

# Fordypningsprosjekt: Manuell Sanking av Sau

TDT4501 / IT3915

Jonassen, Andreas Jensen  
Vassbotn, Lars Erik

Veileder: Hvasshovd, Svein-Olaf

Trondheim, høst 2021

# Innhold

<b>1 Terminologier</b>	<b>4</b>
<b>2 Innledning</b>	<b>5</b>
<b>3 Bakgrunn - Teoretisk Grunnlag</b>	<b>6</b>
<b>4 Funksjonelle Krav og Prototyping</b>	<b>7</b>
4.1 Initiell beskrivelse av løsningen . . . . .	7
4.1.1 Grunnleggende om applikasjonen . . . . .	7
4.1.2 Kart og lokasjon . . . . .	7
4.1.3 Gjøre observasjoner . . . . .	8
4.1.4 Lagring av tilsynsrunde . . . . .	9
4.1.5 Synkronisering . . . . .	9
4.2 Prototyping og design . . . . .	9
4.2.1 Tidlige designavgjørelser . . . . .	9
4.2.2 Presentasjon av første prototype . . . . .	10
4.2.3 Tilbakemelding på første prototype . . . . .	20
4.2.4 Nytt design for registrering av sau . . . . .	20
4.2.5 Tekst-til-tale . . . . .	25
4.2.6 Resultat av nytt design for registrering av sau . . . . .	27
<b>5 Utvikling og Implementasjon</b>	<b>28</b>
5.1 Avgjørelser om Teknologi . . . . .	28
5.1.1 Teknologiske krav . . . . .	28
5.1.2 Valg av rammeverk for mobilutvikling . . . . .	29
5.1.3 Valg av kartteknologi . . . . .	29
5.1.4 Valg av backend-system . . . . .	30
5.2 Valg av plattform for kodelager og koordinering av utvikling . . . . .	31
5.3 Utviklingsprosessen . . . . .	31
5.4 Funksjonalitet utover prototypen . . . . .	31
5.4.1 Mine sporinger . . . . .	32
5.4.2 Valg av farger for øremerker . . . . .	33
5.4.3 Endring av valg av øremerke i saueflokk . . . . .	33
5.4.4 Trekke linjer til observasjonsmarkører . . . . .	35
5.4.5 Rovdyr og "annet" . . . . .	37
<b>6 Brukertest og Tilpasninger</b>	<b>38</b>
6.1 Om utføring av brukertest . . . . .	38
6.2 Første Runde Brukertest . . . . .	38
6.2.1 Oppgaver for første brukertest-runde . . . . .	39
6.2.2 Resultater fra første runde brukertest . . . . .	41
6.2.3 Endringer etter første runde brukertest . . . . .	43
6.3 Andre Runde Brukertest . . . . .	43
6.3.1 Oppgaver for andre brukertest-runde . . . . .	44
6.3.2 Resultater fra andre runde brukertest . . . . .	44

6.3.3	Endringer etter andre runde brukertesting . . . . .	45
6.4	Konklusjon av brukertesting . . . . .	46
<b>7</b>	<b>Konklusjon og Refleksjon</b>	<b>47</b>
<b>8</b>	<b>Videre arbeid</b>	<b>48</b>
<b>Appendix A</b>	<b>Møte 25. september 2021</b>	<b>50</b>
<b>Appendix B</b>	<b>Møte 29. september 2021</b>	<b>53</b>
<b>Appendix C</b>	<b>Møte 6. oktober 2021</b>	<b>55</b>
<b>Appendix D</b>	<b>Møte 20. oktober 2021</b>	<b>57</b>
<b>Appendix E</b>	<b>Møte 24. november 2021</b>	<b>60</b>

# 1 Terminologier

**backend** Del av et datasystem som brukeren ikke direkte interagerer med, men som fungerer i bakgrunnen og typisk understøtter lagring og manipulering av data[1]. 30

**frontend** brukergrensesnitt; det visuelle systemet brukeren direkte interagerer med[2]. 29

**kryssplattform** Innenfor utvikling er kryssplattform, også kalt multiplattform, programvare som er utviklet for å kjøre på flere forskjellige operativsystemer. Innenfor mobilutvikling spesifikt betyr dette at en applikasjon er skrevet slik at den kan kjøre på både Android og iOS, samt potensielt andre mobiloperativsystem. 9, 29

**native** Innenfor utvikling innebærer begrepet 'native' programvare som er utviklet for én spesifikk teknologisk platform (operativsystem), for eksempel iOS på iPhone. 29

**rammeverk** innenfor programmering er et rammeverk en platform for å understøtte utvikling av programvare. Et rammeverk gir typisk tilgang til forhåndsdefinerte klasser og funksjoner som kan brukes til å interagere med maskinvare eller annen programvare[3]. 28



## 2 Innledning

Som en del av arbeidet med masterprosjektet innenfor manuell sankning av sau, har vi i dette fordypningsprosjektet fokusert på lage en løsning som kan brukes videre som en del av et større system for å behandle observasjonsdata om sau på beite. Fordypningsprosjektet er begrenset til å designe, utvikle og teste en løsning for aspektet rundt å opprette denne observasjonsdataen fra en mobiltelefon. Mer spesifikt har prosjektet omhandlet å lage en mobilapplikasjon som tillater brukere å opprette ulike typer observasjonsdata ved å knytte disse til geografiske punkt på et kart. Data fra applikasjonen skal legge grunnlaget for arbeidet med selve masterprosjektet til våren, der observasjoner om blant annet saueflokker og rovdyr skal kunne innhentes, presenteres og dermed kunne tolkes av brukeren.

Gruppen vår består av Andreas Jensen Jonassen og Lars Erik Vassbotn. Andreas går studiet datateknologi (MTDT), og Lars Erik går studiet Master i informatikk (MSIT). Svein-Olaf Hvasshovd har vært veilederen vår under utførelsen av prosjektet.

Arbeidet med løsningen for dette fordypningsprosjektet, en mobilapplikasjon, presenteres i denne rapporten. Arbeidet var delt inn i tre hovedfaser: planlegging og prototyping (seksjon 4), utvikling (seksjon 5), og til sist brukertesting (seksjon 6). Den teoretiske bakgrunnen for prosjektet presenteres i seksjon 3.

### 3 Bakgrunn - Teoretisk Grunnlag

Sauer på beite har tilgang til store områder der de fritt kan bevege seg rundt. De flytter seg som regel som samlede flokker, men visse dyr kan til tider bli adskilt fra flokken. Enkeltdyr kan bli skadet eller drept, for eksempel som resultat av fall eller rovdyrangrep. Hvor flokken har vandret, om sau mangler eller om sauer er skadet, er alle ting som er av interesse for sauebonden. En viktig grunn til dette er av interesse for velferden til dyrene - forskriften om velferd for småfe kravsetter at sau og andre dyr på utmarksbeite skal sees etter minimalt én gang i uka[4]. Det er også økonomiske årsaker til at sauebønder er interessert i opplysninger om dyrene på beite; "forskrift om erstatning når husdyr blir drept eller skadet av rovvilt" legger grunnlag for at bønder kan motta finansiell erstatning for dyr som er skadet eller drept som resultat av angrep fra rovvilt, slik som bjørn eller ulv[5]. For å ha krav på erstatning må bondens dyrehold hatt samsvar med forskriften om velferd for småfe som nevnt tidligere, og det skal kunne dokumenteres sannsynlighet for at skaden(e) skyldes rovvilt. I fjor ble omtrent 14 700 sauer og lam betalt erstatning for som følge av rovdyrangrep, som ga utslag i over 37 millioner kroner i utbetalinger[6].

Sauene i en flokk ute på beite har ikke alltid tilhørighet til samme gård; sau fra ulike gårder blir gjerne sluppet ut på beite fra samme område til omtrent lik tid. Dette danner saueflokker som består av sauer fra ulike gårdsopprinnelse[7], og det kompliserer opptelling av sau på beite, ettersom man ikke kan anta at alle sauene tilhører samme gård. Observasjoner av skadet sau er av spesiell interesse for sauebøndene, ettersom sår og bittmerker er tegn på at rovdyr befinner seg i området. Direkte observasjoner av rovdyr er også viktig å få registrert ettersom det er av interesse for sikkerheten til dyrene og for å kunne dokumentere erstatningskrav.

Informasjon om saueflokker og andre detaljer samlet inn under tilsynsrunder har tradisjonelt vært registrert analogt, på ustrukturert vis, for eksempel ved å skrive på lapper eller i en notatbok. Tanken bak dette prosjektet er å legge til rette for slike registreringer på digitalt vis. Løsningen vil være en mobilapplikasjon som brukes av observatører ute "i felten" hvor registreringene senere kan synkroniseres over internett og dermed kan komme til nytte for andre.

## 4 Funksjonelle Krav og Prototyping

Denne seksjonen redegjør for hvordan løsningen - en mobilapplikasjon - ble designet via prototyping, og hva slags arbeid og hvilke avgjørelser som gikk inn i designe denne prototypen. Prototypen la grunnlaget for hva som skulle utvikles i et senere stadie i prosjektet.

### 4.1 Initiell beskrivelse av løsningen

Under første møte med veileder fikk vi vite omfanget av løsningen og fundamentale detaljer om hvordan observasjoner av sau på beite gjøres i praksis (vedlegg A). Disse ulike aspektene beskrives i delseksjonene under.

Løsningen er tiltenkt som en mobilapplikasjon som benyttes av observatører ute "i felten". Kjernefunksjonen til applikasjonen er at observatører skal gå rundt i beiteområdene til sauen og registrere observasjoner av saueflokker, hvor antall sau og gården(e) de tilhører skal registreres.

#### 4.1.1 Grunnleggende om applikasjonen

Applikasjonen skal brukes ute i sauens beiteområde. Dette er utmarksområder hvor nettilgang ikke er garantert, og det er derfor svært viktig at applikasjonen ikke er avhengig av nettilkobling under bruk.

Observatørene har ikke mulighet til å gå på kloss hold av alle saueflokkene de observerer; det antas at observatører benytter seg av kikkerter for å inspisere detaljene i saueflokken. Det vil derfor være naturlig at oppmerksomheten til observatøren vil veksle mellom mobilapplikasjonen og å se på saueflokken gjennom en kikkert.

Når observatøren ikke aktivt registrerer noe på applikasjonen, er det naturlig at mobilen vil befinne seg i lommen til observatøren. Mobilen er da i en "inaktiv" tilstand.

#### 4.1.2 Kart og lokasjon

Kart er et viktig aspekt av løsningen. Brukerens lokasjon skal illustreres på kartet. Økter hvor applikasjonen brukes kan refereres til som "tilsynsrunder", og tilsynsrunder skal kunne startes og stoppes av brukeren. Når en tilsynsrunde er i gang skal brukerens lokasjon loggføres i et visst intervall. Tidligere brukerlokasjoner fra gjeldende tilsynsrunde skal illustreres som punkter, og det skal trekkes rette linjer mellom disse punktene for å illustrere "ruten" observatøren har gått så langt. Sporing av brukerlokasjon skal fungere selv når applikasjonen ikke er åpen, med andre ord skal sporingen fungere når mobilen er lagt i lommen.

Ettersom applikasjonen skal fungere uten nettilkobling, må brukeren ha evnen til å laste ned kartområder på forhånd, slik at kartdataen for en tilsynsrunde er tilgjengelig lokalt. Dette skal være en integrert funksjon i applikasjonen. Flere kartområder skal kunne være lagret om gangen, og kartområdene skal kunne navngis. Før en tilsynsrunde må brukeren oppgi hvilket lagret kartområde de vil bruke.

#### 4.1.3 Gjøre observasjoner

Det finnes flere ulike observasjoner som kan gjøres, og de ulike observasjonstypene krever ulik type data.

Felles for alle observasjoner er at de skal plasseres på kartet. Under en aktiv tilsynsrunde vises knapper for å gjøre ulike typer observasjoner. Når en slik knapp trykkes instrueres brukeren til å plassere hvor på kartet observasjonen skal plasseres. Brukeren må da plassere en markør på kartet som viser omtrent hvor saueflokken eller noe annet som skal observeres befinner seg. Applikasjonen skal vite om avstanden mellom brukeren og markørens plassering er i overkant av 50 meter. Dette er fordi at ved >50 meters avstand er det spesielt utfordrende å få med seg detaljer på sauene i en saueflokk.

Når posisjonen til observasjonen er bestemt, må det fylles inn ulik informasjon avhengig av hva slags observasjon det dreier seg om. Etter informasjon om observasjonen er fylt inn vil den lagres og markeres på kartet.

##### **Saueflokk**

Ved observasjon av saueflokk skal antall sau og lam telles. Alle sauer har et fargekodet øremerke som indikerer hvilken gård de tilhører, og øremerket til sauene skal registreres. Noen sau har fargekodede slips, og dette indikerer at sauene er en søye med lam, der fargen til slipset indikerer hvor mange lam som har blitt med moren ut på beite. Ved å sammenligne antall lam som skal være i flokken - basert på slipsfarger - og reelt antall sau observert i flokken, kan man finne ut om det er divergens mellom de to.

Når man er ferdig med å telle opp øremerker og slips, bekrefter man observasjonen.

##### **Skadet eller død sau**

Sau som er skadet eller død skal kunne registreres. Et tekstfelt for å kunne beskrive skaden på sauene tilhører her.

##### **Rovdyr**

Man skal kunne registrere observasjoner av rovdyr. Jerv, ulv og bjørn blir beskrevet som de vanligste rovdypene som er problematiske for en saueflokk.

##### **"Annen" registrering**

Det skal kunne registreres observasjoner som ikke tilhører kategoriene beskrevet over. Observasjoner av død elg, reinsdyr eller hoggorm er mulige å komme over i felten, og det vil være logisk med en nedtrekksmeny for å velge disse. Det skal også være mulighet for

å gjøre en observasjon kun med fritekst.

#### 4.1.4 Lagring av tilsynsrunde

Når observatøren er ferdig med tilsynsrunden, trykker de på en knapp for å stoppe den. Da vil lokasjonen til brukeren ikke lenger spores, og tilsynsrunden og alle tilhørende observasjoner lagres lokalt på mobilen. Man kan finne en liste over tidligere tilsynsrunder på applikasjonen.

#### 4.1.5 Synkronisering

Når observatøren er ferdig med tilsynsrundene sine og har nettilgang skal de kunne synkronisere dataen de har lagret lokalt. Dataen vil da lastes opp til et system "i skyen" som lagrer den. Denne dataen vil ikke benyttes direkte i denne applikasjonen, men dataen som blir sendt inn legger grunnlaget for selve masterprosjektet i vårsemesteret.

## 4.2 Prototyping og design

Etter vi hadde en forståelse av hva som var påkrevd av applikasjonen, startet vi arbeid med å designe en prototype av applikasjonen. Denne skulle representere alle de nevnte funksjonalitetene og fungere som en "mal" for hvordan appens brukerflyt og visuelle profil skulle være.

Prototypen ble laget i design- og prototypeverktøyet Figma<sup>1</sup>. Dette er et verktøy for å understøtte prototyping og design av grensesnitt ved å la brukeren dra inn ulike former, figurer, bilder og tekst. Flere brukere kan samarbeide på samme prosjekt samtidig, som var et viktig aspekt for vårt valg av Figma som prototypeverktøy.

### 4.2.1 Tidlige designavgjørelser

Vi ble tidlig informert om at applikasjonen burde være [kryssplattform](#), med andre ord at den skal være tilgjengelig på de to store mobiloperativsystemene; Android og iOS. Ringvirkningene av dette vil redegjøres for mer i seksjon 5.1.2. Dette innebærte at vi måtte designe og programmere applikasjonen til å ha lik oppførsel på både Android og iOS.

En vanlig konvensjon innenfor design og utvikling av brukergrensesnitt er å dele grensesnittet opp i ulike *komponenter*. Hver knapp, for eksempel, er da sin egen komponent. De ulike operativsystemene har sine egne designspråk og tekniske implementasjon om

---

<sup>1</sup><https://www.figma.com>

hvordan komponenter burde og skal se ut. En knapp i iOS vil i bunn og grunn altså se forskjellig ut fra en knapp i Android. Dette kompliserer designprosessen, ettersom man vil ende opp med en applikasjon som ser ulik ut på hvert operativsystem. Dette gjør også utviklingen vanskeligere å forholde seg til fordi visse ting som ser bra ut på ett operativsystem, kan fremstå mye dårligere på det andre operativsystemet.

