av
------	-----------------------	-------------------------	-----------------

Beboere på helseinstitusjon	En av personene i målgruppen vår som bor på en helseinstitusjon, ser på sakte-tv.	Bruke systemet og eventuelt gi tilbakemeldinger og forbedringsforslag	Selvrepresentert
Pleier på helseinstitusjon	Pleierne på helseinstitusjonene som har ansvar for pasientens velvære	Administrere tilpassede spillelister for beboerne og gi tilbakemeldinger	Helseinstitusjon

### 3.3 BRUKERMILJØET

Applikasjonen skal i hovedsak kjøres på smarttelefoner, tv og/eller nettbrett i helseinstitusjoner. Det er ofte passende å bruke denne applikasjonen etter ønske/behov i fellesstue / private rom på institusjonen. Dette er fordi applikasjonens innhold er rolig og saktegående, og kan vises med varierende støynivåer i omgivelsene. Det er i hovedsak underholdning. Derfor kreves det ikke aktiv styring fra brukerne selv og man kan derfor også la applikasjonen kjøre i bakgrunnen.

### 3.4 SAMMENDRAG AV BRUKERNES BEHOV

Behov	Prioritet	Vedrører	Dagens løsning	Foreslått løsning
-------	-----------	----------	----------------	-------------------

Bruke appen på iPhone og iPad	Høy	Applikasjonen	Ingen	Bruke en iMac til å bruke xcode og deploye appen på iOS.
Laste ned en spilleliste	Høy	Applikasjonen	Ingen	Bruke HIVE for å laste ned alle videoene i en spilleliste.
Se hvilke videoer man har sett	Middels	Databasesystemet	Ingen	Logge videoene brukeren har sett i databasen.
Se videoer som er i filmet nærheten	Middels	Databasen og Applikasjonen, GPS	Ingen	Bruke location tracking og geohash for å få de nærmeste videoene.
Bruke Google Cast for å se på en TV	Middels	Google Chromecast	Ingen	Bruke fluttter_cast sin plugin.
Se foreslåtte videoer	Middels	Applikasjonen	Ingen	Lagre brukerens mest sette tags i databasen, og gi brukeren videoer som matcher de taggene.

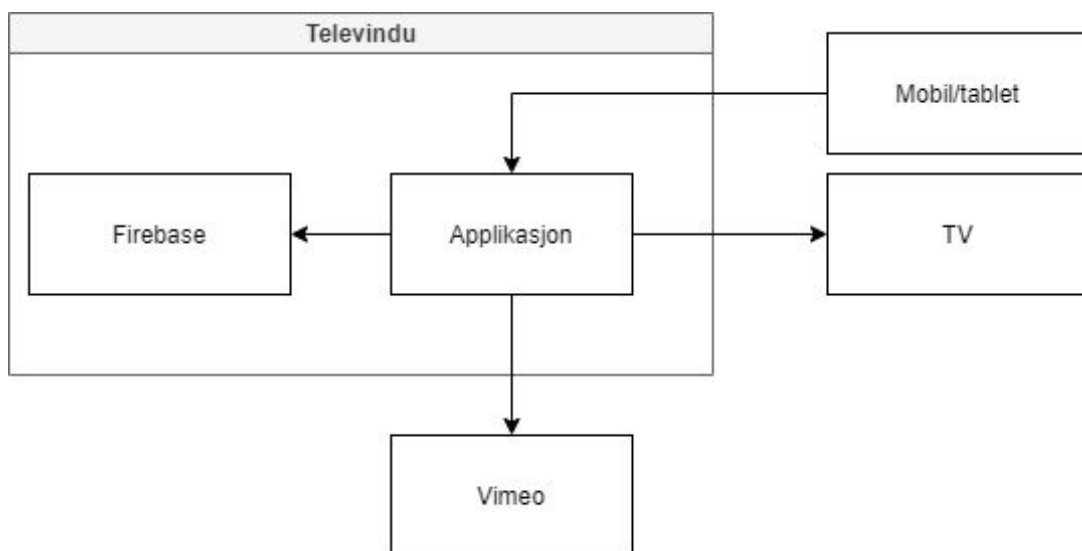
### 3.5 ALTERNATIVER TIL VÅRT PRODUKT

Netflix, NRK, HBO, Amazon Video, Viaplay, Hulu.

## 4 PRODUKTOVERSIKT

---

### 4.1 PRODUKTETS ROLLE I BRUKERMILJØET



### 4.2 FORUTSETNINGER OG AVHENGIGHETER

Det forutsettes at videoene i applikasjonen skal være tilpasset til eldre og personer med kognitive utfordringer, og videoene skal ha en grad fra 0-8 av hvor mye stimuli videoen “produserer”.

Det skal være mulig å laste ned videoer, siden de fleste fellesområdene i helseinstitusjoner har dårlig tilgang til internett.

Det forutsettes også at man må inngå en avtale med Televindu før man kan registrere en business bruker for kunden.

## 5 PRODUKTETS FUNKSJONELLE EGENSKAPER

Tabellen nedenfor viser et førsteutkast til funksjonelle egenskaper i videreutvikling av applikasjonen. Prosjektgruppen kan gjøre endringer etter avtale med oppdragsgiver.

Funksjonelle egenskaper
Lage applikasjonen for tablet.
Lage applikasjonen for iPhone og iPad.
Funksjonalitet for å laste ned spillelister.
Få videoer i nærheten av deg med location tracking.
Støtte for Google Chromecast.
Historie for brukere.
Foreslåtte videoer (viser videoforslag basert på de tagsa brukeren har sett mest).
Integrere Google Analytics [1]: forteller oss hvem som bruker appen og hvordan de bruker appen.
Integrere Firebase Performance [2]: får data om hvor rask appen er i ulike use cases.
Integrere Cloud Messaging [3] for Flutter: f.eks.: mobilen viser notifikasjon når en ny video har blitt lastet opp.
Crash Reporting [4]: for å prioritere ulike bugs og crashes som skjer i produksjon.

## 6 IKKE-FUNKSJONELLE EGENSKAPER OG ANDRE KRAV

Televindu har valgt Flutter [5] som mobil-rammeverk. Det er et krav at dette rammeverket skal brukes i dette prosjektet.

Løsningen skal ha god brukskvalitet og høy brukervennlighet; spesielt for eldre, personer med kognitive utfordringer og ansatte i sykehus og eldreheim. Det skal gjennomføres brukbarhetstesting for å sjekke at applikasjonen møter kravene om brukskvalitet.

Løsningen skal være responsiv utformet, og tilby samme grad av brukervennlighet på alle støttede enheter.

## B Kravdokumentasjon



---

**44**

**Televindu  
Kravdokumentasjon**

**Versjon 0.3**

## Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
20.februar 2020	0.1	Første utkast	Snorre Strand
24.mars 2020	0.2	Andre utkast	Snorre Strand, Erlend Sundøy og Simen Tengs
10.april 2020	0.3	Satte inn nye krav fra Televindu	Snorre Strand, Erlend Sundøy og Simen Tengs

## Innhold

Introduksjon	3
User Stories	3
Applikasjonen for iOS	3
Applikasjon for nettbrett	3
Funksjonalitet for å laste ned spillelister	4
Foreslå videoer i nærheten.	4
Støtte for Google Chromecast	5
Foreslåtte videoer basert på tags	5
Brukerhistorikk	6
Google Analytics	6
Cloud Messaging	7
Firebase Performance	7
Crash Reporting	7
Onboarding	8
Fargekode på videoer	8
Televindu type ikoner	9
Brukerspilleliste funksjon	9
Legg til bruker spilleliste funksjon	10
Søkefunksjon	10
Airplay	11
Modeller	12
Hoved Views i Flutter treet	12
Klasse diagram	13
Televindu miljø	14

## 1. Introduksjon

Dette er kravdokumentasjonen til Bachelor oppgaven nr 44 Televindu.

I kravdokumentasjonen lister vi opp kravene til videreutviklingen av Televindu applikasjonen. Kravspesifikasjonene er avledet av det gitte visjonsdokumentet og gått gjennom med oppdragsgiveren. Dokumentet inneholder en rekke med user stories og en modell av hvordan Flutter widgetsene kommuniserer med hverandre.

## 2. User Stories

### 2.1 Applikasjonen for iOS

Som privat bruker og pleier Ønsker jeg at applikasjonen er tilgjengelig for iOS Slik at jeg kan bruke applikasjonen på iOS enheter
--

**Scenario:** Bruke appen på en iOS enhet

Gitt at jeg er bruker

Når jeg bruker appen på en iOS enhet

Så skal de samme funksjonalitetene på Android applikasjonen være tilgjengelig og UI'en skal være tilpasset iOS.

