

Nikolai Mork
Anders H. Rebner

Manuell oppfølging av sau på beite

Masteroppgave i Informatikk
Veileder: Svein-Olaf Hvasshovd
Juni 2022



Anders H. Rebner

Nikolai Mork
Anders H. Rebner

Manuell oppfølging av sau på beite

Masteroppgave i Informatikk
Veileder: Svein-Olaf Hvasshovd
Juni 2022

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

Abstract

Title:	Manual follow-up of sheep on pasture
Date:	06.06.2022
Participants:	Nikolai Mork Anders Hoelseth Rebner
Supervisor:	Svein-Olaf Hvasshovd
Employer:	NTNU
Contact:	Svein-Olaf Hvasshovd
Keywords:	Application development, Flutter, sheep, pasture, speech recognition, map
Number of pages:	118
Number of appendixes:	6
Availability:	Open

Abstract: The statutory follow-up of sheep on pasture is done by writing with pen and paper, which is demanding in various weather conditions, and results in a time-consuming transition from physical notes to a digital yearly report. The goal of this thesis is to increase the efficiency of these tasks through digitalization, and to explore whether using speech recognition to register observations while looking through binoculars is fitting. A system of a mobile application and a web application that partially automates the tasks was developed. Usability testing of the system shows that digital tools can improve the efficiency, and potentially the quality, of both follow-up and finalizing work, and that the use of speech recognition is fitting and has large potential.

Sammendrag

Tittel:	Manuell oppfølging av sau på beite
Dato:	06.06.2022
Deltakere:	Nikolai Mork Anders Hoelseth Rebner
Veileder:	Svein-Olaf Hvasshovd
Oppdragsgiver:	NTNU
Kontaktperson:	Svein-Olaf Hvasshovd
Nøkkelord:	Applikasjonsutvikling, Flutter, sau, beite, talegjenkjenning, kart
Antall sider:	118
Antall vedlegg:	6
Tilgjengelighet:	Åpen

Sammendrag:	Det lovpålagte oppsynet av sau på utmarksbeite har frem til i dag foregått ved notering med penn og papir, som er utfordrende i vær og vind, og som gjør overføringen fra notater til digital årsrapport tidkrevende. Denne oppgaven har som mål å effektivisere arbeidsoppgavene ved digitalisering, og å utforske om bruk av talegjenkjenning er hensiktsmessig for å registrere observasjoner uten å ta blikket ut av kikkerten. Et system av en mobilapplikasjon og webapplikasjon som delvis automatiserer oppgavene ble utviklet. Brukertesting av systemet viser at digitale verktøy kan effektivisere og potensielt øke kvaliteten på både oppsyn og etterarbeid, og at bruk av talegjenkjenning på oppsynstur er hensiktsmessig og har stort potensiale.
-------------	---

Forord

Denne rapporten er en masteroppgave i Informatikk med spesialisering i programvaresystemer hos Institutt for datateknologi og informatikk ved Norges teknisk-naturvitenskapelige universitet (NTNU).

Vi vil rette en stor takk til professor Svein-Olaf Hvasshovd for domenekunnskap og veiledning gjennom prosjektet.

Anders H. Rebner og Nikolai Mork, Juni 2022

Innhold

Abstract	iii
Sammendrag	v
Forord	vii
Innhold	ix
Figurer	xiii
Tabeller	xv
Kodelister	xvii
1 Introduksjon	1
1.1 Fagområde	1
1.2 Avgrensning	1
1.2.1 Dyrevelferd	1
1.2.2 Sau	2
1.2.3 Oppfølging av sau på beite	2
1.3 Oppgavebeskrivelse	3
1.4 Formål	5
1.4.1 Effektmål	5
1.4.2 Læringsmål	5
1.5 Målgruppe	5
1.6 Bakgrunn og kompetanse	5
1.7 Rammer	6
1.8 Rapportstruktur	6
2 State of the art	7
2.1 Dokumentasjon av dyreobservasjoner	7
2.1.1 Skandobs	7
2.1.2 Artsobservasjoner	8
2.1.3 Viltappen	8
2.2 Posisjonsrelaterte systemer	8
2.2.1 Nofence	8
2.2.2 NorTrace	9
2.2.3 Telespor	9
2.2.4 Findmy	9
2.2.5 Smartbjella	10
2.2.6 Oversikt over priser	10
2.3 Tilsvarende systemer	11