Ved å benytte et spesifikt designspråk – istedenfor å la mobilens operativsystem bestemme hvordan komponentene skal se ut – unngår man dette problemet. I den sammenheng har vi valgt Googles 'Material Design'-designspråk<sup>2</sup>. Dette er et system hvor ulike komponenter, slik som knapper, har et spesifikt, ferdig utformet design. Dette med hjelper å lage konsekvente grensesnitt over flere plattformer, som minsker behovet for å designe og utvikle for hver spesifikke plattform.

#### 4.2.2 Presentasjon av første prototype

Prototypen er i form av ulike skjermbilder som representerer designet til ulike skjermer i appen. De ulike skjermene viser tilstander appen kan være i og forskjellige funksjonaliteter som appen skal tilby.

---

<sup>2</sup><https://material.io/>

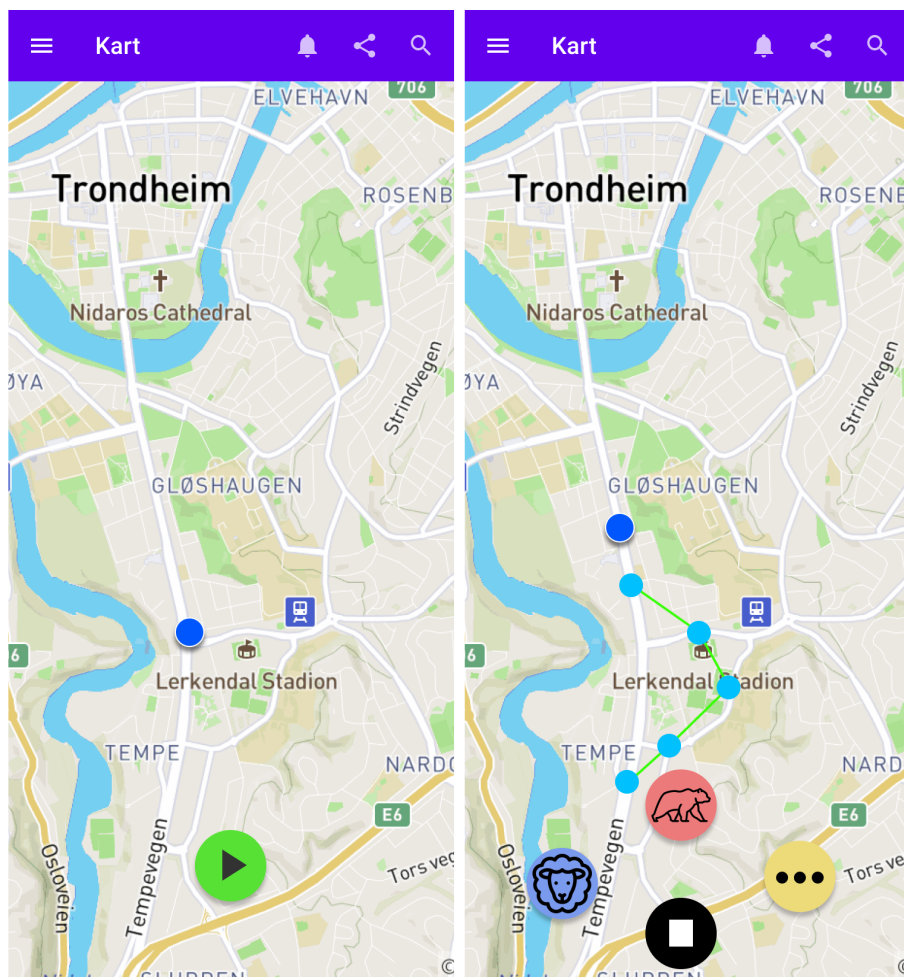


Figure 1: Skjermbilder fra prototypen - sporing av brukeren

Venstre side av figur 1 viser den typiske startskjermen brukeren vil se når de åpner appen. Her presenteres brukeren med et kart, og brukerens nåværende posisjon representeres med en blå sirkel. Når brukeren er på denne skjermen, loggføres ingen data og ingenting kan registreres. For å begynne å spore brukerens posisjon og legge inn observasjoner av saueflokker, rovdyr osv. må man trykke på den grønne knappen med en pil, dette signaliserer "begynn tilsynsrunde".

Høyre side av figur 1 viser skjermen til brukeren etter de har trykket på den grønne knappen. En tilsynsrunde er nå aktiv, og brukerens posisjon vil lagres kontinuerlig i et visst intervall. Brukerens tidligere posisjoner vil plasseres som prikker på kartet, og linjer vil trekkes mellom de ulike tidligere posisjonene. Brukeren har nå også flere knapper i bunnen av skjermen, og ikonene i dem skal signalisere hva slags funksjon de har:

- **Sau:** start registrering av en saueflokk.
- **Bjørn:** start registrering av et observert rovdyr.



- **Tre prikker:** Start en "annen" registrering som ikke faller under kategoriene "sau" eller "rovdyr".
- **Stopp:** Stopp den nåværende tilsynsrunden. All registrert informasjon fra den gjeldende tilsynsrunden vil lagres.

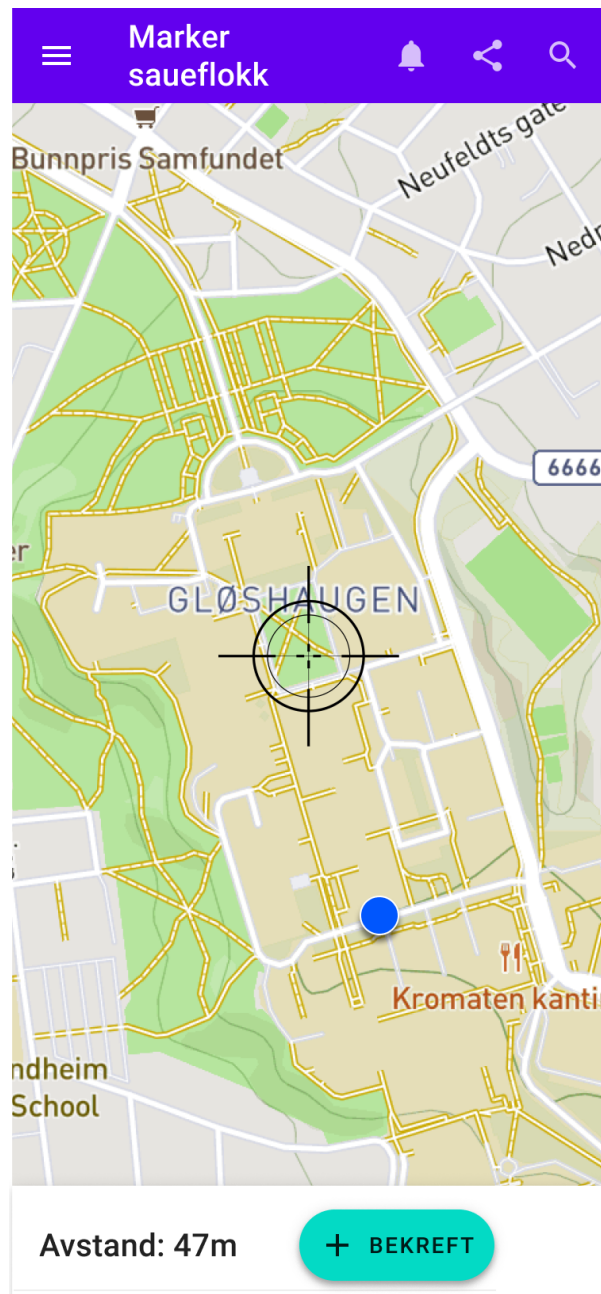


Figure 2: Skjerm bilde fra prototypen - plassering av markør

Figur 2 viser skjermbildet for plassering av en markør. Dette er det første skjermbildet man kommer til etter å ha trykket på "sau"-, "rovdyr"- eller "annet"-ikonet sett i figur 1;



ettersom alle disse krever en spesifikk posisjon på kartet hvor observasjonen skal plasseres. Markøren er låst i midten av skjermen, og brukeren beveger kartet helt til markøren er over det stedet på kartet hvor observasjonen er ønsket plassert. Deretter trykker man på "bekreft"-knappen.

Det neste skjermen som presenteres for brukeren etter markøren er plassert, avhenger av om brukeren er i ferd med å opprette en ny "sau"-, "rovdyr"- eller "annet"-observasjon.

### Registrering av saueflokk

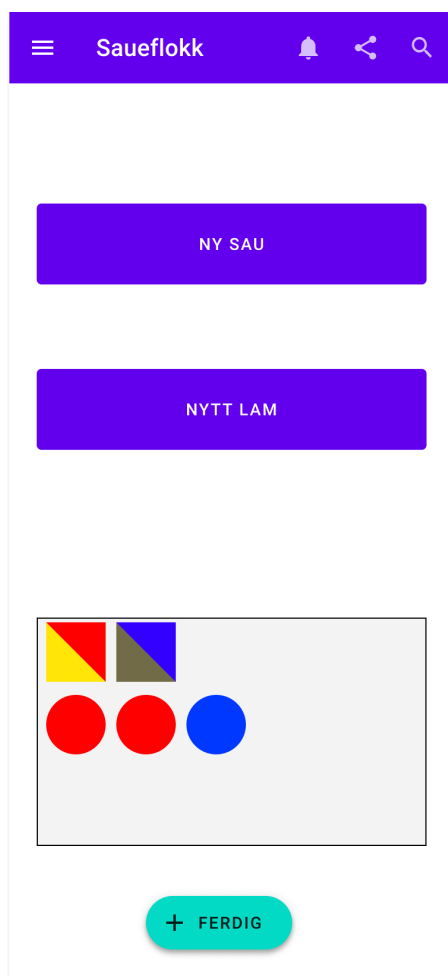


Figure 3: Skjerm bilde fra prototypen - oversikt over saueflokk

Figur 3 viser et skjermbilde av hva brukeren ser etter de har plassert markøren, når de skal lage en ny observasjon av en saueflokk. Denne har to knapper "Ny sau" og "Nytt lam", som brukes for å legge til nye sau og lam til den nåværende flokken. Hver av disse knappene tar brukeren videre til andre skjermer, som vises i figur 4.

Det skal påpekes at brukeren kun tas inn på denne skjermen ved markering av en saueflokk **når saueflokken befinner seg innenfor en spesifikk avstand fra brukeren**. Om

saueflokken er lenger unna enn dette, vil en markør for en saueflokk plasseres på kartet, men brukeren kommer ikke inn på skjermbildet hvor de faktisk må registrere detaljer om flokken. Svein-Olaf beskrev at grunnen for dette er at det ikke vil være tenkelig at en bruker klarer å få med seg detaljene på de individuelle sauene (øremerker og slips) om de er for langt unna.

På bunnen av skjermen i skjermbilde 3 er det en oversikt over sau og lam, og hvilken gård de tilhører. Firkanter representerer sau, og sirkler representerer lam. Som forklart i seksjon 4.1.3 har sau fargekodede øremerker, og søyer med lam har også fargekodede slips. Man kan se at firkantene er delt opp i to farger; venstre side indikerer slipsfargen på sau (om de skulle ha slips - hvis ikke er firkanten ensfarget), og høyre side indikerer gårdstilhørighet. Ettersom sirklene representerer lam og lam ikke er gamle nok til å ha eget avkom, kan sirklene kun ha én farge som da representerer gårdstilhørighet.

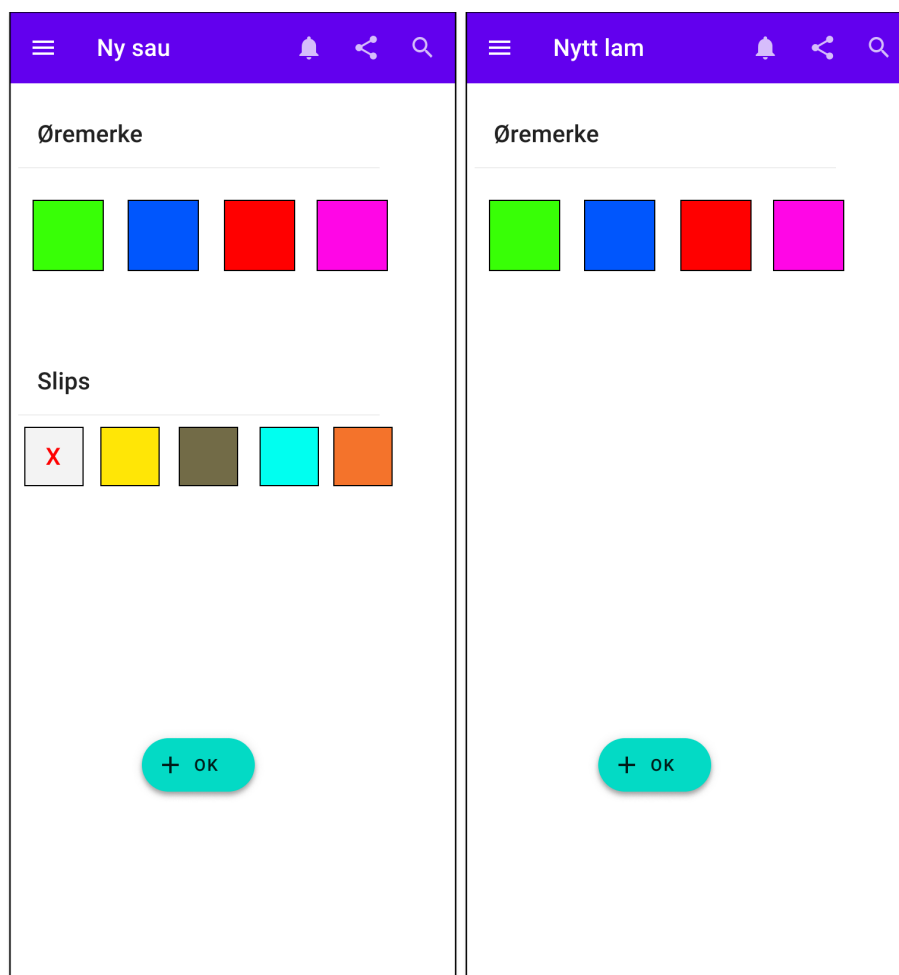


Figure 4: Skjermbilder fra prototypen - Legge til sau (venstre) og lam (høyre)

Skjermbildene i figur 4 viser de to skjermene brukeren kan navigere seg utifra knappene på figur 3. "Ny sau"-skjermen, sett på venstre side av bildet, lar brukeren registrere øremerket til sauene, og hva slags slips sauene måtte ha. Den hvite knappen med et rødt kryss i midten

representerer da at sauene ikke har noe slips. Høyre side viser "Nytt lam"-skjermen, og her kan kun brukeren velge gårdstilhørigheten (representert med fargen på øremerket) til lammet.

På begge skjermene må man bekrefte valget sitt ved å trykke "OK". Den enkelte sauene eller lammet legges da til i oversikten over flokken (skjermbildet i figur 3), og brukeren kan legge til flere sauer eller bekrefte at flokken er ferdig registrert.

Når flokken er bekreftet, plasseres en markør på kartet med sauesymbol.

### Registrering av rovdyr

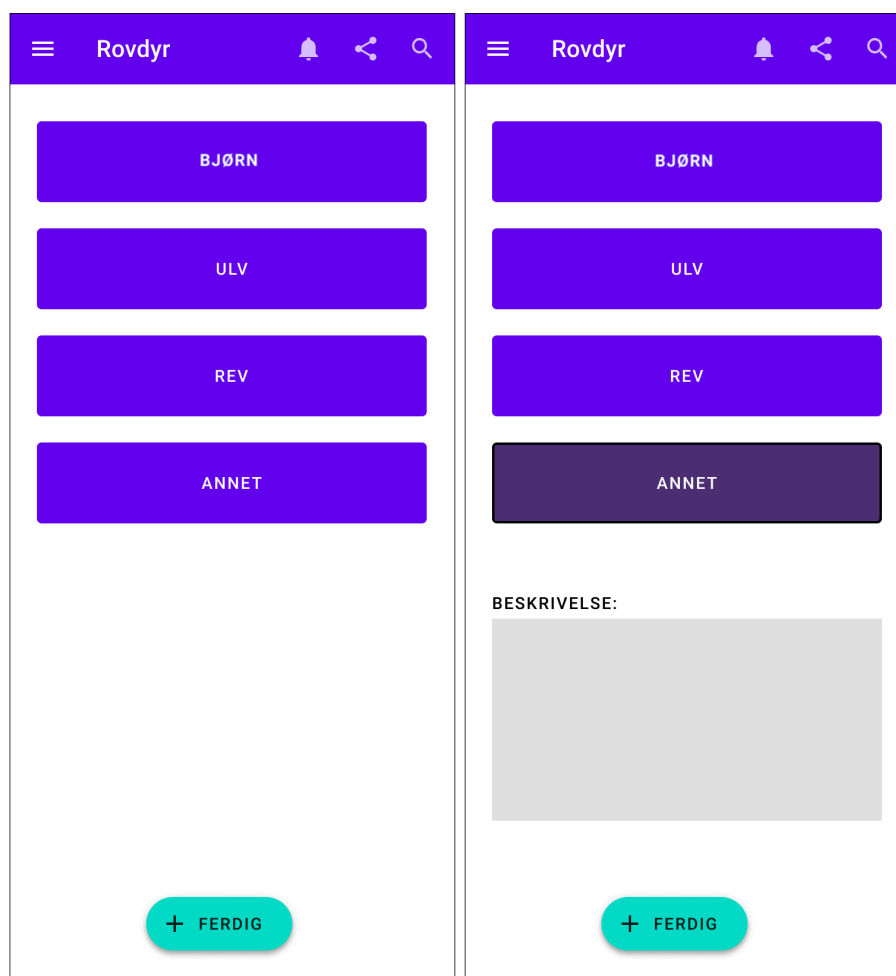


Figure 5: Skjerm bilde fra prototypen - Registrere rovdyr

Figur 5 viser et skjerm bilde av hva brukeren ser etter de har plassert markøren, når de skal lage en ny rovdyrobservasjon. Her er det knapper for å kategorisere rovdyret som bjørn, ulv eller rev. Disse ble valgt fordi de ble beskrevet som de mest vanlige rovdypene som truer sauer på beite. Det er også en "annet" knapp, og som sett på høyre skjerm bilde i figur 5, åpner dette en tekstboks der brukeren selv kan skrive inn hva slags rovdyr de

har sett.