**Akseptansekrav:**

- Nedlastingsfunksjon skal fungere på iOS.
- Videospiller skal være tilpasset til iOS.
- UI skal være tilpasset til iOS.

### 2.2 Applikasjon for nettbrett

Som privat bruker og pleier Ønsker jeg å bruke applikasjonen på nettbrett Slik at jeg kan se videoer på større skjermer
---

**Scenario:** Bruke appen på et nettbrett

Gitt at jeg er en bruker

Når jeg bruker appen

Så skal UI'en være tilpasset til alle størrelser på ulike nettbrett.

**Akseptansekrav:**

- Navbar, font størrelsen, thumbnails og knapper skal være tilpasset nettbrett.
- Videospilleren skal være tilpasset nettbrett.

**2.3 Funksjonalitet for å laste ned spillelister**

Som privat bruker og pleier  
Ønsker jeg å laste ned spillelister  
Slik at jeg kan se på spillelistene uten tilgang til internett.

**Scenario:** Laste ned en spilleliste

Gitt at jeg er en bruker

Når jeg er i playlist detail screen og trykker på «Last ned spilleliste»

Så skal jeg få notifikasjon om at videoene blir lastet ned og se progress i download screen.

**Akseptansekrav:**

- Alle videoene i spillelisten skal settes i nedlastingskøen i download service.
- Det skal vises en push notification i bakgrunn i appen som sier framgangen av alle videoer som blir lastet ned.
- Når spillelisten er lastet ned skal man kunne se spillelisten i download screen, og se på spillelisten uten tilgang til internett.

**2.4 Foreslå videoer i nærheten.**

Som en bruker  
Ønsker jeg å få videoer som har blitt filmet i nærheten av meg  
Slik at jeg kan se på videoer av steder jeg kjenner til.

**Scenario:** Bruke location tracking

Gitt at jeg er en bruker, har tillatt at applikasjonen kan bruke min lokasjon og at det er noen videoer som har blitt filmet i nærheten

Når jeg er på hjemmesiden

Så skal de videoene som har blitt filmet i nærheten vises

**Akseptansekrav:**

- Når brukeren først har logget inn skal applikasjonen be om tillatelse til å bruke enhetens posisjon.
- Hvis applikasjonen ikke har tilgang til enhetens posisjon eller det ikke er noen

videoer innen en gitt radius skal ikke denne widgeten vises.

- Videoene som er i nærheten skal vises på samme måte som alle andre carousel sliders i applikasjonen.
- Man skal ha tilgang til å trykke «view all» oppe i høyre hjørnet av widgeten, som viser alle videoene i nærheten som en scrollbar liste.

## 2.5 Støtte for Google Chromecast

Som en bruker  
Ønsker jeg bruke Google Chromecast funksjonalitet  
Slik at jeg kan se på videoer på TV'er

### **Scenario:** Google Chromecast

Gitt at jeg er bruker, og har tilgang til en chromecast

Når jeg er på «video detail screen» og trykker på knappen i høyre hjørne i appbar

Så skal en liste av tilgjengelige Google Cast enheter vises.

### **Akseptansekrav:**

- Det skal være en casting ikon knapp i appbar på video detail screen som viser tilgjengelige casting enheter.
- Etter man har valgt en kasting enhet så skal man vise en chromecast control screen hvor man kan pause videoen, styre volum og kansellere casting.
- Når man caster en video skal casting ikonet være markert og når man trykker på ikonet igjen skal man gå til chromecast control screen.

## 2.6 Foreslåtte videoer basert på tags

Som en bruker  
Ønsker jeg å få foreslåtte videoer basert på hvilken tags jeg har sett mest på  
Slik at jeg kan lett finne nye videoer som jeg liker.

### **Scenario:** Foreslåtte videoer

Gitt at jeg har sett på tidligere videoer

Når jeg er hjemmesiden

Så skal jeg se foreslåtte videoer basert på tags.

### **Akseptansekrav:**

- Hver gang en bruker har sett ferdig en video skal alle tagene som er i videoen bli inkrementert i brukeren sitt dokument i firestore databasen.
- Videoer som matcher de tagene i bruker databasen best skal vises på home screen som en Carousel Slider med «view all» knapp.

- Hvis brukeren ikke har sett noen videoer eller det er ingen matches så skal man ikke vise noen foreslåtte videoer basert på tags.

## 2.7 Brukerhistorikk

Som en bruker  
Ønsker jeg å se hvilke videoer jeg har sett før  
Slik at jeg kan lettere få oversikt over nye videoer eller finne fram videoer jeg har sett før

### **Scenario:** Bruker historikk

Gitt at jeg er bruker, og har sett en eller flere videoer fra før av

Når jeg er på home screen

Så skal videoer som jeg har sett fra før av merkes

### **Akseptansekrav:**

- Hver gang en bruker har sett ferdig en video, skal video iden, tiden brukeren så videoen og hvor langt brukeren så videoen lagres i brukeren sitt dokument i firestore databasen.
- Det skal være mulig å slette brukerhistorikk i innstillinger.
- Det skal være mulig å fortsette videoen der brukeren var sist i en video.
- Det skal markeres på de videoene brukeren har sett ferdig.

## 2.8 Google Analytics

Som admin bruker  
Ønsker jeg å få anonym data av brukere  
Slik at jeg kan analysere

### **Scenario:** Google Analytics

Gitt at jeg er en admin bruker

Når jeg er firebase nettsiden

Så skal jeg se anonym data av brukerne.

### **Akseptansekrav:**

- Det skal lagre data om hvor mange Android og iOS enheter som bruker appen.
- I admin nettsiden skal det vises hvor mange aktive brukere det er hver dag, hvilket land brukerne kommer fra og hvor lenge brukeren bruker appen.

## 2.9 Cloud Messaging

Som pleier og privat bruker  
Ønsker jeg å bli notifisert når en ny video har blitt lastet opp  
Slik at jeg kan bli oppdatert av nye videoer

**Scenario:** Cloud Messaging

Gitt at jeg er bruker

Når jeg har strøm og internett på mobilen og ikke bruker appen

Så skal jeg få push notification når en ny video har blitt lastet opp.

**Akseptansekrav:**

- Man kan velge å skru av og på denne funksjonaliteten i innstillinger.
- Push notification skal være lydløs.
- Teksten skal norsk, dansk eller engelsk avhengig av brukerens språk.

## 2.10 Firebase Performance

Som admin bruker  
Ønsker jeg å få anonym data av brukere  
Slik at jeg kan analysere

**Scenario:** Firebase Performance

Gitt at jeg er en admin bruker

Når jeg er firebase nettsiden

Så skal jeg se anonym data av brukerne.

**Akseptansekrav:**

- I admin nettsiden skal man se hvor lang tid appen tar for å starte.
- Man skal se performansen basert på enheter, software versjon og land.

## 2.11 Crash Reporting

Som admin bruker og utvikler  
Ønsker jeg å hvor ofte appen krasjer i produksjon  
og hvor i applikasjonen appen krasjer  
Slik at jeg fikse og prioritere bugs

**Scenario:** Crash Reporting

Gitt at jeg er en admin bruker

Når jeg er i admin nettsiden

Så skal jeg få overblikk over alle bugs og krasjer som skjer i produksjon.

**Akseptansekrav:**

- I admin nettsiden skal man se en stack trace av alle krasjene som skjer i produksjon.

## 2.12 Onboarding

Som en bruker

Ønsker jeg å overblikk over applikasjonen

Slik at jeg lettere kan forstå hvordan man bruker appen

**Scenario:** Onboarding

Gitt at jeg er bruker

Når jeg åpner appen for første gang

Så skal jeg se onboarding skjermen.

**Akseptansekrav:**

- Når man åpner appen for første gang skal onboarding screenen vises.
- Man kan ha tilgang til å se onboarding screen med å trykke på spørsmål ikonet i app baren.
- Onboarding skal ha en knapp til å "skippe" onboarding slidene.
- Slidene i onboarding skal dekke home screen, playlist, download, og settings.

## 2.13 Fargekode på videoer

Som en bruker

Ønsker jeg å kunne se fargekode som representerer mengde sansestimuli på videoene

Slik at jeg kan se hvor mye stimuli en video inneholder

**Scenario:** Fargekode på videoer

Gitt at jeg er bruker

Når jeg scroller gjennom videoene

Så skal jeg få se en grønn, gul eller oransje sirkel oppe til venstre i hver thumbnail.

**Akseptansekrav:**

- Hver thumbnail skal inneholde enten grønn (lav stimuli), gul (middels stimuli) eller



orange (høy stimuli).