2.3.1	Beitesnap	12
2.3.2	"Lambo", Dysthe og Kjærstad, 2018	12
2.3.3	"Pecora", Alm og Gård, 2018	13
2.3.4	"Pecora", Schmidt-Hanssen, 2018	14
2.3.5	Halvorsen, 2019	14
2.3.6	Svendsen, 2019	15
2.3.7	"Sheepter", Flytøren, 2020	15
2.3.8	"Sauron", Abtahi og Gjeldseth-Borgen, 2021	16
2.3.9	Carlsen, 2021	17
2.3.10	Hyll og Torgersen, 2021	18
2.3.11	Styve, 2021	18
2.4	Konklusjon	19
3	Kravspesifikasjon	21
3.1	Funksjonelle krav	21
3.1.1	Use cases for webapplikasjon	21
3.1.2	Use cases for mobilapplikasjon	23
3.2	Ikke-funksjonelle krav	26
3.2.1	Generelle krav	26
3.2.2	Operasjonelle krav	26
3.2.3	Krav til leveranse	27
3.3	Mål for brukervennlighet	27
4	Teknisk design	29
4.1	Domenemodell	29
4.2	Overordnet arkitektur	31
4.3	Database	31
4.3.1	Hovedstruktur	32
4.3.2	Dokumentstruktur	33
5	Design av brukergrensesnitt	39
5.1	Egne ikoner	39
5.2	Webapplikasjon	40
5.2.1	Innlogging & opprettelse av brukerkonto	40
5.2.2	Min side	40
5.2.3	Oppsynsturer	48
5.2.4	Årsrapporter	52
5.3	Mobilapplikasjon	52
5.3.1	App-ikon	52
5.3.2	Programflyt	53
5.3.3	Innlogging & opprettelse av brukerkonto	55
5.3.4	Start oppsynstur	55
5.3.5	Hovedskjerm	60
5.3.6	Registreringsskjermer	61
5.3.7	Registrering med tale	66
5.3.8	Opplasting av oppsynstur	67
5.3.9	Innstillinger	69

5.4	Designvalg	69
5.4.1	Felles	69
5.4.2	Webapplikasjon	71
5.4.3	Mobilapplikasjon	71
6	Utviklingsprosess	73
6.1	Utviklingsmetodikk	73
6.1.1	Daily Standup/Morgenmøte	73
6.1.2	Sprint retrospective/Evalueringsmøte	73
6.1.3	Sprint planning/Planleggingsmøte	74
6.1.4	Kundemøte	74
6.1.5	Prosjekttavle og GitHub	75
6.1.6	Code review	75
6.2	Prototyping av Design	76
7	Implementasjon	77
7.1	Valgte teknologier og biblioteker	77
7.1.1	Flutter & Dart	77
7.1.2	Firebase	78
7.1.3	Biblioteker	79
7.2	Grov oversikt over widget-hierarki	80
7.3	Kart	83
7.4	Taleregistrering	85
7.4.1	Konfigurasjon av talegjenkjenning på enheten	85
7.4.2	Forbedring av treffprosent	86
7.4.3	Skjuling av knapper under taleregistrering	87
7.5	Registreringer	88
7.5.1	GPS-intervall	88
7.5.2	Abstrakt klasse	88
7.6	Oppsynstur uten nettverksforbindelse	88
7.7	Håndtering av data fra oppsynsturer	89
7.8	Årsrapport	90
7.9	Kodepraksis	90
8	Testing	91
8.1	Automatisert testing	91
8.2	Brukertesting	93
8.2.1	Prosedyre	94
8.2.2	Resultater	97
9	Diskusjon	103
9.1	Valg av teknologier	103
9.1.1	Rammeverk for frontend	103
9.1.2	Rammeverk for backend	104
9.1.3	Kart-bibliotek	104
9.1.4	Talegjenkjennings-bibliotek	104
9.1.5	Database	105
9.2	Brukertest	105

9.2.1	Deltakerveiledning	105
9.2.2	Avgrensning	105
9.2.3	Teknisk gjennomførelse	106
9.2.4	Læring	106
9.2.5	Feilmeldinger	107
9.2.6	Totalvurdering av resultater	108
9.3	Taleregistrering	108
9.4	Sauebønder som marked	109
9.5	Valg rundt utforming av systemet	110
9.5.1	Verdien av valg	110
9.5.2	Strømlinjeutforming og brukerprivilegier	110
9.6	Refleksjon over fremdrift	111
9.7	Mangler og kjente feil	112
9.7.1	Liste over kjente feil	112
9.8	Utbedringer og videre arbeid	113
9.8.1	Taleregistrering	113
9.8.2	Kombinasjon av systemer	114
9.8.3	Løsningsforslag på det som ble avdekket av brukertester	114
9.8.4	Ytterligere utbedringer	114
9.9	Kritikk av oppgaven	116
10	Konklusjon	117
10.1	Drøfting av resultater	117
10.1.1	Effektmål	117
10.1.2	Læringsmål	117
10.1.3	Resultatmål	117
10.2	Konklusjon	118
	Bibliografi	119
A	Forprosjekt	121
B	Papirprototype webapplikasjon	123
C	Papirprototype mobilapplikasjon	127
D	Årsrapport	131
E	Brukertest: Prosedyre	135
F	Brukertest: Notater	143