Rovdyrobservasjonen bekreftes ved å trykke på "ferdig"-knappen, og en markør for rovdyr vil plasseres på kartet.

### Registrering av andre typer observasjoner



Figure 6: Skjerm bilde fra prototypen - Registrere "annet"

Figur 6 viser et skjermbilde av hva brukeren ser etter de har plassert markøren, når de skal lage en ny "annet"-observasjon. Dette er for observasjoner som hverken er saueflokker eller rovdyr. Vi ble informert om at observasjon av for eksempel døde elg, hoggorm i terrenget og reinsdyr var mulige ute i felten. Derfor er alle disse mulige å velge fra nedtrekksmenyen. "Annet" er også en mulighet fra nedtrekksmenyen, om observasjonen ikke gjelder noen av verdiene over. Denne skjermen har også en tekstboks som kan fylles ut uansett hvilken verdi man har valgt fra nedtrekksmenyen, og brukes for å oppgi ekstra detaljer rundt observasjonen.

Man bekrefter observasjonen ved å trykke på "ferdig"-knappen på bunnen av skjermen,

og en markør for "annet" vil plasseres på kartet.

### Markører



Figure 7: Skjerm bilde fra prototypen - Markører på kartet

Figur 7 viser hvordan de ulike markørene ser ut plassert på kartet.

### Laste ned kart for offline-bruk

Som beskrevet i seksjon 4.1.1 er det kritisk at applikasjonen fungerer selv uten nettilgang ettersom beiteområdet til sauen typisk har begrenset eller ingen dekning. Den vanlige virkemåten til kartkomponenten i applikasjonen er at den kontinuerlig henter ut kartdata fra internett når brukeren navigerer seg rundt på kartet. Et slik oppsett er ikke mulig i vårt tilfelle. Det er derfor nødvendig at kartdataen er lokalt tilgjengelig på brukerens enhet. Å ha all kartdataen over hele Norge lastet ned lokalt på enheten er urealistisk, ettersom dette ville tatt opp svært mye lagringsplass på mobilen. Det ville vært bedre om brukeren angir et spesifikt område av kartet til å laste ned, slik at de kun har de kartområdene de faktisk trenger tilgjengelig lokalt på mobilen. Dette området burde da være dekkende for der brukeren av appen tenker å gå på tilsynsrunder.

I sammenheng med dette får brukeren instruks om å laste ned et kartområde når de først bruker appen. Brukeren kan også legge til nye kartområder senere, om de har behov for å laste ned ekstra kartområder på et senere tidspunkt.

Figur 8 viser meldingene som instruerer brukeren om å laste ned en kartregion. Disse vises om brukeren åpner appen og ikke har noen lagrede kartregioner på enheten sin, som da gjerne vil være når brukeren bruker applikasjonen for første gang.

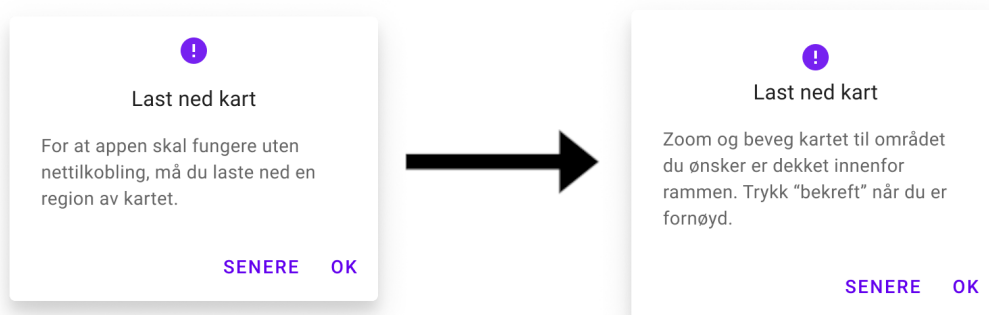


Figure 8: Utklipp fra prototypen - Informasjon om nedlastning av kart

For å ikke gjøre meldingene for lange er de splittet i to sekvensielle meldinger. Ved å trykke OK på meldingen som vises på venstre side av figur 8, får de så meldingen som vises til høyre. Ved å trykke OK på meldingen som vises på høyre, tas brukeren til en skjerm hvor de må bevege kartet helt til det ønskede området er innenfor en blå ramme som dekker kartet. Når brukeren trykker "bekreft", tas de til skjermen som vises i figur 9.

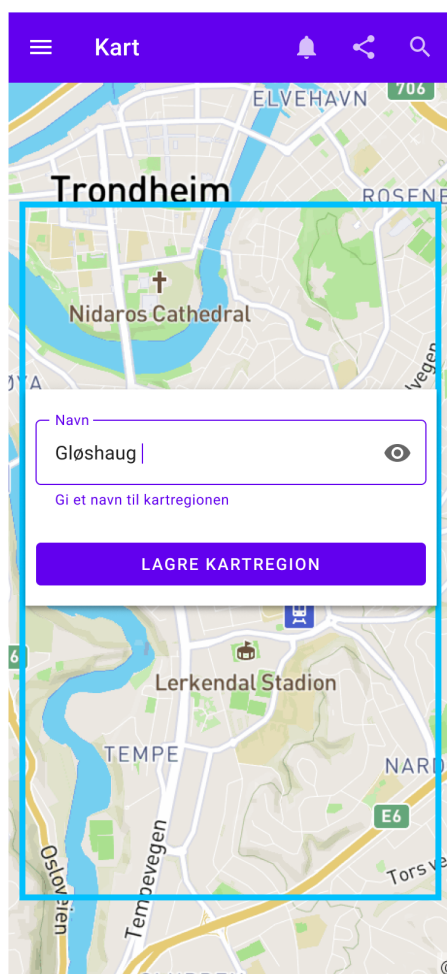


Figure 9: Skjerm bilde fra prototypen - Lagring av valgt kartområde

Figur 9 viser skjermen der brukeren må navngi kartområdet de har valgt. Ved å trykke "lagre kartregion" lagres denne delen av kartet lokalt på enheten og applikasjonen er nå klar for bruk uten nettilkobling.

### Navigasjonsmeny

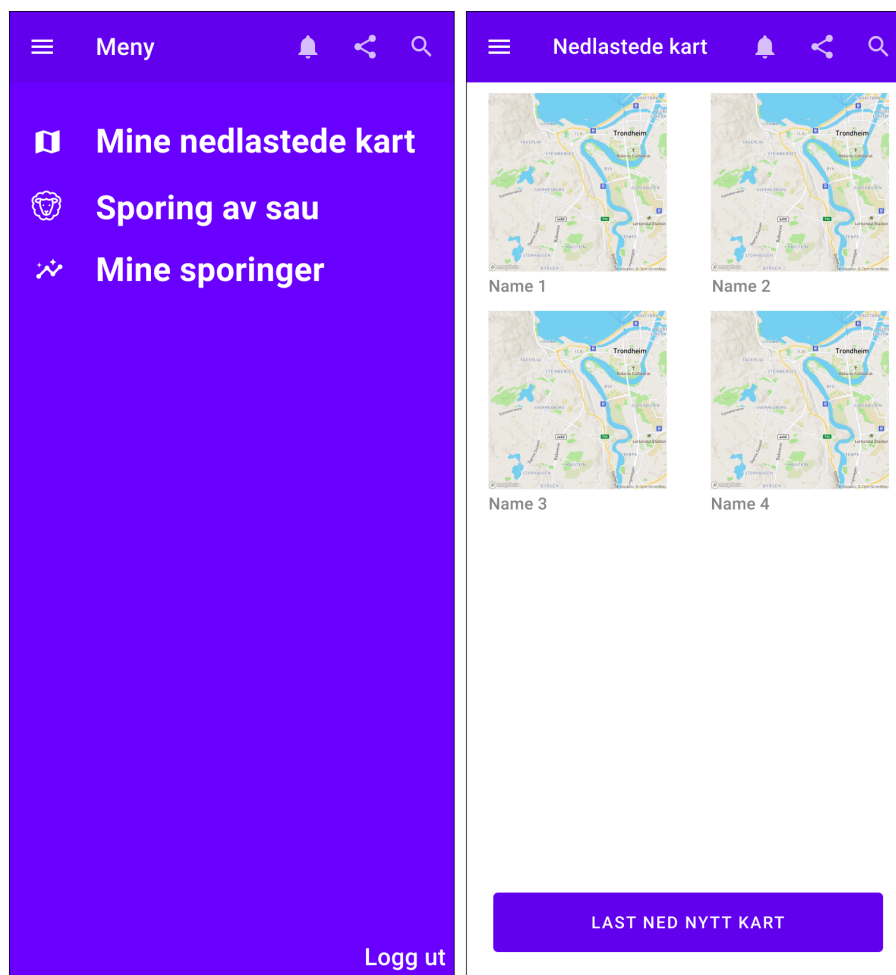


Figure 10: Skjerm bilde fra prototypen - Navigasjonsmeny og oversikt over nedlastede kart

Figur 10 viser to forskjellige skjjermer - på venstre er navigasjonsmenyen, og til høyre er oversikten over nedlastede kartregioner.

Navigasjonsmenyen trekkes fram hver gang man trykker på menyikonet som viser øverst i venstre hjørne på alle skjerm bildene. Her får man valg om å gå til tre ulike skjjermer; "nedlastede kart", "sporing av sau" (det grunnleggende skjerm bildet vist i figur 1 og "mine sporinger" - en liste over tidligere tilsynsrunder. "Mine sporinger" er ikke en del av denne prototypen.

Oversikten over nedlastede kartregioner, sett på høyre side av figur 10, viser de ulike kartregionene brukeren har lastet ned. Her har man også mulighet til å laste ned en ny

kartregion.

Dette konkluderer presentasjonen av den første versjonen av prototypen.

#### 4.2.3 Tilbakemelding på første prototype

Før vi startet på utviklingen av selve applikasjonen ønsket vi en godkjenning av prototypen vår fra veileder Svein-Olaf. Tanken bak dette var å kunne få bekreftelse på at designet vårt var i samsvar med hva han forventet fra applikasjonen med tanke på brukerflyt og funksjonalitet.

Tilbakemeldingen vi fikk (vedlegg B) var at Svein-Olaf var fornøyd med alt utenom designet av skjermen for å registrere en saueflokk (sett i skjermbildene i figur 3 og 4). Det første som ble påpekt som et problem var at symbolene ikke var forståelige - at firkanter og sirkler henholdsvis representerer sau og lam var ikke veldig forståelig. Det andre problemet var at brukbarheten simpelthen ikke var god nok. Ved bruk "i felten" må det være intuitivt å registrere sauer selv om man sjeldent ser på mobilen, eller ikke ser på den i det hele tatt. Vi måtte derfor øke brukbarheten for et slikt bruksscenario.

#### 4.2.4 Nytt design for registrering av sau

Det var to problemer som måtte løses innenfor skjermene for registrering av saueflokker. For det første måtte sauer og lam som er registrert representeres på en mer gjenkjennelig måte, slik at brukeren umiddelbart skjønner om det er en sau eller et lam som har blitt registrert. For det andre måtte brukerflyten være god selv når brukeren ikke nødvendigvis ser på mobilen, eller kun tar et sjeldent blick på mobilen i løpet av prosessen av å registrere sauer og lam.



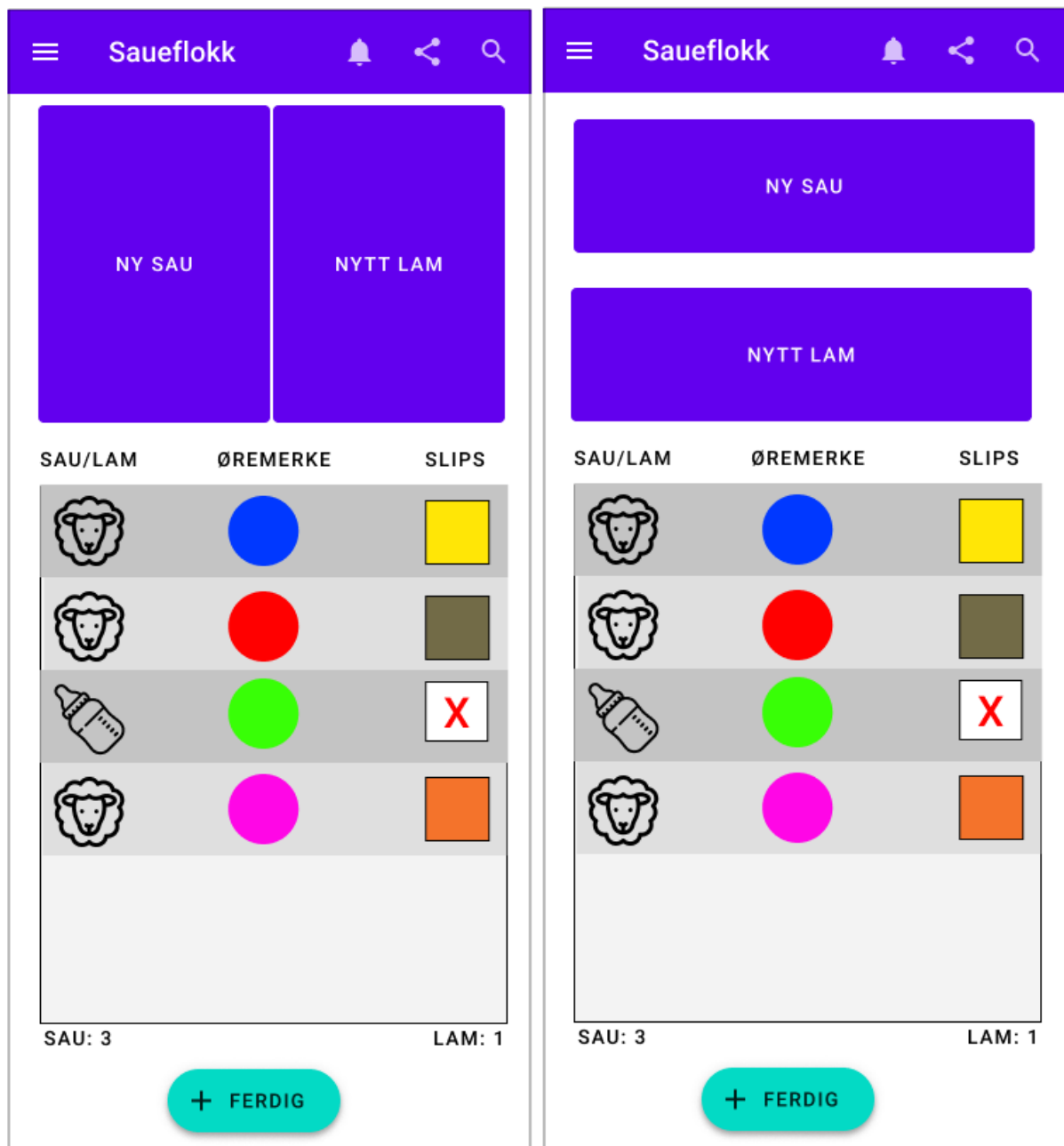


Figure 11: Forslag for nytt design på skjermbildet - Horisontale knapper (venstre) og vertikale knapper (høyre)

Det første vi endret med det nye designet var å endre listen over sauer og lam som hittill er registrert og gjøre denne mer forståelig. Istedet for å bruke former, bruker vi nå ikoner som skal være mer gjenkjennelige uten tidligere erfaring med applikasjonen. Et ikon av en sau brukes for å representere en sau, mens en flaske brukes for å representere lam. Dette gir et tydelig skille mellom de to og er lett å skille mellom. Det var et aktivt valg å ikke bruke et ikon av en "mindre sau" som et ikon for lammene, ettersom dette hadde gjort det vanskeligere å skille mellom de to typene - vi strebet etter å sørge for at ikonene

var svært forskjellige fra hverandre for å unngå at brukeren kunne forveksle dem.