## 2.14 Televindu type ikoner

Som en bruker  
Ønsker jeg å lettere å se hvilken type en video er  
Slik at jeg lettere kan forstå hva innholdet i en video er

**Scenario:** Televindu type ikoner  
Gitt at jeg er bruker  
Når jeg scroller gjennom videoene  
Så skal jeg få se en ikon som tilsvarer kategorien til videoene.

### Akseptansekrav:

- Hver video skal ha en tilsvarende Televindu ikon som sier hvilken type videoen er i thumbnail.

## 2.15 Brukerspilleliste funksjon

Som en bruker  
Ønsker jeg å kunne lage og redigere min egen brukerspilleliste  
Slik at jeg kan lettere organiserer videoene jeg vil se.

**Scenario:** Navigere til lag brukerspilleliste  
Gitt at jeg er bruker  
Når jeg er på playlist screen  
Så skal det være to tabs, televindu og mine spillelister og i mine spillelister skal det være en FAB som jeg kan trykke på for å lage en ny spilleliste.

**Scenario:** Lag brukerspilleliste  
Gitt at jeg er bruker  
Når jeg er i lag brukerspilleliste screen  
Så skal jeg kunne skrive tittelen til spillelisten og velge hvilken videoer som skal være i spillelisten.

**Scenario:** Rediger spilleliste  
Gitt at jeg er bruker  
Når jeg er i playlist detail screen i min egen brukerspilleliste  
Så skal jeg kunne redigere tittelen, hvilken videoer som er i spillelisten, rekkefølgen på

videoene og kunne slette spillelisten.

**Akseptansekrav:**

- Det skal være en FAB med et “add” ikon i my playlist tab.
- Etter man har trykket på knappen skal det det være en textfield til tittelen til spillelisten, en liste av alle videoer med en checkbox, en knapp til å lage spillelisten og en knapp til å kansellere.
- Hvis man trykker på “view all” i en brukerspillelsite skal en FAB med “edit” ikon vises.
- Hvis man trykker på knappen så skal det vises en texfield der man kan redigere tittelen, liste med videoene i spillelisten, en knapp til å lage endringer og en knapp til å slette videoen, og en knapp til å kansellere.

**2.16 Legg til bruker spilleliste funksjon**

Som en bruker  
Ønsker jeg å kunne legge en video til min spilleliste direkte fra video screen  
Slik at jeg lettere kan legge videoer til spillelisten min

**Scenario:** Legg til spilleliste funksjon

Gitt at jeg er bruker

Når jeg er på video detail screen

Så skal jeg ha mulighet til å adde videoen i en av mine eksisterende spilleliste, eller lage en ny spilleliste.

**Akseptansekrav:**

- Det skal være en add til spilleliste knapp i video detail page.
- Etter man har trykket på knappen skal en liste av brukerens spilleliste komme opp, og en textfield nederst der brukeren kan lage en ny spilleliste.

**2.17 Søkefunksjon**

Som en bruker  
Ønsker jeg å kunne søke etter videoer  
Slik at jeg lettere kan finne igjen videoer jeg har sett tidligere.

**Scenario:** Søkefunksjon

Gitt at jeg er bruker

Når jeg er på home screen

Så skal jeg ha mulighet til å trykke på en search ikon for å søke etter videoer.

**Akseptansekrav:**

- Det skal være en "seach" ikon i home screen, som navigerer til en search screen.
- I search screen kan man søke etter titlene til videoene og hvis man trykker på en video vil man komme til video detail screen.

**2.18 Airplay**

Som en bruker og bruker en iOS enhet  
Ønsker jeg å kunne bruke Airplay når jeg ser en video  
Slik at jeg kan hoste en video på en TV

**Scenario: Airplay**

Gitt at jeg er bruker og bruker en iOS enhet

Når jeg er på video detail screen og går til fullskjerm

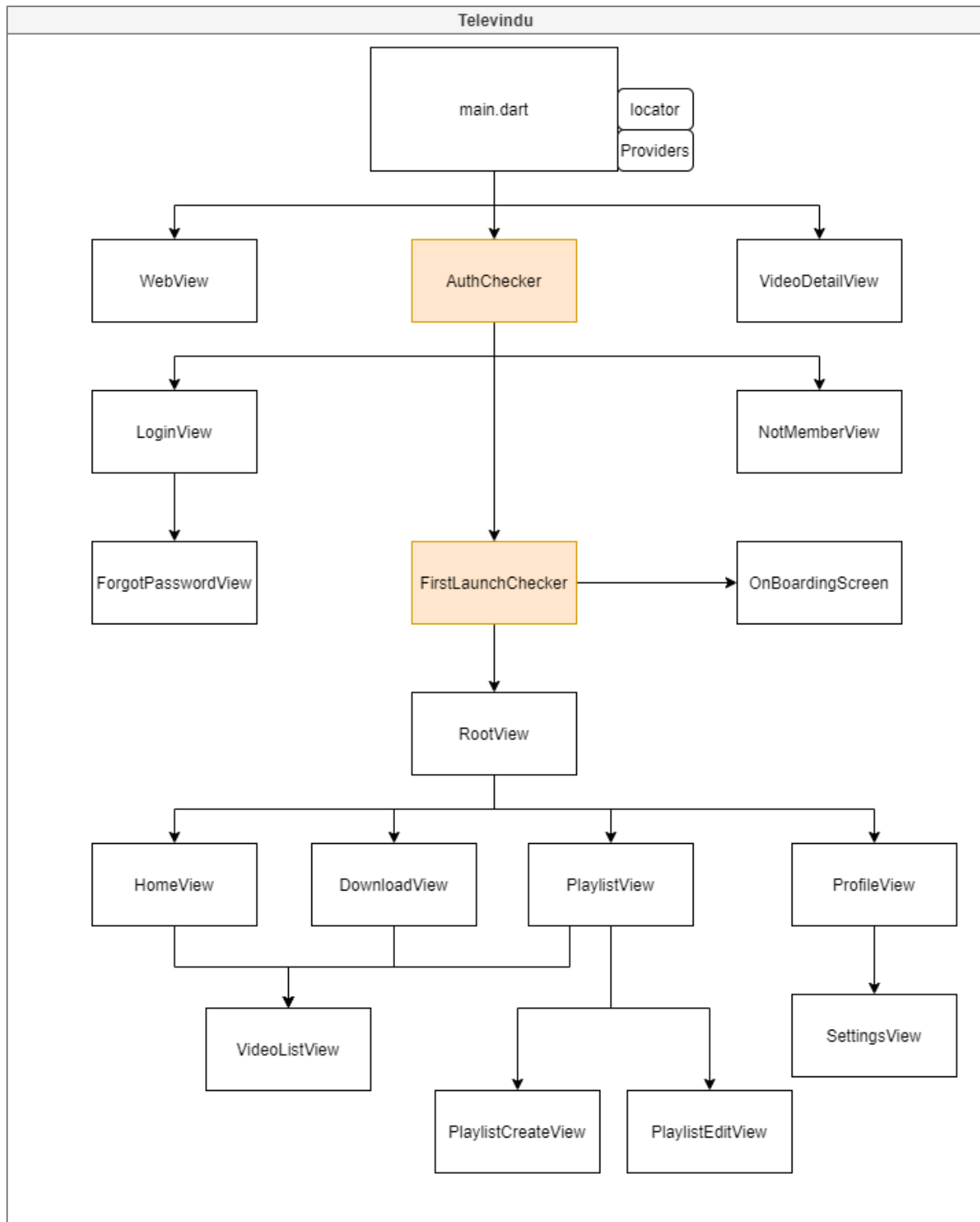
Så skal iOS sin AVPlayer spilles av i fullskjerm der man kan bruke Airplay.

**Akseptansekrav:**

- AVPlayer og Flutter sin video\_player skal kommunisere positionen man er i videoen slik at overgangen er brukervennlig.
- Hvis en video er lastet ned skal AVPlayeren finne den nedlastede videoen.
- Hvis man har sett ferdig videoen i AVPlayer så skal man gå direkte til neste video i fullskjerm.