Figurer

1.1	Søye og to lam på beite (foto: Anders H. Rebner)	4
2.1	Registreringsskjerm i Beitesnap	13
2.2	De forskjellige oppsettene for opptelling med sveiping i "Sheepter"	16
4.1	Domenemodell	30
4.2	Klient-tjener-arkitektur	31
4.3	Firestore datastruktur: Alternativ 1	32
4.4	Firestore datastruktur: Alternativ 2	33
4.5	Firestore datastruktur: Alternativ 3	34
5.1	Egne ikoner	39
5.2	Egne kartmarkører med utvalg av ikoner	40
5.3	Webapp: Innlogging	41
5.4	Webapp: Opprette brukerkonto	41
5.5	Webapp: Tilbakemelding ved ugyldig input	42
5.6	Webapp: Min gård	43
5.7	Webapp: Mine beiteområder	44
5.8	Webapp: Legg til nytt beiteområde	45
5.9	Webapp: Legg til nytt beiteområde, markert	45
5.10	Webapp: Mine øremerker	46
5.11	Webapp: Endre øremerkefarge	46
5.12	Webapp: Mine slips	47
5.13	Webapp: Slipsendring	48
5.14	Webapp: Mitt oppsynspersonell	49
5.15	Webapp: Se oppsynstur	49
5.16	Webapp: Se oppsynstur (nedscrollet)	50
5.17	Webapp: Popups for å se enkeltregistreringer	51
5.18	Webapp: Generer årsrapport	52
5.19	Mobilapp: App-ikon	53
5.20	Mobilapp: Programflyt fra brukerens perspektiv	54
5.21	Mobilapp: Innlogging og brukerregistrering	55
5.22	Mobilapp: Start oppsynstur	56
5.23	Mobilapp: Kan ikke starte oppsynstur	57

5.24 Mobilapp: Start oppsynstur uten nettverksforbindelse	58
5.25 Mobilapp: Synkroniseringsstatus	59
5.26 Mobilapp: Hovedskjerm	59
5.27 Mobilapp: Summert turoversikt	60
5.28 Mobilapp: Registrering	61
5.29 Mobilapp: Registrering av sau	63
5.30 Mobilapp: Eksempler på validering av saueregistrering	63
5.31 Mobilapp: Registrer skadd sau	65
5.32 Mobilapp: Registrering av sauekadaver	65
5.33 Mobilapp: Registrering av rovdyr	66
5.34 Mobilapp: Registrering av notat	66
5.35 Mobilapp: Opplasting av oppsynstur	68
5.36 Mobilapp: Innstillinger	68
5.37 Eksempler på popup-vinduer for å bekrefte handling	70
5.38 Webapp: Spinning circle under generering av årsrapport	70
5.39 Mobilapp: Plassering av knapper på registreringskjermer	71
6.1 Eksempler på sprint retrospective	74
6.2 Utviklingstavle	75
7.1 Webapp: Widget-hierarki	82
7.2 Mobilapp: Widget-hierarki	82
7.3 Kartflis på zoomnivå 17 som viser Studentersamfundet i Trondheim	84
8.1 Planlagt rute for brukertest	96
8.2 Utklipp av saueregistrering før og etter endring	99
8.3 Utklipp av plussmeny-knapp før og etter endring	100

Tabeller

2.1	Priser for sporingssystemer	10
3.1	Use case: Opprette brukerkonto	21
3.2	Use case: Logge inn	21
3.3	Use case: Administrere eget oppsynspersonell	22
3.4	Use case: Se gjennomførte oppsynsturer	22
3.5	Use case: Laste ned årsrapport	22
3.6	Use case: Definere beiteområder	22
3.7	Use case: Definere øremerker	22
3.8	Use case: Definere slips	22
3.9	Use case: Opprette brukerkonto	23
3.10	Use case: Logge inn	23
3.11	Use case: Gjennomføre oppsynstur	23
3.12	Use case: Laste ned kart	23
3.13	Use case: Se kart	23
3.14	Use case: Se egen posisjon i kart	24
3.15	Use case: Se eget bevegelsesmønster på kart	24
3.16	Use case: Se posisjon på registrering på kart	24
3.17	Use case: Se siktlinje til registrering på kart	24
3.18	Use case: Registrere sau	25
3.19	Use case: Registrere skadd sau	25
3.20	Use case: Registrere død sau	26
3.21	Use case: Registrere rovdyr	26
7.1	Antall kodelinjer, beregnet i Powershell	77
7.2	Eksempel på egenskaper ved kartdata for et område på 10km ²	85
7.3	Justeringsfilter for tall	86
9.1	Eksempel på årlig inntekt generert av en søye	110