For å sikre god brukervennligheten selv når brukeren ikke aktivt stirrer på skjermen, gjorde vi knappene for "Ny sau" og "Nytt lam" større.

Resultatet fra denne delen av det nye designet kan sees i figur 11. Vi produserte to forskjellige skjermbilder, der den første har knapper som er horisontalt sidestilte, og den andre versjonen har knapper som er stablet vertikalt. Etter litt betenkningstid kom vi fram til at de horisontalt sidestilte knappene hadde høyere brukervennlighet for en bruker som ikke aktivt ser så mye på applikasjonen, fordi det er lettere for en bruker å "blindt" skille mellom høyre og venstre side av mobilen ved å føle seg fram til kantene, enn det er for en bruker å "blindt" prøve å trykke på et spesifikt vertikalt punkt på skjermen som hverken er helt øverst eller helt nederst.

På grunn av resonnementet som ble beskrevet over, valgte vi å gå videre med designet som vises til venstre i figur 11.

Den neste delen som måtte få nytt design var skjermbildene for å oppgi øremerket og slipset til den spesifikke sauene, som man navigeres inn på etter å ha trykket "Ny sau" eller "Nytt lam". Vi produserte to forskjellige prototyper av dette, som vi refererer til som "Nytt design A" og "Nytt design B".

*Vi presiserer at fargene for øremerker og slips vist i skjermbildene, og antall mulige farger, er kun demonstrative ettersom vi fremviser en prototype.*

### **Valg av øremerke og slips: Nytt design A**

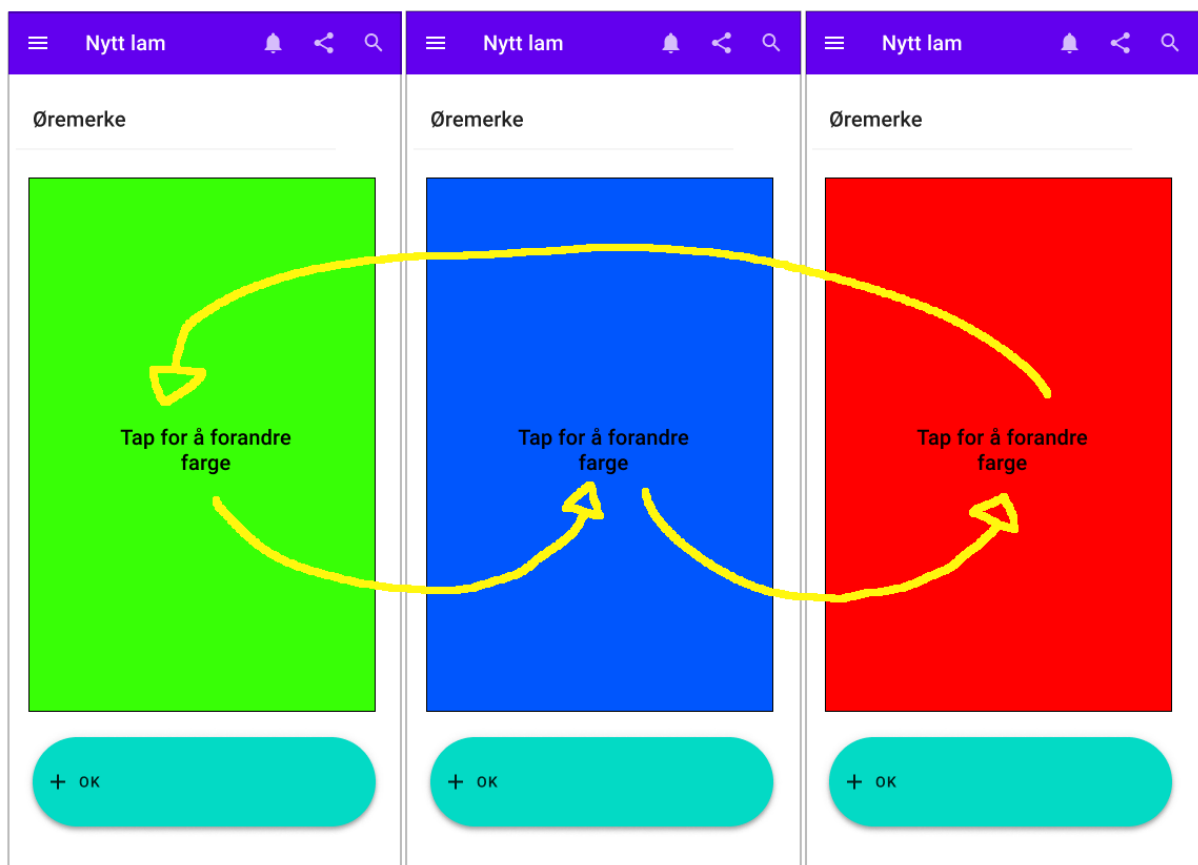


Figure 12: Nytt design A - valg av øremerke



Figure 13: Nytt design A - valg av slips

Nytt design A demonstreres i figur 12. Nytt design A består av en stor boks som er farget med den "nåværende" fargen. Ved å trykke på boksen, går boksen over til neste farge (øremerke). Man veksler altså gjennom de tilgjengelige øremerkene, ett steg av gangen, og trykker på "OK"-knappen nederst når den nåværende fargen er den man ønsker. Valg av slipsfarge fungerer på samme måte, som vist i figur 13. En sau uten slips representeres med et rødt kryss med hvit bakgrunn.

### Valg av øremerke og slips: Nytt design B

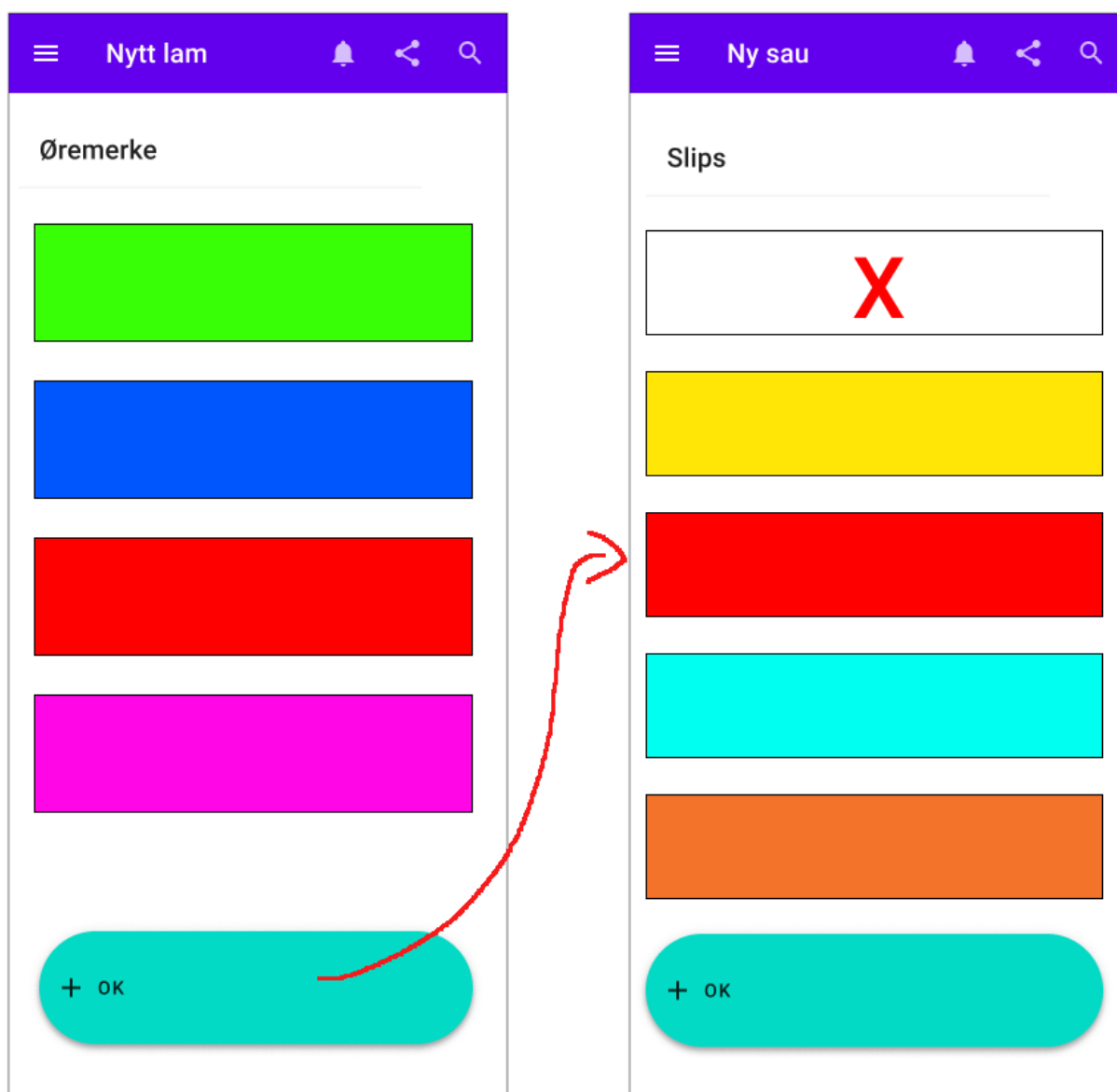


Figure 14: Nytt design B - valg av øremerke og slips

Figur 14 viser hvordan alternativ B for nytt design av valg av øremerke og slips for en sau ser ut. Her går man gjennom to skjermer; den første for øremerke og den andre for slips. På første skjerm velger man en spesifikk øremerke-farge og trykker "OK" for å gå til valg av slips.

#### 4.2.5 Tekst-til-tale

Som tidligere nevnt var det et ønske om at registrering av sau skal kunne utføres selv om brukeren sjeldent ser på mobilen i løpet av prosessen, eller ikke ser på mobilen i det hele tatt. Som designere av mobilapplikasjonen synes vi dette var et nokså unikt

problem, ettersom det viktigste virkemiddelet vårt – det visuelle – ikke lenger alene var nok til å støtte brukerens arbeidsoppgave. For oss var det tydelig at vi måtte blande inn andre sanser enn syn for å kunne understøtte dette. Utenom en skjerm er smarttelefoner utstyr med høyttalere, og tanken vår var derfor å benytte oss av lyd for å hjelpe brukeren gjennom prosessen med å registrere sau og lam.

Vi vurderte forskjellige auditive virkemidler som kunne hjelpe med dette. Vi presenterer en redegjørelse for disse her:

**Talestyring** var det første av disse. Dette innebærer at man snakker og at dette tas opp av mobilens mikrofon, slik at applikasjonen kan tolke talen og deretter gjøre handlinger utifra hva som sies. Ved videre undersøkning viste talestyring seg til å være vanskelig å implementere samtidig som det er relativt upålitelig, spesielt når talespråket ikke er engelsk. Vi valgte derfor å se bort fra talestyring.

**Lydeffekter** ble vurdert for å gi tilbakemelding til brukeren når de gjør handlinger. En lydeffekt kan for eksempel gi tilbakemelding på at brukeren akkurat har registrert et lam til flokken. Vi fant likevel et par ulemper med lydeffekter som auditivt virkemiddel:

- Hver lydeffekt må hentes fra et lydbibliotek, noe som ofte krever en lisens til det gitte lydbiblioteket.
- Alternativt kan lyden lages selv, men vi har lite erfaring med dette.
- Hva lyden representerer må være umiddelbart forståelig for brukeren. Dette er problematisk, ettersom det er vanskelig å angi ulike lydeffekter til for eksempel "sau", "lam", eller "rødt øremerke" som intuitivt forstås av brukeren.

På grunn av disse utfordringene konkluderte vi med at lydeffekter heller ikke var en god løsning.

**Tekst-til-tale** endte opp med å være det vi valgte å gå videre med. Tekst-til-tale går ut på at et dataprogram tar tekst og bruker en kunstig stemme til å uttale denne teksten som lyd. Dette lar oss gi tilbakemelding for hver handling som blir gjort i prosessen av å registrere en saueflokk. Hver gang brukeren for eksempel legger til en sau, vil "Ny sau" spilles ut av mobilens høyttalere som tale. Når den nåværende slipsfargen er blå, vil mobilen spille lyden av en stemme som sier "blå". Dette gir auditiv tilbakemelding til brukeren over hvor de er i prosessen, slik at brukeren kan foreta registrering av sauer mens de bruker minimal tid til å se på mobilskjermen. Kombinasjonen av bruk av store knapper - der deres relative plassering på skjermen er lett å huske - og lyd som gir tilbakemelding, gjør det mulig for brukeren å registrere slipsfarge på sauen uten å måtte se på mobilen.

Tekst-til-tale har som fordel over lydeffekter at fraser uttalt gjennom tekst-til-tale er umiddelbart gjenkjennelige, i motsetning til lydeffekter som kan ha ukjent betydning for brukeren og deres betydning må "læres" med bruk. En annen fordel er at tekst-til-tale er lettere å legge til, fordi tekst-til-tale støttes av alle Android og iOS-smarttelefoner, og

selve frasene som skal sies lett kan programmeres inn i form av tekststrenger som praktisk talt ikke tar noe plass på mobilen. Dette står i kontrast til lydeffekter, hvor hver lydeffekt må lagres i form av en lydfil og dermed tar opp mer lagringsplass på enheten.

#### **4.2.6 Resultat av nytt design for registrering av sau**

Når de to designene hadde blitt ferdigstilt, tok vi de videre til Svein-Olaf for tilbakemelding og godkjenning. Vi ba han om å velge ut det designet han syntes var mest hensiktsmessig, og design A ble utpekt som det beste. Han var også fornøyd med det nye designet på oversikten over saueflokken, og bruken av tekst-til-tale (vedlegg C).

Med dette var all funksjonaliteten som ble beskrevet som krav for applikasjonen representert i prototypen vår, og hele prototypen hadde blitt godkjent av veileder Svein-Olaf. Vi gikk derfor over i en ny fase i prosjektet, der fokuset gikk over til å utvikle selve applikasjonen basert på designet utlagt i prototypen.

## 5 Utvikling og Implementasjon

I denne seksjonen redegjøres det for implementasjonen av løsningen gjennom å programmere en mobilapplikasjon, og hvilke valg angående teknologi som ble tatt underveis.

### 5.1 Avgjørelser om Teknologi

Før utviklingen kunne begynne måtte vi ta avgjørelser om hvilke [rammeverk](#) og teknologier vi skulle benytte oss av for å utvikle mobilapplikasjonen.

#### 5.1.1 Teknologiske krav

Det var svært få tekniske krav spesifisert av veileder Svein-Olaf som vi måtte oppfylle, så lenge de ønskede funksjonelle kravene var implementert på ett eller annet vis. Vi stod derfor fritt til å velge teknologi og rammeverk selv utifra hva vi anså som mest hensiktsmessig. Det var i hovedsak kun tre tekniske krav som måtte oppfylles av applikasjonen:

1. Sporing av brukerens lokasjon må fungere selv om brukeren ikke aktivt har applikasjonen "i fokus". Dette betyr at sporingen må fungere selv om mobilen for eksempel befinner seg i brukerens lomme og at mobilen er i en "inaktiv" (men fortsatt skrudd på) tilstand - gitt at brukeren har valgt å starte sporing av lokasjon.
2. Applikasjonen må kunne støtte all funksjonalitet selv om den ikke har en aktiv tilkobling til nettet (funksjoner som åpenbart krever nettilkobling, slik som synkronisering av data, er unntatt fra dette kravet).
3. Applikasjonen må la brukeren laste ned kartområder til enheten slik at kartdata for disse områdene er tilgjengelige når enheten ikke er tilkoblet nettet.