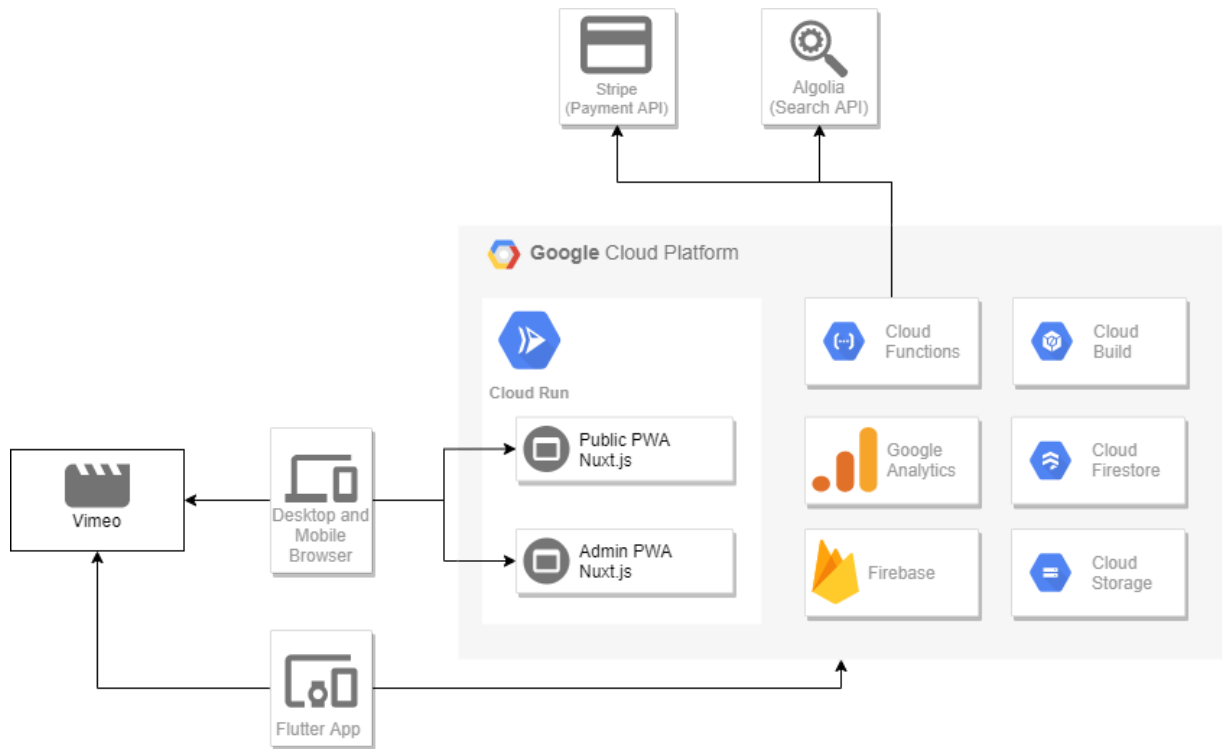
### 3. Modeller

#### 3.1 Hoved Views i Flutter treet





### 3.3 Televindu miljø



## C Scrum

# Scrum Proses

## Contents

Sprint 0: Uke 6 - 8 (3. februar til 23. februar)	<b>2</b>
Sammendrag	2
Sprint 1: Uke 9 - 10 (24. februar til 10. mars)	<b>2</b>
Sammendrag	2
Issue board	3
Sprint 2: Uke 11-12 (9. mars til 22. mars)	<b>3</b>
Sammendrag	3
Issue board	4
Retrospektiv	4
Sprint 3: Uke 13 - 14 (23. mars til 5. april)	<b>5</b>
Sammendrag	5
Issue board	5
Sprint 4: Uke 15 - 16 (6. april til 19. april)	<b>5</b>
Sammendrag	5
Issue board	6
Retrospektiv	6
Sprint 5: Uke 17 - 18 (20. april til 3. mai)	<b>7</b>
Sammendrag	7
Issue board	7
Sprint 6: Uke 19 - 20 (4. mai til 20. mai)	<b>8</b>
Sammendrag	8
Issue board	8



## Sprint 0: Uke 6 - 8 (3. februar til 23. februar)

### Sammendrag

- Kurset i Dart og Flutter
- Ferdigstilte visjonsdokument
- Satt opp førsteutkast av kravdokument
- Lagde Framdriftsplan
- Utforsket reglement for innsamling av pasientdata
- Begynte å implementere Onboarding/Hjelp funksjon
- Fikset bugs i videospilleren

Vi begynte med prosjektet 3. februar. I startfasen var det mye planlegging og organisering som måtte gjøres. For ryddighetens skyld har valgt å kalle de første ukene for sprint 0, som var før vi begynte med å faktisk ta i bruk scrum prosess. Den første uka gikk til å ha diverse møter, blant annet med oppdragsgiver og veileder, samt påbegynne visjonsdokument og kravdokument. Det gikk også relativt mye tid til å lete etter kontorplasser, da oppdragsgiver hadde en del alternativer. Vi endte opp med ikke å bruke noen av disse.

## Sprint 1: Uke 9 - 10 (24. februar til 10. mars)

### Sammendrag

- Fortsatte kursing i Flutter og Dart
- Satte oss inn i den eksisterende arkitekturen
- Konfigurerte applikasjonen for iOS
- Satte opp Firebase App Distribution for iOS og Android
- Tilpasse brukergrensesnittet for nettbrett
- Begynte å implementere Google Cast
- Begynte å implementere Firebase Analytics
- Fikset bugs av nedlastninger av videoer
- Ble ferdig med Onboarding

# Issue board

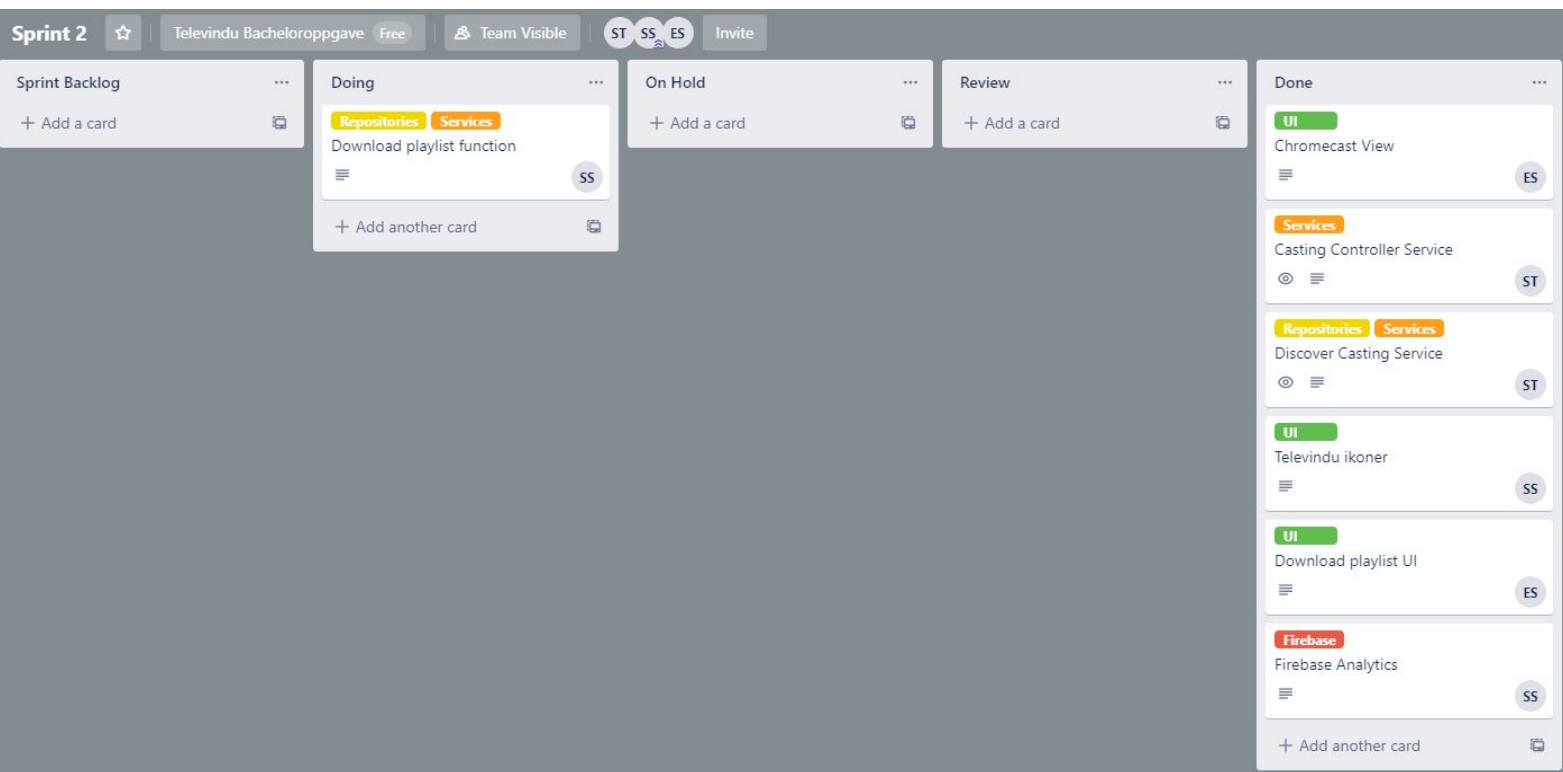
The screenshot shows a Jira issue board for 'Sprint 1'. The board is organized into columns: 'Sprint Backlog', 'Doing', 'On Hold', 'Review', and 'Done'. The 'Doing' column contains two cards: 'Firebase Analytics' (with a 'SS' status) and 'Discover Casting Service' (with a 'ST' status). The 'Done' column contains five cards: 'iOS konfigurering' (with 'UI' and 'iOS' labels and 'ES' status), 'Firebase appdistribusjon for iOS' (with 'Firebase' and 'iOS' labels and 'ES' status), 'Firebase appdistribusjon for Android' (with 'Firebase' and 'Android' labels and 'ST' status), 'Download bugfix' (with 'Services' label and 'SS' status), and 'Tilpasse UI for Tablets' (with 'UI' label and 'SS' status). The 'Onboarding' card at the bottom of the 'Done' column has 'UI' label and 'ES' and 'ST' statuses. The board also shows a 'Sprint Backlog' column with '+ Add a card' and a 'Review' column with '+ Add a card'. The top of the board shows 'Sprint 1' with a star icon, 'Televindu Bacheloroppgave' with 'Free' status, 'Team Visible' icon, and 'ST', 'SS', 'ES' status indicators and an 'Invite' button.

## Sprint 2: Uke 11-12 (9. mars til 22. mars)

### Sammendrag

- Økonomi eksamen og korona tok mye av tiden i denne sprinten.
- Implementere ferdig Google Cast.
- Implementerte Televindu designet ikoner for video types.
- Startet å utvikle funksjonalitet for å laste ned spillelister.
- Ble ferdig med å integrere Firebase Analytics.

## Issue board



## Retrospektiv

### Hva gikk bra:

- Alle møtte opp til planlagt tid
- Vi har vært flinke på standup møter
- Bra samarbeid mellom gruppemedlemmene.
- Bra arbeidsflyt
- Bra fordeling av arbeidsoppgaver

### Hva gikk ikke like bra:

- Selv om vi hadde bra kommunikasjon kunne vi holdt trello mer oppdatert
- Lite dokumentasjon av kode
- Brukte en del tid på å finne kontor plass, endte opp på biblioteket uansett

### Hva vil vi forbedre til neste sprint:

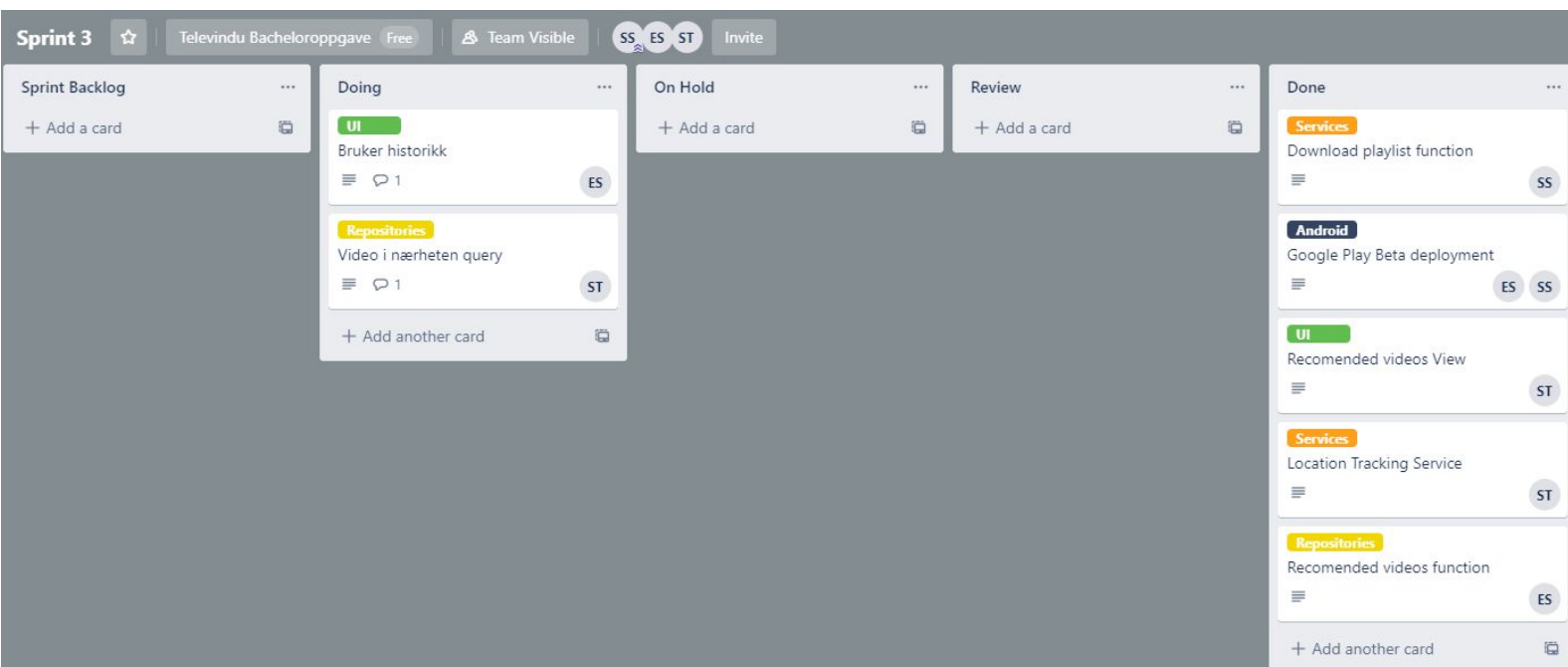
- Holde Trello mer oppdatert
- Bør bli bedre på dokumentasjon
- Bedre merge reviews
- Automatisering av distribusjon

## Sprint 3: Uke 13 - 14 (23. mars til 5. april)

### Sammendrag

- Begynte å utvikle brukerhistorikk.
- Ferdig med laste ned spilleliste funksjon.
- Begynte å utvikle lokasjonsbaserte videoer funksjonalitet.
- Implementerte anbefalte videoer
- Publiserte en betaversjon av appen på Google Play Store. Televindu ville at den skulle bli tilgjengelig på Android tidligere enn forventet på grunn av korona, og den tidlige iOS versjonen var allerede publisert.