Ettersom applikasjonen er tiltenkt bruk ute på beiteområder, er en naturlig antagelse å ta at tilgjengelighet til nettet ikke er garantert. Teknisk krav #2 og #3 er derfor passende i forhold til et realistisk bruksmønster til applikasjonen. Ettersom applikasjonen er tiltenkt som et verktøy brukt av en observatør ute på tilsynsrunde, er det også naturlig å anta at observatøren ikke vil ha applikasjonen "i fokus" under hele tilsynsrunden, men at applikasjonen tas i bruk når det er relevant under tilsynsrunden. Teknisk krav #1 er knyttet til dette. Alt i alt gir dette uttrykk for at de tekniske kravene ikke er tilfeldige, men tilstede for å sikre god brukbarhet for et realistisk bruksscenario.



### 5.1.2 Valg av rammeverk for mobilutvikling

Innenfor utvikling av *frontend* til mobilapplikasjoner er det to retninger man kan gå - såkalt *native* eller *kryssplattform*. Vi ble informert av Svein-Olaf om at applikasjonen helst skulle være kryssplattform, men at det var mulighet for å kun støtte enten iOS eller Android (de to mest utbredte operativsystemene for mobiler[8]) om vi kunne begrunne det godt. Ettersom vi utviklerne, Lars Erik og Andreas, har henholdsvis Android-telefon og iPhone, valgte vi å gå for en kryssplattform-løsning ettersom det ville tillate oss begge å teste applikasjonen på våre egne mobiler.

Å skrive en kryssplattform mobilapplikasjon krever bruk av rammeverk, ettersom at de ulike operativsystemene er avhengig av kode som er skrevet spesifikt for den plattformen. Rollen til rammeverk for kryssplattform mobilutvikling er todelt: for det første må rammeverket gi tilgang til å interagere med mobiltelefonen, for eksempel å lese av mobilens GPS-koordinater eller åpne en varselbeskjed på mobilen. For det andre må rammeverket ta programvare som er skrevet i ett språk (det programmererne skriver) og oversette det til kode som forstås av mobilen. En Android-applikasjon vil ikke kjøre på iOS og motsatt, og derfor må rammeverket kunne oversette koden til ny kode for hver enkelt plattform.

Innenfor kryssplattform mobilutvikling er det to rammeverk som regjerer; **React Native**<sup>3</sup>, utviklet av Facebook, og **Flutter**<sup>4</sup>, utviklet av Google. Disse utfyller samme oppgave - bygge applikasjoner for flere plattformer utifra én enkelt kodebase - men har store forskjeller i bruk. Mens React Native baserer seg på rammeverket for webutvikling **React**<sup>5</sup> fra Facebook, og dermed har en syntaks som ligner på JavaScript, benytter Flutter seg av programmeringsspråket Dart.

Fordi både Lars Erik og Andreas har tidligere erfaring med utvikling i React-rammeverket, som React Native baserer seg på, valgte vi å ta i bruk React Native ovenfor Flutter. Dette var for å kunne få en hurtigere start på utviklingen, ettersom dette prosjektet var avhengig av at vi fikk utviklet applikasjonen relativt hurtig grunnet et behov for å kunne komme i gang med brukertesting så tidlig som mulig.

### 5.1.3 Valg av kartteknologi

Å støtte visning av kart er svært viktig ettersom flere funksjoner i applikasjonen er avhengig av å ha et kart. Som beskrevet i seksjon 5.1.1 var det videre et krav om at det skal være mulig å bruke kartet uten nettilkobling, og det vil derfor kreve at kartdata for et område kan lastes ned på forhånd slik at dette er tilgjengelig "ute i felten" der nettdekningen kan være dårlig. Ettersom markører (for eksempel markører for hvor brukeren har gått, eller markør for saueflokk) skal plasseres på kartet, var det også viktig at kartteknologien vi valgte støttet dette. Et siste krav var at kartteknologien var kompatibel

<sup>3</sup>React Native - <https://reactnative.dev/>

<sup>4</sup>Flutter - <https://flutter.dev/>

<sup>5</sup>React-<https://reactjs.org/>

med React Native, ettersom vi hadde bestemt oss for å bruke dette rammeverket.

Utifra disse kravene vurderte vi ulike kartteknologier. Iblant disse var **Mapbox**<sup>6</sup>, **Leaflet**<sup>7</sup> og **Google Maps**<sup>8</sup>. Den eneste teknologien utifra disse som oppfylte alle kravene våre var Mapbox, så vi valgte derfor Mapbox som kartløsningen vår.

#### 5.1.4 Valg av backend-system

En **backend** er nødvendig for å understøtte spesifikke funksjoner i applikasjoner, deriblant:

- registrere en brukerkonto
- logge inn på brukerkonto
- lagre data om utførte tilsynsrunder
- laste ned og laste opp (synkronisere) data om utførte tilsynsrunder

Vi hadde visse krav til backenden som bestemte backend-systemet vi endte opp med å bruke:

1. **Høy tilgjengelighet:** backenden burde så og si "alltid" være operasjonell slik at den alltid kan kommuniseres med. Nedetid for backend skal derfor unngås.
2. **Støtte for automatisk synkronisering og offline-bruk:** Brukeren er ikke garantert å ha tilgang til nett under tilsynsrunder. Det er derfor ønskelig at synkronisering av data fra mobilen skal kunne skje når brukeren får nettilgang igjen, og at data da sendes til backend uten at brukeren selv må gjøre en innsats for å få dataen synkronisert.
3. **Aksepterbart responstid:** Om brukeren må vente lenge på respons fra backend, vil dette redusere den opplevde brukskvaliteten av applikasjonen. Det er derfor viktig at backenden fort sender den dataen som trengs til brukeren, slik at en handling som for eksempel å logge inn, ikke har lang ventetid. Vi setter ingen "harde" krav til responstid, men brukertesting kan eventuelt avdekke om responstiden oppleves som for høy.
4. **Gratis i drift:** Vi har ikke et budsjett til utførelsen av dette prosjektet, og vi har derfor ingen midler til å finansiere serverdrift for et backend-system. Backend-systemet må være gratis i drift, i det minste for vår bruk (få antall brukere, kun for testing).

---

<sup>6</sup>Mapbox - <https://www.mapbox.com/>

<sup>7</sup>Leaflet - <https://leafletjs.com/>

<sup>8</sup>Google Maps - <https://www.google.com/maps/about/>

Kravet om høy tilgjengelighet, sammensatt med kravet om gratis drift, gjorde at vi måtte utelukke å ta i bruk en egen fysisk server, ettersom vi ikke har tilgang til nok fysisk maskinvare for dette (høy tilgjengelighet er effektivt et krav om å ha flere servere plassert på forskjellige steder for å sikre robusthet mot strømbrudd og andre potensielle "brekkende" hendelser, såkalt *redundans*). Vi måtte derfor se etter tilgjengelige løsninger for en sky-basert backend, der serverene driftes av en leverandør som for eksempel **Amazon Web Services**<sup>9</sup> eller **Google Cloud Platform**<sup>10</sup>.

Utifra kravene våre valgte vi **Cloud Firestore**<sup>11</sup> fra Googles Firebase-plattform som vår backend-løsning. Dette er et produkt som utfyller alle kravene våre, og har som fordel at den gjør det særdeles enkelt å kontinuerlig synkronisere data lagret lokalt på mobilen til skyen. I tillegg har Firebase-plattformen svært generøse premisser når det kommer til grenser for hvor mye data som kan lagres før brukeren må begynne å betale for tjenesten, slik at vi kan være sikre på at vi slipper å måtte bruke penger på denne løsningen. Til sist har Firebase-plattformen robuste sikkerhetsregler som standard, slik at sjansen for datainnbrudd i løsningen vår minimeres uten at sikkerhet er noe vi må aktivt forholde oss til, ettersom leverandøren selv tar hånd om sikre alt som lagres på dens plattform.

## 5.2 Valg av plattform for kodelager og koordnering av utvikling

Vi benyttet oss av **GitLab** som kodelager (*repository*). Vi tok også i bruk GitLab for å koordinere hvilke kodeoppgaver vi skulle takle til enhver tid, og å koordinere utdeling av kodeoppgaver oss imellom. Dette hjalp oss med å holde styr over hvem som holdt på med hva, og å sørge for at de ulike tekniske aspektene ved applikasjonen faktisk ble implementert.

## 5.3 Utviklingsprosessen

Utviklingen av applikasjonen lot seg utføre uten nevneverdige problemer eller utfordringer. Ettersom designet av applikasjonen var ferdigjort i prototype-fasen av prosjektet, bestod utviklingen i hovedsak av å gjenskape prototypen som en faktisk funksjonell mobilapplikasjon, gitt de rammeverkene og verktøyene vi har gjort rede for tidligere.

## 5.4 Funksjonalitet utover prototypen

Fordi den faktiske applikasjonen i stor grad er en gjenskapelse av prototypen, unngår vi her å vise alle skjermbildene fra den faktiske applikasjon, fordi disse er demonstrert godt i prototypen i seksjon 4.2.2. Istedenfor viser vi her de skjermbildene som ikke var

---

<sup>9</sup><https://aws.amazon.com/>

<sup>10</sup><https://cloud.google.com/gcp>

<sup>11</sup><https://firebase.google.com/products/firestore>

representert i prototypen, eller divergerer så mye fra prototypen at vi mener deres nye design burde presenteres her.

#### 5.4.1 Mine sporinger

Ett aspekt som var et krav til applikasjon men som ikke var representert av prototypen var "Mine sporinger" - en skjerm med oversikt over tidligere utførte tilsynsrunder. Intensjonen var at man herfra kan gå inn på en eksisterende tilsynsrunde og redigere den - for eksempel slette en eksisterende saueflokk eller legge til nye observasjoner. Man skal også ha mulighet til å slette hele tilsynsrunden. Denne skjermens implementasjon i den faktiske applikasjonen vises i figur 15.

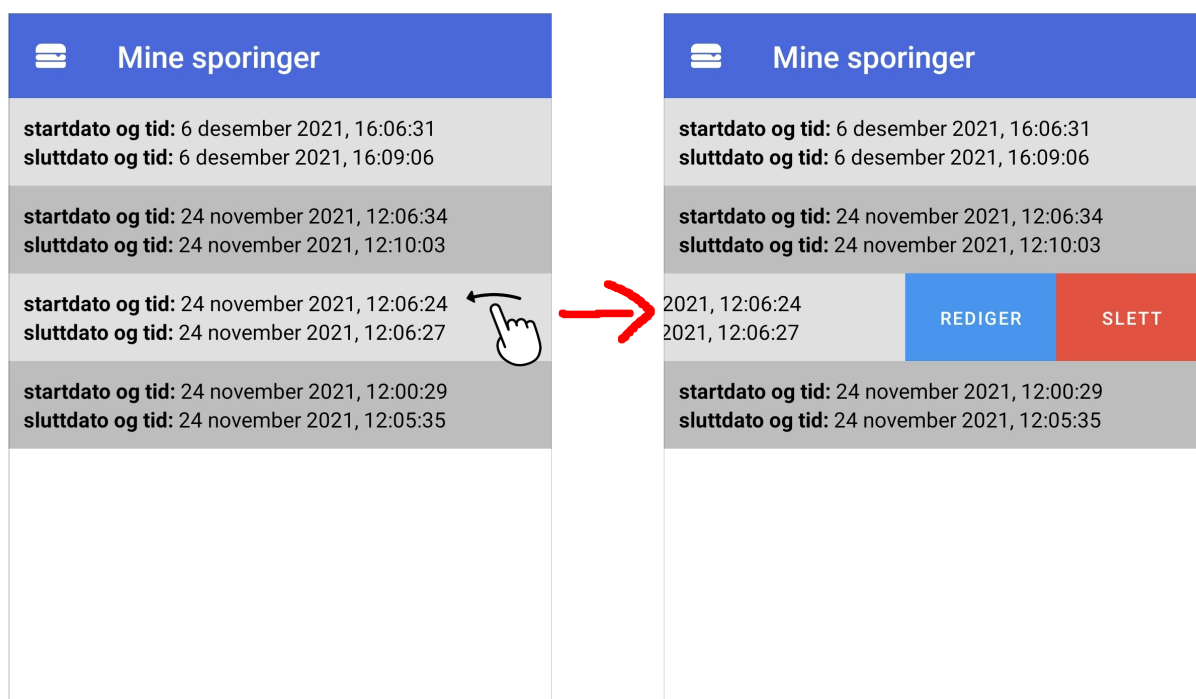


Figure 15: Skjerm bilde - liste av tilsynsrunder. Brukeren kan "sveipe" (engelsk: *swipe*) for å avsløre knapper som gjør handlinger på det enkelte elementet.

Hvert element i listen tillater en 'sveipe'-handling som avslører knappene for redigering og sletting. Ved å trykke på "redigering" tas brukeren til samme skjerm som når de gjør en vanlig tilsynsrunde ("sporing") - som vist til høyre i figur 1. Data fra denne tilsynsrunden (markører, brukerens tidligere lokasjoner osv.) vil da være lastet inn, og brukeren kan trykke på markørene for å redigere dem - de kan for eksempel endre hvilket rovdyr som har blitt observert, eller slette hele markøren. Når brukeren er ferdig med å redigere avslutter de sporingen på vanlig måte ved å trykke på "stopp"-knappen som sett i figur 1.

### 5.4.2 Valg av farger for øremerker

I et møte 20. oktober ble vi bedt om å endre oppsettet for øremerke-fargene (vedlegg D, side 1). Opp til det tidspunktet hadde øremerkene vært "hardkodete" og kunne ikke endres av brukeren. Den nye løsningen skulle være at brukeren selv kan lagre hva fargene for de ulike øremerkene skal være, ettersom fargen vil avhenge fra gård til gård. Fargene skulle legges inn i en viss rekkefølge, der den første fargen i listen ansees som "hovedgården".

Implementasjonen av fargevelgeren vises i figur 16.

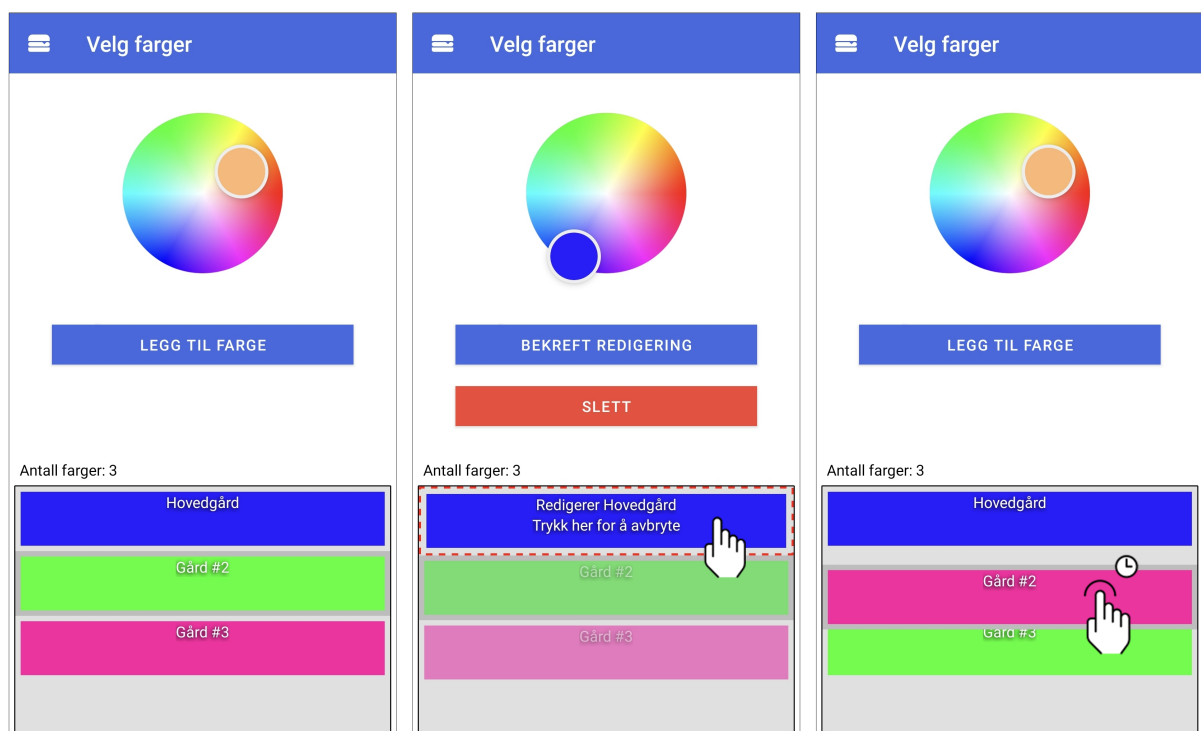


Figure 16: Skjermbilder - Fargevelger. Til venstre: fargevelgeren, ingen handlinger er gjort. I midten: brukeren har gjort et "short press" på en farge, og kan redigere (eller slette) denne fargen. Til høyre: brukeren gjør en "long press", og endrer rekkefølgen på fargene ved å dra på de fargede boksene

Brukeren legger til nye farger ved å flytte den lille sirkelen inne i den flerfargede, større sirkelen. Når brukeren er tilfreds med den nåværende fargeverdien, trykker brukeren på "Legg til farge", og fargen legges da til i listen over farger.