### Issue board

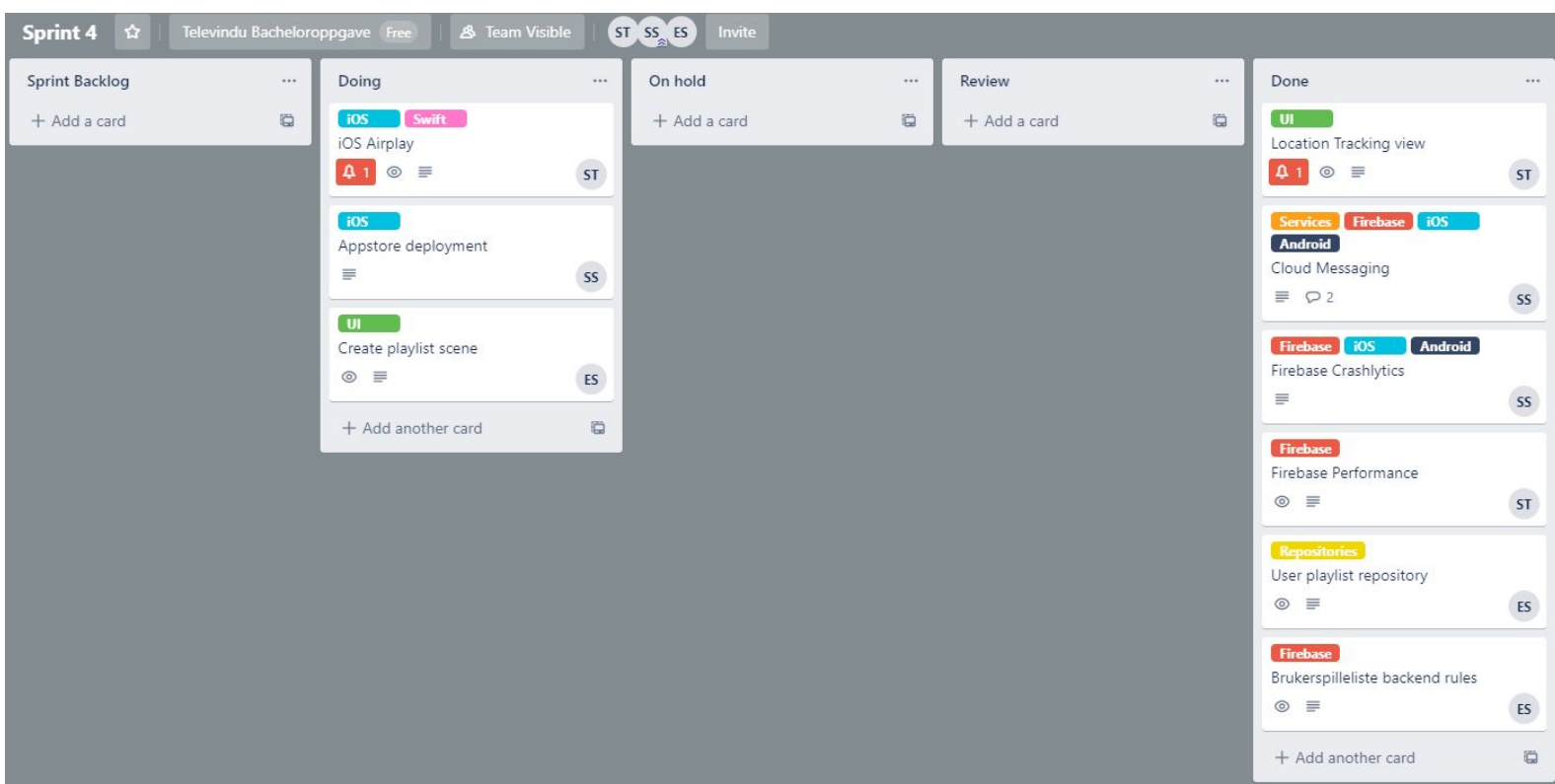


## Sprint 4: Uke 15 - 16 (6. april til 19. april)

### Sammendrag

- Begynte å implementere Airplay for iOS.
- Ble ferdig med lokasjonsbaserte videoer.
- Integreerte Firebase Performance Monitoring.
- Integreerte Firebase Crashlytics.
- Integreerte Cloud Messaging.
- Har holdt på med å utvikle brukerspillelister.
- Hadde flere møter med oppdragsgiver og veileder.

## Issue board



### Retrospektiv

Hva gikk bra:

- Fortsatt bra på standups
- Trello var aktivt oppdatert
- Vi var flinke med å hjelpe hverandre når noen satt fast
- Vi ble bedre på å dokumentere kode

Hva gikk ikke like bra:

- Jobbing i fellesskap over Discord gikk ikke like bra som da vi kunne jobbe på samme arbeidsplass
- Algolia funka ikke så nice som vi trodde
- Ble ikke ferdig med Appstore deployment og iOS Airplay.

Hva vil vi forbedre:

- Bedre samhandling på Discord Voice
- Bedre kommunikasjon med oppdragsgiver

## Sprint 5: Uke 17 - 18 (20. april til 3. mai)

### Sammendrag

- Ble ferdig med brukerhistorikk.
- Bugfikning & generelle forbedringer.
- Implementerte søkerfunksjon.
- Ble ferdig med å utvikle brukerspillelister.
- Ble ferdig med utvikling av Airplay for iOS.

### Issue board

The screenshot shows a Jira issue board for 'Sprint 5'. The board is organized into columns: Sprint Backlog, Doing, Review, On Hold, and Done. The 'Done' column is the most populated, containing several completed tasks. Each task card includes a title, a list of labels (e.g., iOS, Swift, Firebase, UI), and a status icon (SS, ES, ST). The tasks listed in the 'Done' column are: iOS Airplay (iOS, Swift, SS), VideoPlayer PlatformChannel (iOS, Swift, SS), History backend rules (Firebase, ES), Edit playlist scene (UI, SS), History repository (Repositories, ES), Video seach Service (Services, ST), Video search View (UI, ST), and Userhistory UI (UI, ES, SS). The board also features a 'Sprint Backlog' column with an 'Add a card' button, and 'On Hold' and 'Review' columns with one task each. The top of the board shows the sprint name 'Sprint 5', a star icon, the project name 'Televindu Bacheloroppgave', the status 'Free', 'Team Visible', and user avatars for SS, ES, and ST, along with an 'Invite' button.

## Sprint 6: Uke 19 - 20 (4. mai til 20. mai)

### Sammendrag

- Ferdig med deployment på Google Play Store & Apple App Store
- Implementerte fargeikoner for sansestimuliskala av videoer.
- Ferdigstilling av hovedrapport
- Skrev brukerveiledning
- Skrev installasjonsmanual

### Issue board

**Sprint 6** ☆ Televidu Bacheloroppgave Free Team Visible SS ES ST Invite

**Sprint Backlog** ... + Add a card

**Doing** ... + Add a card

**On Hold** ... + Add a card

**Review** ... + Add a card

**Done** ...

- UI** Fargeikoner på videoer ES ST 4
- Android** Google Play deployment SS
- iOS** Appstore deployment SS

+ Add another card

## D Prosjekthåndbok



# Prosjekthåndbok

## Bacheloroppgave - Televindu

<b>Framdriftsplan</b>	<b>2</b>
<b>Møteinnkalling</b>	<b>3</b>
Møteinnkalling 01 - Oppstartsmøte (04.02.20)	3
Møteinnkalling 02 - Visjons- og kravdokument (20.02.2020)	4
Møteinnkalling 03 - Datainnsamling (16.04.2020)	5
<b>Møtereferat</b>	<b>6</b>
Møtereferat 01 - Oppstartsmøte (04.02.20)	6
Møtereferat 02 - Visjons- og kravdokument (20.02.2020)	7
Møtereferat 03 - Datainnsamling (16.04.2020)	8
<b>Timelister</b>	<b>9</b>
Sammendrag	9
Rapport fra Clockify	11

# Framdriftsplan

## **Estimert ferdig 1.mars:**

1. Visjonsdokument.
2. Første utkast av Kravdokumentet.
3. Utforsk reglement for innsamling av data.
4. Kursing og gjøre oss kjent med Flutter og Dart.
5. Konfigurere applikasjonen for iOS.
6. Firebase appdistribusjon for iOS
7. Onboarding.
8. Firebase Analytics.

## **Estimert ferdig 1.april:**

9. Tilpasse brukergrensesnitt for Tablets
10. Funksjonalitet for å laste ned spilleliste
11. Historie for brukere
12. Anbefalte videoer
13. Støtte for Google Cast

## **Estimert ferdig 1.mai:**

14. Integre Firebase Performance Monitoring
15. Firebase Crashlytics
16. Cloud Messaging
17. Få videoer i nærheten av deg med location tracking
18. Teste brukerapplikasjonen på sykehjem i Moss (**fikk ikke til**)

## **Estimert ferdig 20.mai:**

19. Google Play deployment
20. App Store deployment

## **Gjennomgående i hele prosjektet:**

- Research om problemstillingen
- Hovedrapport
- Dokumentasjon av kode
- Code Reviews, Standups, Retrospectives & andre smidige prosesser
- Integrer Responsiv Design i eksisterende kode og i ny kode

# Møteinnkalling

## Møteinnkalling 01 - Oppstartsmøte (04.02.20)

Tidspunkt/sted: Tirsdag 04.02.20 kl 14.30 – 15.15, IT-syd

Følgende personer innkalles:

Simen Tengs

Snorre Strand

Erlend Sundøy

Alexander Holt (veileder)

Øystein Hansen (prosjektgiver)

### Agenda:

Sak nr 01 Drøfte prosjektet

Sak nr 02 Krav til dokumentasjon

Sak nr 03 Retningslinjer for vurdering

Sak nr 04 Ambisjonsnivå

Sak nr 05 Signere avtale

Møtet planlegges avsluttet ca kl. 11.00

Mvh

Gruppe 44

Trondheim 04.02.2020

## Møteinnkalling 02 - Visjons- og kravdokument (20.02.2020)

Følgende personer innkalles:

Simen Tengs

Snorre Strand

Erlend Sundøy

Alexander Holt (veileder)

### Agenda:

Sak nr 01      Visjonsdokument

Sak nr 02      Framdriftsplan

Sak nr 03      Timelister

Sak nr 04      Diverse spørsmål

- Problemstilling spesifikk til Televindu
- Innsamling av brukerdata (anonym / personlig)

Møtet planlegges avsluttet ca kl. 12:00

Mvh

Gruppe 44

Trondheim 17.02.2020

## Møteinnkalling 03 - Datainnsamling (16.04.2020)

Følgende personer innkalles:

Simen Tengs

Snorre Strand

Erlend Sundøy

Alexander Holt (veileder)

### Agenda:

- |           |   |
|-----------|---|
| Sak nr 01 | Datainnsamling / intervjuer / forskning / problemstilling påvirket av korona  |
| Sak nr 02 | Spørsmål om hovedrapport  |
| Sak nr 04 | Diverse saker <ul style="list-style-type: none"><li>· Presentasjon etter levering</li><li>· Oppgaveteksten</li><li>· Visjonsdokument / Kravdokument</li></ul> |

Møtet planlegges avsluttet ca kl. 12:45

Mvh

Gruppe 44

Trondheim 16.04.2020

# Møtereferat

## Møtereferat 01 - Oppstartsmøte (04.02.20)

### Dokumentasjon:

- Prosjektet skal hovedsakelig bli dokumentert i disse dokumentene: Hovedrapport, prosjekthåndbok, visjonsdokument, fremdriftsplan.
- Teknisk dokumentasjon: Vi skal dokumentere det som føles naturlig av selve koden. Vi skal også lage en intuitiv brukermanual og ellers bruke de diagrammene som passer.
- Vi ble enige om å ikke bruke Gantt diagram.
- Ellers dokumentere alt vi gjør (møter, beslutninger, progresjon osv).