### 5.4.3 Endring av valg av øremerke i saueflokk

I sammenheng med fargevelgeren, ble vi også bedt om å gjøre en endring angående hvordan øremerker velges for en saueflokk (vedlegg D). Opp til nå hadde øremerke blitt valgt for

hver enkelte sau. Svein-Olaf bestemte at dette var unødvendig, og at øremerker heller skulle velges *per flokk*. Med dette ble det ment at man ikke skal velge hvilket øremerke hver enkelt sau har, men heller at alle øremerke-fargene som er representert i flokken, skal registreres. Det betyr at det ikke har noe å si om det er én eller ti sauer i en flokk med en spesifikk øremerke-farge. Det eneste som skulle ha noe å si er om det er ett eller flere av en øremerke-farge i en gitt flokk (i så fall skal dette registreres). Registrering av slips skulle ikke endres på.

På grunn av dette måtte vi nok en gang endre på siden for registrering av saueflokk. Resultatet vises i figur 17.

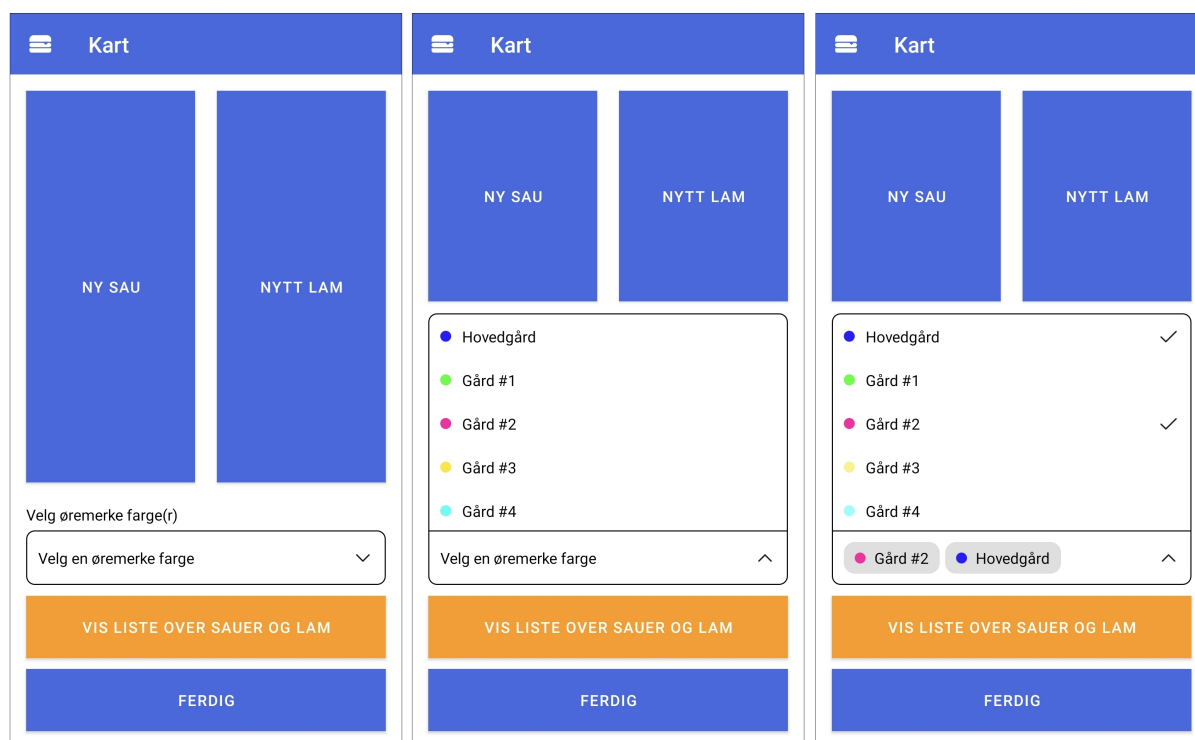


Figure 17: Skjermbilder - registrering av saueflokk. Venstre: den grunnleggende skjermen. Midten: brukeren har åpnet listen for valg av hvilke øremerket som er representert i flokken. Høyre: brukeren har valgt to farger fra listen.

Det nye designet innebærer at når man trykker på "Ny sau", blir brukeren kun bedt om å registrere hva slags farge sauen har på slipset. Ettersom lam ikke kan ha slips, blir brukeren ikke navigert til noen ny skjerm når de trykker "Nytt lam".

Fordi denne skjermen nå har en komponent for å velge farger for øremerkene, var det ikke lenger plass til listen over sauer og lam registrert hittil i flokken. Denne har derfor blitt flyttet ut av dette skjermbildet. For å vise listen, må brukeren trykke på den oransje knappen som sier "Vis liste over sauer og lam". Denne vises i figur 18.

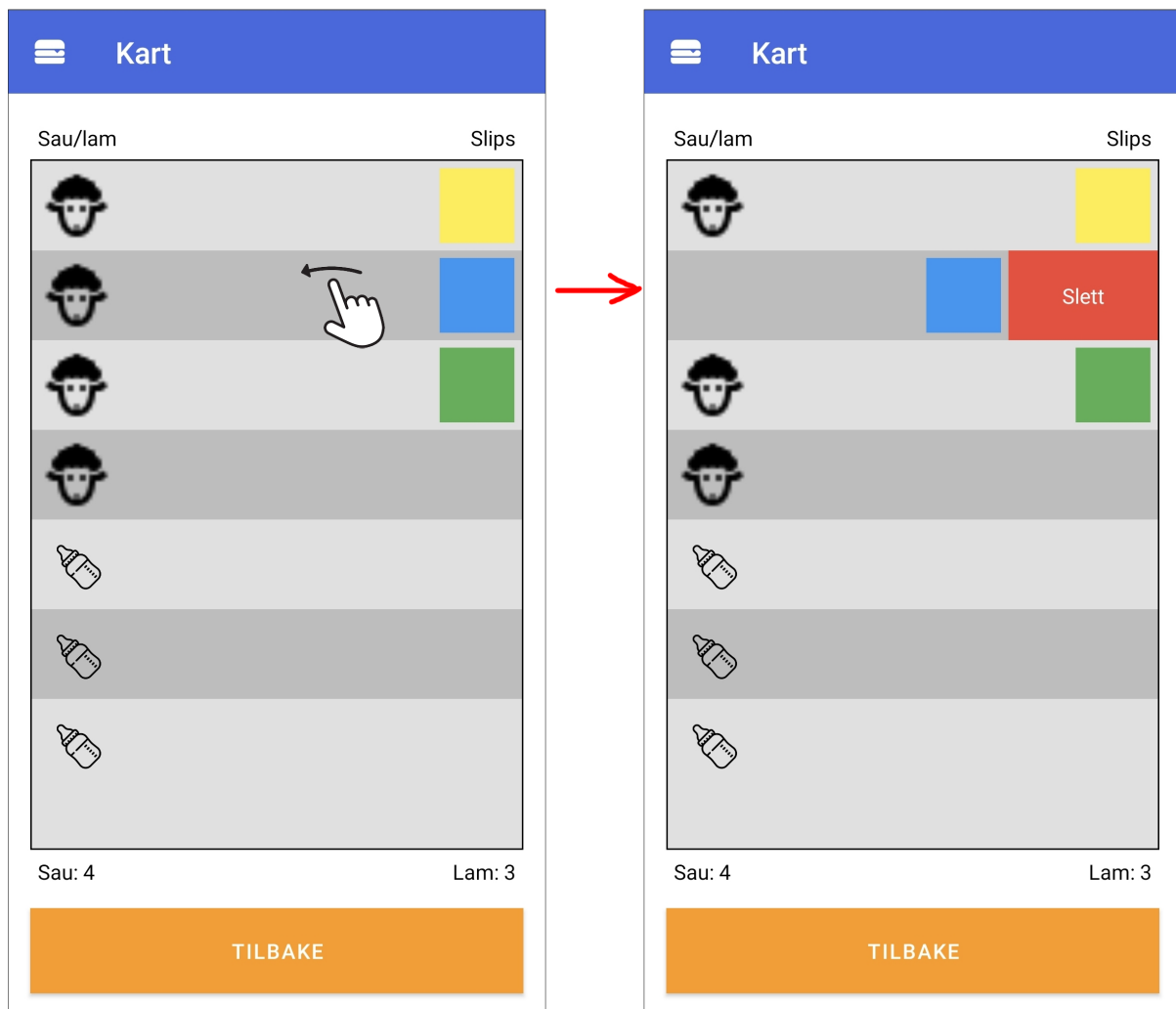


Figure 18: Skjerm bilde - liste over sauer i flokken. Brukeren kan sveipe for å avsløre slette-knappen. Denne fjerner den enkelte sauen/lammet fra flokken.

#### 5.4.4 Trekke linjer til observasjonsmarkører

I møtet 20. oktober (vedlegg D, side 2) ble vi også bedt om å legge til at for hver markør for observasjoner som blir plassert, skal det trekkes en linje fra der brukeren stod når de plasserte markøren, til der markøren er. Når en observasjon blir redigert, skal det da trekkes en linje fra hvor brukeren stod når redigeringen skjedde til den relevante markøren. Linjen til en "ny" markør, og linjen som viser at en redigering har blitt gjort, skal ha forskjellig stil. Implementasjonen av dette vises i figur 19.



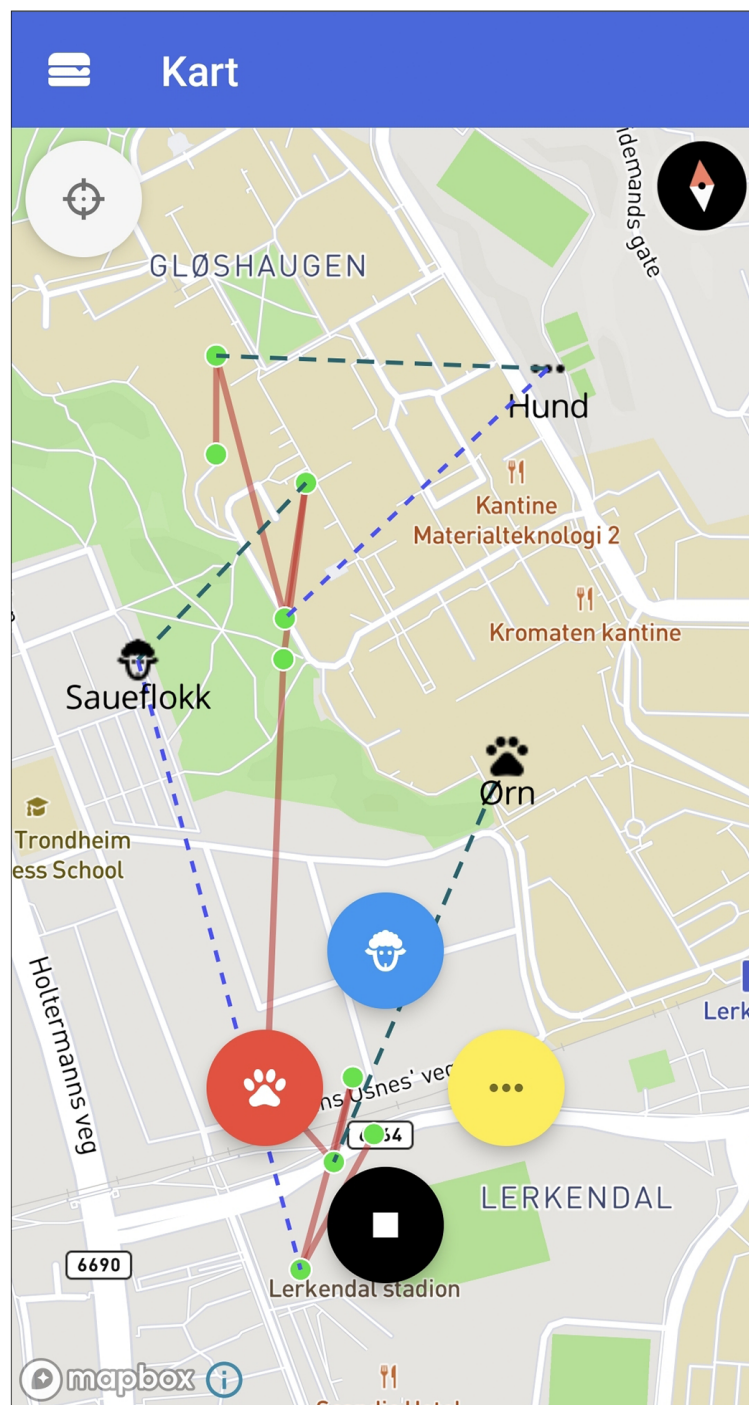


Figure 19: Skjerm bilde - linjer trekkes til markører.

Linjene som trekkes til der markørene først ble opprettet har en mørk turkis farge og er stiplede. Linjene som trekkes fra brukerens posisjon når de redigerte en observasjon har en lysere, lilla-blå farge, og er stiplet med mindre mellomrom. Fargene ble valgt fordi de har en god kontrast til bakgrunnskartet, samtidig som de to linjefargene er lette å skille fra hverandre.



#### 5.4.5 Rovdyr og "annet"

I løpet av utføringen av prosjektet fikk vi tilbakemelding på hva slags dyr som burde være valgmuligheter ved observasjoner av "Rovdyr" og "Annet"-kategoriene, basert på hva Svein-Olaf mente var mest hensiktsmessig (vedlegg B og D (side 2)). Den endelige listen over hva som kan registreres for hver kategori er:

##### Rovdyr

- Jerv
- Bjørn
- Ulv
- Ørn
- Annet\*

*\* Om brukeren velger "Annet"-verdien for rovdyr kan de skrive fritekst for å beskrive hvilket rovdyr de så.*

##### Annet

- Elg
- Hoggorm
- Hund
- Reinsdyr
- Annet

Når brukeren gjør en observasjonen innenfor "Annet"-kategorien kan de valgfritt legge til fritekst som beskriver observasjonen.

## 6 Brukertesting og Tilpasninger

Det ble utført brukertesting i to forskjellige stadier av prosjektet. Resultatet av disse, samt tilpasningene som ble gjort på bakgrunn av brukertestene, utredes her.

### 6.1 Om utføring av brukertesting

Brukertestene ble utført med basis i teoriene fra faget TDT4180 Menneske-maskin-interaksjon på NTNU Gløshaugen. Utføring av brukertesting ble også basert på teori fra boka *Interaction Design: Beyond Human-Computer Interaction*[9].

Brukertestene startet med at vi beskrev hva applikasjonen er for og hva testingen ville gå ut på. Brukerne ble informert om at det er *løsningen* som testes, ikke brukerne. Brukerne ble fortalt at de skulle gjøre spesifikke oppgaver som representerer en eller flere funksjonaliteter i applikasjonen. Brukerne måtte selv finne ut hvordan de skulle interagere med applikasjonen for å utføre oppgavene, og måtte selv bestemme når de mente de var "ferdige" med en gitt oppgave. Brukerne fikk så utdelt neste oppgave, og dette ble repetert helt til alle brukeroppgavene var gjennomgått.

For at vi som observatører av testene skulle lære så mye som mulig fra brukernes handlinger, ble brukerne oppfordret til å tenke høyt når de utførte oppgavene. Detaljer rundt brukernes måte å løse oppgavene på ble notert av oss. Etter alle oppgavene var gjennomgått og testingen var over, hadde vi en diskusjon med brukerne om hva de tenkte om applikasjonen og hvilke forbedringsforslag de eventuelt hadde.

Et prinsipp i faget TDT4180 Menneske-maskin-interaksjon, med inspirasjon fra brukervennlighetseksperter som Jakob Nielsen, er at man ikke trenger mer enn fem testere for å utføre brukertesting som gir betydningsfulle resultater[10]. Vi har derfor forsøkt å teste på opptil fem personer. Dette lot seg uheldigvis ikke gjøre, men vi fikk utført tester på fire personer, som fortsatt skal gi gode resultater i følge Nielsen.

### 6.2 Første Runde Brukertesting

De første brukertestene ble utført mot slutten av oktober. I dette stadiet var mesteparten av det som var krav til applikasjonen på plass, med unntak av støtte for nedlasting av kartregioner for offlinebruk. Derfor ble ikke nedlasting av kartdata et aspekt som ble sett på i denne runden av brukertesting.

I den første runden med brukertesting ble applikasjonen testet på to personer, begge studenter på dataorienterte linjer på NTNU.

### 6.2.1 Oppgaver for første brukertest-runde

Oppgavene brukerne måtte utføre i første runde brukertest vises i tabell 1.

Table 1: Oppgaver for første runde brukertester.

Oppgave ID	Beskrivelse
1	Som en uregistrert bruker, vil jeg registrere meg.
2	Jeg vil gi tillatelse til at appen sporer lokasjonen min.
3	Velge farger. Hovedgård: blå. Gård #2: grønn. Gård #3: lilla
4	Jeg vil starte en ny tilsynsrunde.
5	(Legg mobilen i lommen og gå rundt)
6	Jeg har observert en bjørn, og vil plassere den på kartet.
7	Jeg har observert en saueflokk, og vil plassere den på kartet.
8	(Hold opp bilde (figur 20)) Det er sju sauer i flokken. Registrer alle.
9	Sauene i flokken har totalt tre forskjellige øremerker. Registrer alle disse.
10	Jeg har registrert alt jeg ønsker fra flokken. Lagre flokken.
11	Jeg har observert en hund, og vil plassere den på kartet.
12	Jeg er ferdig med tilsynsrunden min.
13	Jeg har gjort en feil - jeg vil redigere tilsynsrunden min. Det viste seg at jeg så syner, og registrerte en sau for mye. Jeg vil fjerne en av saueene som har blått øremerke og rødt slips fra flokken.
14	I etterkant kom jeg også på at jeg under samme tilsynsrunde så en ørn, men glemte å plassere den på kartet. Plasser en ørn på kartet.
15	Jeg er fornøyd med endringene mine, og kan avslutte redigeringen.
16	Jeg er ferdig på jobb i dag, og ønsker å logge ut.



Figure 20: Bilde som ble brukt under brukertesting.



### 6.2.2 Resultater fra første runde brukertestning

Hvilke brukbarhetsproblemer som oppstod under utførelsen av spesifikke oppgaver i brukertestene, vises i tabell 2.

Table 2: Brukernes utfordringer under første runde brukertestning.

Oppgave ID	Beskrivelse av oppstått brukbarhetsproblem
3	Skjønner ikke hva "farger" trengs for.
8	Forvirring mellom slips og øremerke. Tror de registrerer øremerke når de egentlig registrerer slips.
8	La inn feil slipsfarge på sau -> sliter litt med å finne ut hvordan dette kan korrigeres. Går senere inn på listen over sauer i flokken. Skjønner ikke at man kan slette (vet ikke at det trengs swipe-handling).
13	Finner fort listen over sporinger, men skjønner ikke hvordan man kan gjøre noe med hver enkelt sporing i listen. Samme problem med manglende forståelse av at man kan swipe som i oppgave #8.

Det viktigste problemet som oppstod, som skjedde både i oppgave #8 og #13, var at brukerne ikke forstod at det var mulig å gjøre en "sveip"-handling på elementene i ulike lister for å avsløre knapper. Et eksempel på slike sveipe-handlinger ble vist i figur 15. Knappene som var nødvendige for å fullføre oppgavene var gjemt bak en sveipe-handling, og fordi brukerne aldri gjorde dette, lot oppgavene seg ikke gjennomføre. Vi opplevde at brukerne prøvde å bare trykke på elementene i listen, istedenfor å gjøre en sveipe-handling på dem.

Sveipe-handlinger er rimelig utbredt på applikasjoner kjørende på iOS, men er mindre utbredt på Android-applikasjoner. Dette er også gjeldende for applikasjoner som eksisterer på begge plattformer[11].

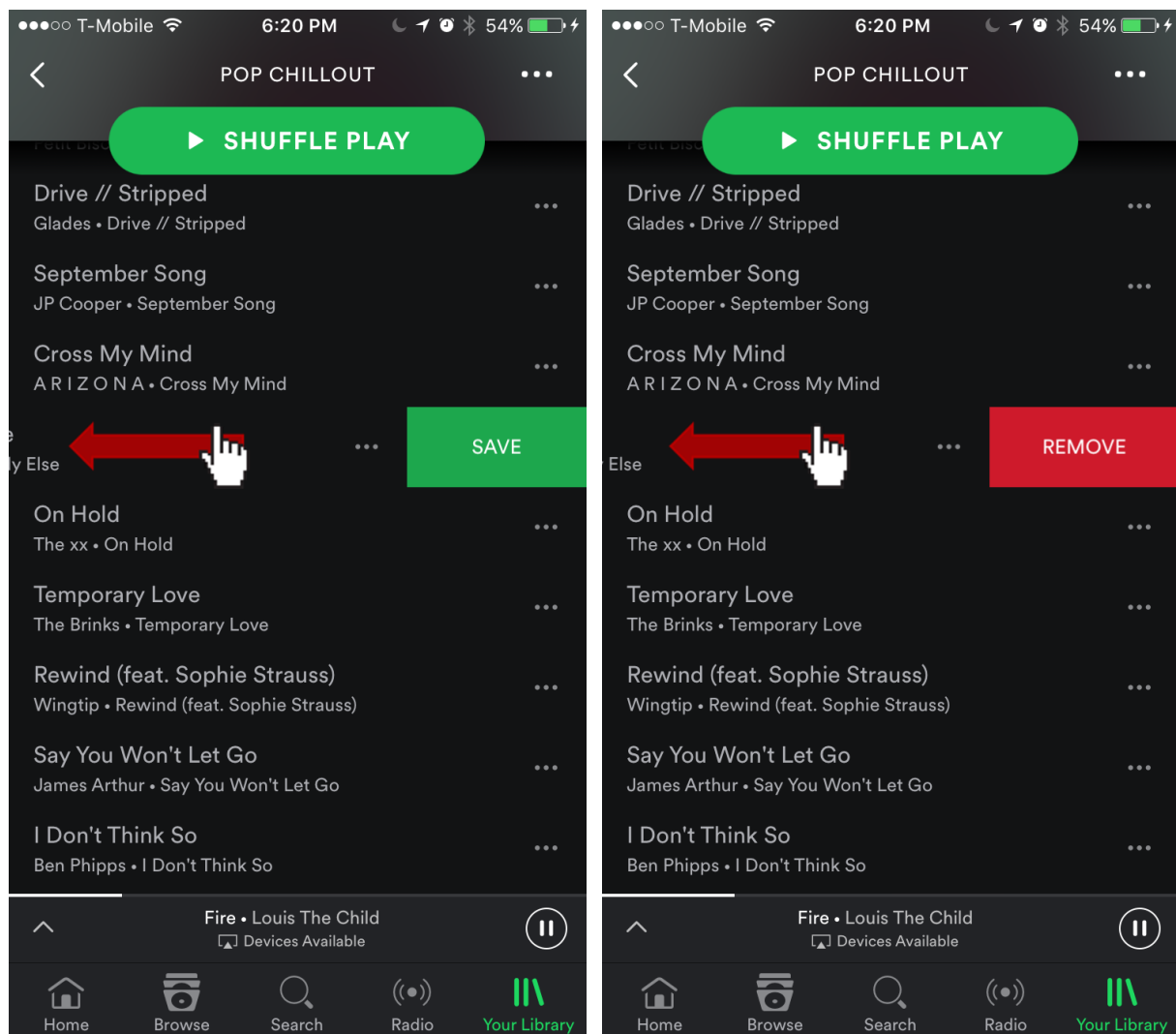


Figure 21: Skjerm bilde av applikasjonen Spotify kjørende på iOS, tatt fra artikkel [11]. Sveip-handlinger avslører ulike handlinger på den enkelte sangen i listen av sanger. Tilsvarende applikasjon kjørende på Android tillater ikke sveipehandlinger.

Dette er med på å forklare hvorfor dette brukbarhetsproblemet oppstod. Testbrukerne våre var Android-brukere, og var ikke vant til bruk av sveipe-handlinger. De forventet ikke at sveipe-handlinger var mulig, og forsøkte derfor ikke å sveipe. De forsøkte istedenfor å bare trykke, men dette hadde ingen effekt.

I oppgave #3 ble brukerne bedt om å lagre spesifikke farger. Én bruker slet med å forstå hvorfor de måtte gjøre dette. På skjermen hvor farger registreres (sett i figur 16), var det ikke noe som tydet mot hva fargene skulle brukes til.

### 6.2.3 Endringer etter første runde brukertesting

For å løse sveipe-problemet, gjorde vi at å trykke på et element automatisk "sveipet" til siden, slik at det å bare trykke på et element avslører de skjulte knappene. Brukeren kan derfor enten sveipe eller trykke, og det vil ha lik effekt.

For å bedre antyde til at fargevelgeren er for å angi øremerke-fargene, endret vi tittelen på siden fra "Velg farger" til "Velg farger (øremerke)".

Problemet med at én bruker trodde da la inn øremerke når de egentlig registrerte slips (oppgave #8), mener vi ikke er et reelt problem. Når brukeren registrerer slips, står det "slips" på skjermen for å minne brukerne på hva de nå registrerer. Å velge øremerkefarger gjøres ikke for hver sau, men for hele flokken. Over fargelisten står det også "Velg øremerke farge(r)". Vi mener derfor at vi har tydeliggjort og hvilken handling man gjør for både valg av slipsfarge og øremerke-farger, og at dette kun var et resultat av at brukeren ikke fulgte så godt med. Derfor synes vi det ville vært unødvendig å endre oppsettet på hvordan dette gjøres.

## 6.3 Andre Runde Brukertesting

Andre runden brukertesting ble utført i midten av november, rett etter at endringene fra første runde med brukertesting hadde blitt implementert, og funksjonalitet for bruk av "offline-kart" var lagt til. I andre runden hadde vi også to personer som testbrukere, som begge er studenter på datarettede linjer på NTNU. I denne fasen av testing var all funksjonalitet på plass, og vi fikk derfor testet "hele" appen. Listen med brukeroppgaver ble derfor endret fra første testrunde.

### 6.3.1 Oppgaver for andre brukertest-runde

Table 3: Oppgaver for andre runde brukertester.

Oppgave ID	Beskrivelse
1	Som en uregistrert bruker, vil jeg registrere meg.
2	Jeg vil lagre et kartområde for offlinebruk. Området må dekke hvertfall Moholt og Midtbyen.
3	Jeg vil gi tillatelse til at appen sporer lokasjonen min.
4	Lagre disse øremerke-fargene: Hovedgård: blå. Gård 2: grønn. Gård 3: mørk lilla.
5	Jeg vil starte en ny tilsynsrunde.
6	(Legg mobilen i lommen og gå rundt)
7	Jeg har observert en bjørn, og vil plassere den på kartet.
8	Jeg har observert en saueflokk, og vil plassere den på kartet.
9	(Hold opp bilde (figur 20)) Det er sju sauer i flokken. Registrer alle.
10	Sauene i flokken har totalt tre forskjellige øremerker. Registrer alle disse.
11	Jeg går mer rundt, og mister netttilkobling. (Aktiver flymodus på mobilen for å simulere at du mister nett).
12	Velg et offline-kartområde som dekker der jeg går.
13	Jeg har observert en hund, og vil plassere den på kartet.
14	Jeg er ferdig med tilsynsrunden min.
15	Jeg har gjort en feil - jeg vil redigere tilsynsrunden min. Det viste seg at jeg så syner, og registrerte en sau for mye. Jeg vil fjerne en av sauene som har blått øremerke og rødt slips fra flokken.
16	I etterkant kom jeg også på at jeg under samme tilsynsrunde så en ørn, men glemte å plassere den på kartet. Plasser en ørn på kartet.
17	Vi er fornøyde med endringene våre, og kan avslutte redigeringen.
18	(Koble til nett igjen).
19	Jeg er ferdig på jobb i dag, og ønsker å logge ut.

### 6.3.2 Resultater fra andre runde brukertest

I brukertestene i andre runde fikk vi oppdaget et par tekniske problemer som vi ikke selv hadde klart å oppdage i egen testing. Disse var ikke brukbarhetsproblemer, men heller uforventet oppførsel knyttet til koden som vi ikke selv hadde oppdaget. Disse var knyttet til bruk av applikasjonen uten netttilkobling, som ikke var en faktor under første runde med brukertest. Disse problemene ble fikset og er ikke interessante nok til å gå inn på her.

Et brukbarhetsproblem vi oppdaget var at brukerne forsøkte å bruke "tilbake"-knappen på mobilen. I runde to var begge brukerne vant til Android-mobiler, og testene ble også utført på Android-mobil.



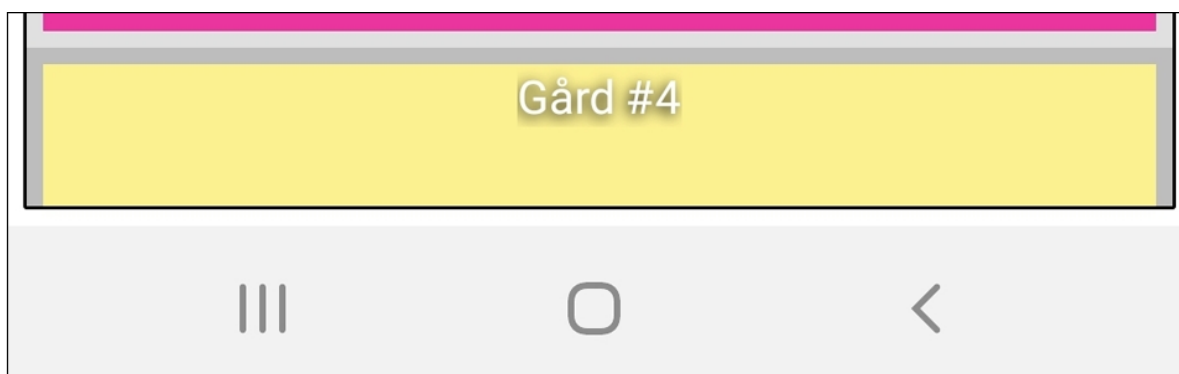


Figure 22: Skjerm bilde - navigasjonsmenyen på bunnen av en Android-telefon. Knappen helt til høyre er tilbakeknappen.

Figur 22 viser hvordan navigasjonsmenyen ser ut på en Android-mobil. Ved bruk av applikasjonen vår på en Android-telefon er denne alltid synlig fordi den er knyttet til operativsystemet til mobilen, og dens tilstedeværelse styres ikke av applikasjonen.

Vi hadde tidligere skrudd av at tilbakeknappen hadde noen funksjon i applikasjonen, for å forhindre tilfeller der brukeren for eksempel logger inn og så trykker på tilbakeknappen, som da ville ha navigert dem tilbake til innloggingsskjermen, noe som er uønsket oppførsel. Fordi å bruke tilbakeknappen er svært naturlig for Android-brukere, prøvde testbrukerne å bruke denne knappen i tilfeller der de ville navigere seg én skjerm bakover - men siden knappen var skrudd av, hadde dette ingen effekt.

Vi skjønnte derfor at å skru av tilbakeknappen helt ikke gir god brukerflyt for Android-brukere, ettersom de bruker denne knappen nærmest instinktivt.

Et annet brukbarhetsproblem som ble oppdaget var at én bruker, som la til fargene i feil rekkefølge, ikke skjønnte at man kunne endre rekkefølgen på fargene i fargelisten ved å holde inne og dra. De prøvde å trykke på elementet i listen, men dette er knyttet til å en annen handling (redigere den spesifikke fargen). Brukeren skjønnte ikke at det var mulig å gjøre et "long press" på listen for å endre rekkefølgen.

### 6.3.3 Endringer etter andre runde brukertest

Vi valgte å endre bruk av tilbakeknappen slik at den ikke er helt skrudd av, men heller kun deaktiveres i de scenariene der det gir uønsket oppførsel, slik som at brukere navigerer seg tilbake til påloggingsskjermen etter de har logget inn. Dermed kan brukerne nå bruke tilbakeknappen i de fleste tilfeller.

For å bedre tyde til at det er to forskjellige handlinger brukeren kan gjøre på listen av farger (redigere farge ELLER endre rekkefølge på listen), la vi til forklarende tekst for å kommunisere dette til brukeren. Dette vises i figur 23.



Figure 23: Skjerm bilde - oppdatert fargeliste. Brukeren får nå instruksjoner om hvilke handlinger som er tilgjengelig.

## 6.4 Konklusjon av brukertestning

Etter andre runde brukertestning hadde vi avdekket og fikset flere ulike brukbarhetsproblemer som testbrukerne støtte på under testing. Etter andre runde hadde vi ikke tid til flere iterasjoner av brukertestning, men vi så heller ikke mye behov for å utføre flere brukertester på applikasjonen da vi mener vi hadde oppnådd et godt resultat når det kommer til brukeropplevelse på applikasjonen. Ettersom alle de planlagte funksjonalitetene var implementert når andre brukertest ble utført, hadde det heller ikke vært noen nye funksjonaliteter å teste i en eventuell tredje runde brukertestning.