### Prosess

- Vi går inn for å ha jevnlig møter med veileder som skal skje ca. annenhver uke.

### Arbeidsmetode:

- Det virker naturlig for oss å ta i bruk SCRUM av praktiske årsaker, vi kommer da til å bruke en modifisert versjon der noen aspekter av scrum ikke kommer til å fulgt helt slavisk.

### Timeregistrering:

- Vi ble enige om at vi kunne bruke det digitale verktøyet 'clockify' til å registrere timer da det er lett å bruke, oversiktlig og har PDF-utskrift.

### Ambisjonsnivå:

- Vi går for toppkarakter, ved å få høy uttelling og vise høyt kompetansenivå på prosess, metode og sluttprodukt.

### Forskningsdel:

- Diskuterte og ble gjort oppmerksomme på at forskning er en sentral del av dokumentasjonen, og at vi fra dag en skal fokusere på å formulere og planlegge fremdriften med dette i tankene, slik at det blir en rød tråd gjennom prosjektet som helhet.

- All dokumentasjon sendt inn til veileder skal være i PDF format.

- Alle emails angående bachelor skal ha formatet [BO44 TELEVINDU]{emne} i emnefeltet.

## Møtereferat 02 - Visjons- og kravdokument (20.02.2020)

### Visjonsdokument

- Skal inneholde mer overordnet informasjon, for eksempel om funksjonelle / ikke funksjonelle egenskaper.
- Skal ikke føre alt slavisk, og ikke ting fra mal som ikke er relevant for oss.
- Detaljert informasjon om egenskaper hører heller hjemme i kravdokument.

### Framdriftsplan

- Kan godt inneholde start og slutt tid.
- Det viktigste er slutt-tid når et punkt er ferdig.
- For eksempel: "i slutten av mars skal vi være ferdig med x".

### Timeliste

- Kan bruke det systemet vi har begynt å føre inn i (clockify).
- Kan godt ha med statusrapport. Dette kan for eksempel være hvis noe har tatt en hel uke eller lignende. Dette kan være en liten paragraf.

### Problemstilling

- Bruker den problemstillingen vi har foreløpig.

### Research

- Skal bygge videre på og referere det televindu / andre studenter har forsket på i hovedrapporten.

### Innhenting av brukerdata

- Finne ut om det er anonymisert eller de-identifiserbare data.
- Er Televindu sitt ansvar å ha oversikt over dette

## Møtereferat 03 - Datainnsamling (16.04.2020)

### Datainnsamling

- Vi hadde tidligere planlagt å basere mye av forsknings og vitenskapelig-resultat delen av hovedrapporten på brukertester og intervjuer av prototype på diver syke- og gamle hjem. Dette er ikke lenger mulig grunnet korona-situasjonen. Vi skal nå heller basere oss mer på teori og tidligere forskning angående vår problemstilling.

### Hovedrapport

- Harvard og APA stil skal brukes som referansestil.
- Det må dokumenteres at 'livestreaming' ikke skal være med, ettersom oppdragsgiver har endret dette kravet, som var med i den originale oppgaven.
- Vi skal ha med overordnet SCRUM dokumentasjon. Dette skal være med som vedlegg. Eks sprint reviews osv. Trenger ikke ha med detaljert scrum-board fra sprintene.



# Timelister

## Sammendrag

- Uke 6, 7 og 8

Snakket med oppdragsgiver om funksjonelle implementasjoner og kom opp med et utkast for problemstilling. Brukte mye tid på å finne en kontorplass i Trondheim. Lagde framdriftsplan, visjonsdokument og kravdokument. Begynte å lære Flutter.

- Uke 9 og 10

Satte opp GitLab prosjekt med issue board, og delte opp use casene i issues og sprinter.

Erlend og Simen fortsatte seg å lære Dart og Flutter. Snorre fikset download funksjon, satte opp iOS distribution og begynte å integrere google cast.

- Uke 11 og 12

Fortsatte utvikling for Chromecast, lagde onboarding screen, fikset at UI'en ble mer responsive, satte inn Televindu's egne video ikoner. Vi brukte en del tid på å øve på systemtenking og økonomi eksamen, og det ble mye vanskeligheter med å finne arbeidsplass etter korona virus.

Angående om det går greit med prosjektet angående korona, så får vi ikke testet appen med sykehjem og eldrehjem. De har mye å tenke på angående coronavirus så de har ikke kapasitet til å bistå lenger mtp. undersøkelser, testing og feedback.

- Uke 13 og 14

De to siste ukene har vært ganske hektiske. Televindu har bestemt å gjøre plattformen gratis fram til sommeren på grunn av korona situasjonen. Dette førte til at vi måtte publisere appen på google play store tidligere enn forventet, i tillegg måtte strukturen på prosjektet bli endret til å tilpasse gratis kunder. Alle er godt i gang med utviklingen av produktet. Har påbegynt skriving av hovedrapport.

Som nevnt tidligere så kommer vi ikke til å få noe tilbakemelding fra brukere pga koronavirus. Dette vil da gjøre at vi ikke kan drøfte funnene og tilbakemeldingene fra testene vi hadde tenkt å gjennomføre. Dette fører til at vi må fokusere mer på teori og research. Si gjerne om du har noen tilbakemelding på at vi gjør det slik, og om det er et ok alternativ til at vi ikke får brukertestet produktet vårt på sykehjem.

- Uke 15 og 16

Vi har fortsatt med hovedrapporten og implementert airplay for iOS og lokasjonsbaserte funksjoner som videoer basert på hvor du befinner deg.

Videre har vi hatt noen møter, bl. a. med veileder og med oppdragsgiver.

- Uke 17 og 18

Vi har snakket med oppdragsgiver og folk i Televindu om research angående vår problemstilling. Vi har begynt prosessen med å deploye applikasjonen på Google Play store og App store. Vi gjorde oss ferdig med brukerhistorikk og har brukt en del tid på diverse bugfiksing og forbedringer. Mesteparten av tiden har gått på å skrive i hovedrapporten.

# Rapport fra Clockify

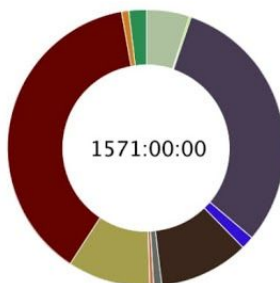
## Summary report

01/01/2020 - 12/31/2020

Total: 1571:00:00 Billable: 00:00:00

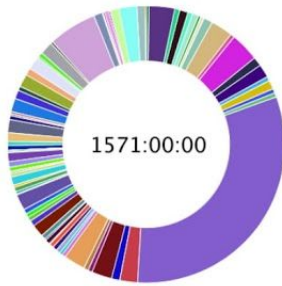


## Tag



Distribusjon	32:00:00
Research	13:00:00
Koding	595:00:00
Planlegging	152:30:00
Dokumentasjon, Møte, Planlegging	07:00:00
Møte	14:30:00
Research, Koding	165:00:00
Research, Planlegging	24:00:00
Dokumentasjon	485:00:00
Møte, Planlegging	03:00:00
Dokumentasjon, Research	80:00:00

## Time Entry



● Oppstartsmøte med veileder	01:30:00
● Kursing i Dart/Flutter	16:00:00
● Flutter koding.	32:00:00
● Integreerte github actions	17:00:00
● Planlegge bachelor oppgaven	02:00:00
● Planlegging & research.	08:00:00
● Videre planlegge bachelor oppgaven	02:30:00
● Oppstartsmøte	02:00:00
● Bugfixing	16:00:00
● Koding	88:00:00
● Fikset iOS distirbution	04:00:00
● Scrum oppsett	24:00:00
● Spilleliste	08:00:00
● Sette seg inn i bachelor oppgaven og planlegge møter og visning av lokale	03:00:00
● App Store distribusjon	24:00:00
● Krav dokumentasjon	13:00:00
● Google Play distribusjon	24:00:00
● Researcha Google Cast for dart	08:00:00
● GUI carousel for videoer i nærheten	16:00:00
● Flutter koding & Kurs.	24:00:00
● Sattt opp prosjektet på gitlab	05:00:00
● Introduksjonsmøte med oppdragsgiver	03:00:00
● Sette opp scrum-backlog for de kommende ukene. Vi brukte verktøyet trello for å visualisere dette digitalt.	07:00:00
● Koding på brukerhistorikk	24:00:00
● Integreerte agolia i backend	15:00:00
● Få videoer i nærheten basert på lokasjonen	08:00:00
● Download bug fix	08:00:00

● Møte med veileder	01:00:00
● Endret user database	02:00:00
● Møte veileder	02:00:00
● Integreerte sendgrid i backend	16:00:00
● Så etter kontorplasser	10:00:00
● Gitlab oppsett	08:00:00
● Visning lokale på DIGS	01:00:00
● On boarding bug fix	03:00:00
● Endret registrering form	06:00:00
● Flutter koding	16:00:00
● Lagde user locale funksjon	04:00:00
● Intro til flutter, oppsett av prosjekt	08:00:00
● Planlagge bacheloroppgave	04:00:00
● Forberedelse til- og møte med veileder	02:00:00
● Sette seg inn i bachelor oppgaven	30:00:00
● Lagde onboarding/help screen	08:00:00
● Oversikt eksisterende system, flutter kursing	08:00:00
● Videoer i nærheten	08:00:00
● Planlegge / Researche bachelor oppgave	09:00:00
● Koding div	24:00:00
● Fikset aspectratio på videospiller	08:00:00
● Planlegge bachelor oppgave	03:00:00
● Begynte på brukerhistorikk	08:00:00
● Visning av lokale på DIGS	02:00:00
● Første utkast visjonsdokument	08:00:00
● Testet appen, flutter koding.	08:00:00
● Integreere Televindu playlists i homeview	08:00:00
● Intro til Dart	05:00:00

● Lage permissions ved første innlogging som ber om å bruke enhetens lokasjon.	07:00:00
● Satte oss inn i bacheloroppgaven	39:00:00
● Litt visjonsdokument, intro til Dart	08:00:00
● Andre utkast visjonsdokument	08:00:00
● Koding i Flutter	37:00:00
● Lage struktur for å lagre lokasjonsdata lokalt	16:00:00
● Airplay for iOS	32:00:00
● Hovedrapport	496:00:00
● Lagde mail templates for sendgrid	08:00:00
● Publiserte beta i goole playstore	08:00:00
● Andre utkast av visjonsdokument	16:00:00
● La till Televindu ikoner for vidoene	08:00:00
● Jobbet med framdriftsplan	24:00:00
● Bugfiksing	16:00:00
● Flutter kursing	53:00:00
● Videospiller bug fix	08:00:00
● Hente brukerhistorikk	40:00:00
● Integreerte Google Cast	16:00:00
● Chromecast controller	07:00:00
● Sett etter kontorplasser	05:00:00
● Satt opp google play store	11:00:00
● Gjorde onboarding responsive	08:00:00
● Første utkast av visjonsdokument	16:00:00
● Flutter koding & Kurs. Innføring.	08:00:00
● Kursing i Dart	48:00:00
● Fikset diverse UI bugs	05:00:00

Tag/Time Entry	Duration
<b>Distribusjon</b>	<b>32:00:00</b>
App Store distribusjon	16:00:00
Google Play distribusjon	16:00:00
<b>Research</b>	<b>13:00:00</b>
Intro til Dart	05:00:00
Researcha Google Cast for dart	08:00:00
<b>Koding</b>	<b>595:00:00</b>
Lagde onboarding/help screen	08:00:00
Oversikt eksisterende system, flutter kursing	08:00:00
Flutter koding.	32:00:00
Integrerte github actions	17:00:00
Koding div	24:00:00
Fikset aspectratio på videospiller	08:00:00
Begynte på brukerhistorikk	08:00:00
Bugfixing	16:00:00
Koding	88:00:00
Fikset iOS distirbution	04:00:00
Testet appen, flutter koding.	08:00:00
Integrere Televindu playlists i homeview	08:00:00
Spilleliste	08:00:00
Lage permissions ved første innlogging som ber om å bruke enhetens lokasjon.	07:00:00
App Store distribusjon	08:00:00
Google Play distribusjon	08:00:00
Lage struktur for å lagre lokasjonsdata lokalt	16:00:00
Airplay for iOS	32:00:00

GUI carousel for videoer i nærheten	16:00:00
Lagde mail templates for sendgrid	08:00:00
Publiserte beta i goole playstore	08:00:00
La till Televindu ikoner for vidoene	08:00:00
Bugfiksing	16:00:00
Flutter kursing	21:00:00
Videospiller bug fix	08:00:00
Satt opp prosjektet på gitlab	05:00:00
Hente brukerhistorikk	40:00:00
Integrerte Google Cast	16:00:00
Chromecast controller	07:00:00
Koding på brukerhistorikk	24:00:00
Integrerte agolia i backend	15:00:00
Få videoer i nærheten basert på lokasjonen	08:00:00
Download bug fix	08:00:00
Satt opp google play store	11:00:00
Endret user database	02:00:00
Gjorde onboarding responsive	08:00:00
Integrerte sendgrid i backend	16:00:00
On boarding bug fix	03:00:00
Endret registrering form	06:00:00
Flutter koding	16:00:00
Lagde user locale funksjon	04:00:00
Intro til flutter, oppsett av prosjekt	08:00:00
Fikset diverse UI bugs	05:00:00
<b>Planlegging</b>	<b>152:30:00</b>



Jobbet med framdriftsplan	08:00:00
Planlegge bachelor oppgaven	02:00:00
Planlegge / Researche bachelor oppgave	09:00:00
Planlegge bachelor oppgave	03:00:00
Videre planlegge bachelor oppgaven	02:30:00
Sett etter kontorplasser	02:00:00
Scrum oppsett	24:00:00
Sette seg inn i bachelor oppgaven og planlegge møter og visning av lokale	03:00:00
Første utkast av visjonsdokument	08:00:00
Så etter kontorplasser	10:00:00
Gitlab oppsett	08:00:00
Satte oss inn i bacheloroppgaven	39:00:00
Planlagge bacheloroppgave	04:00:00
Sette seg inn i bachelor oppgaven	22:00:00
Andre utkast av visjonsdokument	08:00:00
<b>Dokumentasjon, Møte, Planlegging</b>	<b>07:00:00</b>
Sette opp scrum-backlog for de kommende ukene. Vi brukte verktøyet trello for å visualisere dette digitalt.	07:00:00
<b>Møte</b>	<b>14:30:00</b>
Møte veileder	02:00:00
Oppstartsmøte med veileder	01:30:00
Visning lokale på DIGS	01:00:00
Introduksjonsmøte med oppdragsgiver	03:00:00
Oppstartsmøte	02:00:00
Møte med veileder	01:00:00
Forberedelse til- og møte med veileder	02:00:00
Visning av lokale på DIGS	02:00:00

<b>Research, Koding</b>	<b>165:00:00</b>
Flutter kursing	32:00:00
Kursing i Dart/Flutter	16:00:00
Flutter koding & Kurs. Innføring.	08:00:00
Koding i Flutter	37:00:00
Kursing i Dart	48:00:00
Flutter koding & Kurs.	24:00:00
<b>Research, Planlegging</b>	<b>24:00:00</b>
Litt visjonsdokument, intro til Dart	08:00:00
Planlegging & research.	08:00:00
Sette seg inn i bachelor oppgaven	08:00:00
<b>Dokumentasjon</b>	<b>485:00:00</b>
Jobbet med framdriftsplan	16:00:00
Første utkast av visjonsdokument	08:00:00
Krav dokumentasjon	13:00:00
Andre utkast visjonsdokument	08:00:00
Hovedrapport	424:00:00
Andre utkast av visjonsdokument	08:00:00
Første utkast visjonsdokument	08:00:00
<b>Møte, Planlegging</b>	<b>03:00:00</b>
Sett etter kontorplasser	03:00:00
<b>Dokumentasjon, Research</b>	<b>80:00:00</b>
Videoer i nærheten	08:00:00
Hovedrapport	72:00:00