Etter at endringene basert på andre runde brukertestning hadde blitt gjort, la vi dermed en stopper for utviklingen av applikasjonen.

## 7 Konklusjon og Refleksjon

Vi har nå vært gjennom en prosess der vi har designet, utviklet og testet en applikasjon for å utføre tilsynsrunder av sau. Prosessen har vært preget av hurtig prototyping, smidig utvikling og kontinuerlige endringer basert på tilbakemeldinger. Det endelige resultatet er noe vi er godt fornøyd med, og dens kvalitet har vært vurdert både av oss, veileder Svein-Olaf gjennom demonstrasjoner av appen, og testbrukernes tilbakemeldinger.

Vi kom inn i prosjektet med en forventning av at hovedutfordringene ville være knyttet til den tekniske utføringen av prosjektet, spesielt knyttet til bruk av kart og sporing av brukerens lokasjon gjennom enhetens GPS. Det viste seg likevel at de største utfordringene kom i form av å utforme gode brukergrensesnitt for mobilenheter, ettersom dette var et helt nytt fagområde for oss.

Vi synes prosjektet har vært svært lærerikt, og applikasjonen gir oss et solid grunnlag for arbeidet med hoved-masterprosjektet til våren. Det viktigste vi har lært er om bruk av React Native for mobilutvikling, samt hvordan brukere av henholdsvis Android- og iPhone-telefoner har ulike forventninger til interaksjonsmuligheter med applikasjoner på enhetene deres.

## 8 Videre arbeid

Applikasjonen er kun ett aspekt av den større masteroppgaven innenfor manuell sanking av sau som skal fullføres våren 2022. Selv om vi ikke har fått hele omfanget av selve masteroppgaven definert for oss enda, har vi fått oppgitt hva den kommer til å gå ut på (vedlegg E). Mens applikasjonen støtter registrering av forskjellig data under tilsynsrunder, vil selve masterprosjektet til våren basere seg på å lage et PC-basert "dashbord" der data som har blitt samlet inn fra brukere av mobilapplikasjonen kan benyttes til forskjellige formål. Dette kan for eksempel være å se på bevegelsesmønstrene til sauene over lengre perioder, eller gjøre en vurdering av trusselen rovdyr utgjør i et område. Mens applikasjonen fra dette fordypingsprosjektet sin tiltenkte brukergruppe er observatører som jobber på vegne av bøndene, er det bøndene selv som skal ta i bruk dashbord-løsningen.

Noen av oppgavene vi sannsynligvis vil jobbe med i neste delen av masterprosjektet vil derfor være:

- Å lage et grensesnitt som gir god oversikt over ulik informasjon som er samlet inn.
- Vurdere hva slags informasjon som er nyttig for løsningen, og hvordan den burde tolkes og presenteres for brukeren.
- Gjøre testing av applikasjonen i en "ekte" tilsynsrunde for å verifisere dens brukskvalitet i et realistisk bruksscenario.
- Gjøre brukertesting på den PC-baserte løsningen for å sikre god brukeropplevelse.

## References

- [1] P. Christensson. (2020). “Backend definition,” [Online]. Available: <https://techterms.com/definition/backend> (visited on Dec. 6, 2021).
- [2] P. Christensson. (2020). “Frontend definition,” [Online]. Available: <https://techterms.com/definition/frontend> (visited on Dec. 6, 2021).
- [3] P. Christensson. (2013). “Framework definition,” [Online]. Available: <https://techterms.com/definition/framework> (visited on Dec. 5, 2021).
- [4] Lovdata, *Forskrift om velferd for småfe*. [Online]. Available: <https://lovdata.no/forskrift/2005-02-18-160/%C2%A719> (visited on Oct. 14, 2021).
- [5] Lovdata, *Forskrift om erstatning når husdyr blir drept eller skadet av rovvilt*. [Online]. Available: <https://lovdata.no/dokument/SF/forskrift/2014-05-30-677> (visited on Oct. 14, 2021).
- [6] Rovbase. (2020). “Erstatning for sau,” [Online]. Available: <https://www.rovbase.no/erstatning/sau> (visited on Nov. 1, 2021).
- [7] L. E. Vassbotn and A. J. Jonassen, *Digitalt møte med svein-olaf 25. september, vedlegg A*, 2021.
- [8] International Data Corporation, *Smartphone market share*, 2021. [Online]. Available: <https://www.idc.com/promo/smartphone-market-share> (visited on Dec. 6, 2021).
- [9] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2002, ch. 14.
- [10] J. Nielsen, *Why you only need to test with 5 users*, <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>, Hentet 06/12/2021, 2000.
- [11] A. Li. (2007). “Using swipe to trigger contextual actions,” [Online]. Available: <https://www.nngroup.com/articles/contextual-swipe/> (visited on Dec. 7, 2021).

## Appendix A   Møte 25. september 2021

# Notater møte med Svein-Olaf 25.08.2021

---

Vi velger utviklingsmiljø selv.

Det skal være kart i appen. Kun Norge skal på kartet. En app for Norge.

GPS-støtte trengs. Systemet må kunne tracke hvor du går selv om mobilen er pakket ned i mobilen.

To prosesser; en som styrer frontend som viser et GUI. Og en egen backend som tracker hvor du går.

Trenger kun trekke rette linjer mellom punktene du går.

Lokasjon registreres hvert 30s eller hvert minutt.

Når du har funnet et sted for å registrere (flokk) med sau. Du registrere da en linje fra DEG til hvor sauene er. Altså du registrer hvor sauene er.

To metoder for dette: Kors i midten, og du flytter kartet. En annen metode er at kartet er statisk, og at du flytter krysset.

Når du har registrert hvor sauene er, skal du registrere sauene. Hvis sauene er mer enn 50m fra deg, klarer du ikke å registrere de fine detaljene. Appen skal selv vite om du er over 50m fra sauene.

Når du først har sett sauene, registrere antall voksne og barnesau (lam). Noen bønder registrerer farge på sauene, men vi trenger det ikke.

Det du skal registrere er øremerke på sauene. Du må forvente at du sitter med kikkert hele tida (ikke tar ned kikkert) for å holde konstant sightline med sauene og ikke forveksler sauer. Når vi snakker om kikkert mener vi en faktisk kikkert, ikke noe digitalt.

Sauer har ett øremerke i hvert øre, samme merke. Tilhørighet er fargekodet per sau, og du må telle antall sauer med x farge.

Blandingsflokker: flere gårder sine sauer i samme flokk. Flere gårder kan slippes ut sauer samme sted samme tid.

Lam følger ikke nødvendigvis sin egen mor.

Lam kan "forveksle" flokk og bli en del av en annen flokk en mor.

Alle mammasauer har ett "slips". Fargen på slips forteller hvor mange lam de har. Gult f.eks betyr 2 lam. Teller opp lam hvor mange ser, og så teller du opp i forhold til slips.

Bonden vil ALLTID vite om det har vært observert om det mangler lam, det er indikator på rovdyr.

Da har du fått registrert det du skal.

Registrerer bare effekter (på sau, som skader og hva skaden er). Bonden selv gjør tolkning om det kan være rovdyr.

Når du er ferdig, gjør du ny runde registrering. Det kan overlappe med sauer mellom ulike observasjonsrunder. Korrigering kan skje.

Sauer kan gjemme seg og senere bli synlig.

Registrering av rovdyr: kan se jerv, ulv og bjørn. Gaupe kan ta sau, men veldig sjeldent du ser gaupe.

"Andre registreringer": Død elg, reinsdyr, hoggorm.

Må ha mulighet for litt tekstlig registrering også.

Ingen bønder er så systematiske på å få ned denne infoen her. De noterer kanskje bare 2 stikkord på papirlapper.

---

Start med å se på utviklingsmiljø, og utviklingsmiljø på kart. Legge ting på overlay over et kart. GPS-posisjoner på en overlay.

---

Høstprosjekt: gå ut i felt og registrere.

Vårprosjekt: helt andre krav enn beskrevet over, denne delen skal være for bonden. Én applikasjon over hvor han har sine sauer, og hvor de beveger seg gjennom sesongen. Hvor finner jeg sauene når jeg skal hente dem inn igjen? Har mange sauer blitt drept? Burde jeg skyte rovdyr inne på området mitt?

"De som sanker inn, og fellessystemer".

---

Høstprosjekt: standalone, anta ikke nettilgang. Last ned kart på forhånd, og last opp data til sky senere. Mange områder uten dekning, spesielt der sauene går.

Må kjøre på mobiltelefon. Bør være flerplattform, kan være enkeltplattform kun vis det er veldig god grunn for det.

---

Synkronisering av notater kan enten være automatisk eller manuelt aktivert.

Skalering for ekstremt mange brukere er ikke viktig

Veldig få som gjør denne typen arbeid samtidig (observasjoner).

Neste møte om to uker, annenhver uke blir det møte. Om vi ønsker oftere så kan vi endre på det.

---

Det er tiltenkt brukertesting. Tester ikke på bønder, men på studenter. Studenter har mindre kunnskap om gårdsarbeid.

Utvikle fort, ha god tid til testing og korreksjon.

---

Tiltent en rapport for innlevering som blir godkjent/ikke godkjent. Hvordan det blir nå i høst er ukjent, men antar samme opplegg. Forvent å skrive en rapport.

Denne rapporten er ikke en del av selve masteren.



## **Appendix B   Møte 29. september 2021**

# Møte med Svein-Olaf

Onsdag 29. september

## Agenda:

- **Få godkjent prototype-designet, spørre om tilbakemelding.**
  - Rework hvordan man registrerer saueflokk
    - under 30 meter fra flokken: du skal kunne registrere alt “uten å se på mobilen” -> store knapper, haptisk feedback?
    - over 30 meter fra flokken: kun registrere at det finnes en flokk, ingen detaljer?
  - Alle andre aspekter av designet er flott, Svein-Olaf er fornøyd
- **Spørre om gode kilder angående sauehold. Trenger dette til rapport.**
  - Kilder om gårdsbruk av sau: “myndigheter vil at sauebonder skal følge opp saueflokker 1 gang i uka”. Dette er for å kunne dokumentere krav til erstatning ved rovdyrangrep. Finn en artikkel/side om dette.
- **Engelsk eller norsk på rapporten?**
  - Norsk er lov på rapporten, velg hva du vil. Selve masteren kan også skrives på norsk.

## Notater:

- Jerv er hoved-rovdyr. Ulv og bjørn også. Rev er ikke farlig dyr for sau, fjern rev som valg.
- Fint å kunne registrere hunder som en “annen” registrering.

## Appendix C   Møte 6. oktober 2021

# Master møte med Svein-Olaf

Onsdag 6. oktober  
Kun Andreas, Lars Erik er syk

## Agenda:

- Få godkjent nytt design av sheep herd registrering. Vis de forskjellige mulighetene.
- Er det lov for oss å anta at brukeren først logger inn med nettilgang? Antar det er en grei antagelse å ha fordi du må laste ned kart før bruk uansett.
- Firebase

## Forslag (krever godkjenning av Svein):

- forskjellige ikon for saueflokker
  - ett ikon for når alt er OK med saueflokk-detalljer
  - ett annet ikon for når noe skurrer med detaljene om flokken
  - ett siste ikon for flokk uten detaljer
- Varsle hvis det ikke er samsvar på antall lam indikert av slips, og antall sau som har blitt talt

## Til rapport:

- Vi tenkte først å bruke Micronaut (Kotlin) backend, men tenker nå å bruke Firebase istedenfor. Det har offline caching av data som fungerer bra for vår brukscase. Ikke-relasjonell database er også bra for vårt bruk.

## Notater under møte:

- Neste onsdag - ingen møte. neste møte 20. oktober
- Design OK - høyre venstre / tap tap
- Ikke vits med forskjellige ikon
- kan anta at bruker har internett første gang de registrerer/bruker appen
- Ser bra ut!

## Appendix D   Møte 20. oktober 2021

# Møte med Svein-Olaf

20. oktober 2021

## Agenda:

- Appen nærmer seg ferdig (dog forbedringer kan skje etter brukertesting)
- Trenger bekreftelse på at valgene ved registrering samsvarer med det Svein-Olaf ønsker. Er rovdypene riktige? Er “annet” slik han ønsker?

## Spørsmål:

- I tidligere møte ble øremerke beskrevet slik: “To øremerke-farger som er vanlige, en som er mindre vanlig, en som er veldig sjelden. “
  - Hvilke er disse vanlige fargene? Vi har kun eksempel-farger på appen nå, men vil gjerne vite de “ekte” fargene.
    - **Svar:** varierer fra gård til gård. En gård vil velge annen farge fra naboene.
    - **Bonden må få lov til å velge fargene selv.** Du bestemmer på forkant hva de mulige fargene er.
    - Når man er på tilsynsrunde: du har “hovedgården”, og så opptil 4 eller 5 andre farger som kan påtreffes, men er veldig sjeldne. Viktig med en rekkefølge. Gårdens eget kommer først, dette er viktig.
    - Du skal kunne legge inn opptil 5 farger, og så bestemme i etterkant rekkefølgen på fargene.
- Hvor mange muligheter er det for slips, mao. hvor mange mulige farger? Leste i en artikkel et sted at kun opptil 2 lam som følger med søye på beite er vanlig.
  - **Svar:** En søye ute på beite prøver man å slippe å ha mer enn to lam medfølgende. De øvrige lammene adopteres bort til mødre med 1 lam eller ingen lam. De ekstra lammene ender opp på inngjerdet område, en “barnepark”.
  - Sjeldent mer enn to, men må støtte flere.
  - Slipsfarger er samkjørt på landsbasis. Tidligere var det kommunebasis.
  - Kan finne hva fargene er på nett.
  - Eget slips for ingen lam.
  - Det burde være eget slips for “har ikke slips på”.
  - Bukker (“mannesau”) er ikke ute på beite.
- Har du noen formeningar/krav på hvordan vi skal gjøre brukertesting?
  - **Svar:** Gjør som dere vil.

## Notater:

Notere slips er veldig viktig.

Øremerke registrering er ikke viktig per sau. **Øremerke er per-flokk.**

Normalt er det bare 1 type øremerke i en sau. Av og til er det 2. “Omtrent aldri” flere enn 2.

Trygt å anta at man ser på skjermen når man registrere øremerket. Men ser ikke ofte på skjermen når det gjelder slips

**Du vil markere hvor du stod når du gjorde en spesifikk markering. Også når du har redigert vil du ha en linje fra der du er til den du redigerte på.** (Linje fra når du originalt plasserte markøren, og når du redigerte den, er altså to forskjellige “stiler”).  
Nøyaktig *hva* endringer er, bryr vi oss ikke om å registrere.

Få med “ørn” som et rovdyr. Vurdere å skille mellom kongeørn og havørn. Har du dårlig plass så bare ta ørn.

GI testbruker et bilde av saueflokk med slips og øremerke.  
Eventuelt tegn det selv.

Trenger maks fem personer til brukertesting. Kun tre personer på brukertest er også greit.

## **Appendix E   Møte 24. november 2021**



# Møte med Svein-Olaf

24. november 2021

## Agenda:

- Vi har gjort ny runde brukertesting
  - Færre problemer med brukbarhet
  - Et par tekniske problemer som var vanskelig å avdekke
- Rapport:
  - Trenger vi GANTT-diagram?
  - Forskningsplan?
- Trenger ikke møte neste uke
  - Ta neste møte 8. desember? Dette blir da aller siste møte.

## Notater:

- Rapport :
  - Skriv planer om prosjektet til våren.
    - Problemstillingen, eller problemstillingene, vi vil ha til vår.
      - “Videre arbeid”-seksjon med fremtidige mål
      - Trenger ikke gå veldig mye inn på det. Mest nevnt at det er bonden som skal bruke det og hva det skal brukes til. Trenger ikke si mye om hvordan vi skal løse det.
    - Trenger ikke GANTT eller forskningsplan.
    - Rapporten skal være minst 5 sider. Trenger ikke overtenke rapporten. Bare lever noe og ha med de viktigste seksjonene (bakgrunn, utvikling, testing osv.).
- **Neste semester:**
  - Da er det fokus på bonden.
  - Måte å se alle rapportene som har kommet inn. Kunne få en formening på “hvor har sauene vært, hvor er sauene, hvor er det sannsynlig at den kan være i fremtiden”. Har det vært rovdyrtilfeller? Skadd eller drept sau?
  - Dette skal være PC-basert, ikke en app.
- Send epost neste uke noen timer før møtet, og si at vi ikke trenger møte.
- Neste møte blir 8. desember.