Kodelister

4.1	Felter i Users-dokumentet	34
4.2	Felter i Farms-dokumentet	35
4.3	Felter i Trips-dokumentet	36
4.4	Felter som er felles for alle Registrations-dokumenteter	36
4.5	Unike felter i Registrations-dokument for registrering av sau	37
4.6	Unike felter i Registrations-dokumentet for registrering av skadd sau	37
4.7	Unike felter i Registrations-dokumentet for registrering av kadaver	38
4.8	Unike felter i Registrations-dokumentet for registrering av rovdyr .	38
4.9	Unike felter i Registrations-dokumentet for registrering av notat . .	38
8.1	Kommando for å kjøre integrasjonstester i app og web	92

Kapittel 1

Introduksjon

I dette kapitlet gis en kort innføring i oppgavens tematikk og hvilket utbytte som er ønskelig av prosjektet. Først beskrives det overordnede fagområdet, som stegvis avgrenses ned til oppgavebeskrivelsen. Deretter presenteres prosjektets mål og gruppens kompetanse, før det til slutt gis en oversikt over rapportstrukturen.

1.1 Fagområde

”Husdyr er dyr som mennesket har temmet og tatt i sin tjeneste”¹. Husdyrene (også kalt tamdyr) var opprinnelig villdyr som gjennom selektiv utvelgelse (avl) har blitt tilpasset menneskelige behov. Eksempler på slike dyr er hund, katt, hest, storfe, gris, geit, sau og høns. Med ”menneskelige behov” menes eksempelvis at hesten ble temmet for å brukes til transport og annet arbeid, at storfe ble temmet for produksjon av kjøtt og melk, og at hunden ble temmet for å bli brukt til jakt, gjeting og vakthold. ”I dag er det vanlig å skille mellom produksjonsdyr som for eksempel storfe, og husdyr som holdes for andre formål, slik som kjæledyr, hobbydyr, sportsdyr eller tjenestehunder”¹.

1.2 Avgrensning

1.2.1 Dyrevelferd

Dyrevelferd er den første av oppgavens tre avgrensninger. Av etiske hensyn og samfunnspektiver som bærekraft og økonomi, har det blitt stadig viktigere å sørge for at dyrevelferden og dyrehelsen er god. Dyrehelserapporten fra 2020² forteller at norsk dyrehelse, i internasjonal sammenheng, er i verdenstoppen, med fravær av mange alvorlige smittsomme sykdommer. Dette bekreftes av Verdens

¹<https://snl.no/husdyr>, besøkt 16.04.2021

²<https://www.vetinst.no/rapporter-og-publikasjoner/rapporter/2021/dyrehelserapporten-2020>, besøkt 16.04.2021

dyrehelseorganisasjon (OIE) og Den europeiske myndighet for næringsmiddeltrygghet (EFSA) i rapporten. Videre påpeker rapporten at den gode dyrehelsen også bekreftes av det lave forbruket av antibiotika hos dyr, noe som er gunstig da sykdom virker negativt på dyrenes velferd, og på dyreeierens økonomi og om-dømme.

Lov om dyrevelferd §3 slår fast at ”Dyr har egenverdi uavhengig av den nytteverdien de måtte ha for mennesker. Dyr skal behandles godt og beskyttes mot fare for unødige påkjenninger og belastninger”³. Videre forteller §23 i samme lov at ”Dyreholder skal sikre at dyr holdes i miljø som gir god velferd ut fra artstypiske og individuelle behov, herunder gi mulighet for stimulerende aktiviteter, bevegelse, hvile og annen naturlig atferd”³. En av hovedårsakene til at vi har dyr på beite er altså at det er ansett som god dyrevelferd. Beitebruk har også andre fordeler knyttet til klima, biologisk mangfold, minskning av temperaturøkningen og redusert import av fôr⁴.

1.2.2 Sau

Oppgaven er videre avgrenset til å omhandle sau. Sauer er mellomstore drøvtyggere som på verdensbasis utgjorde over én milliard dyr i 2010⁶. ”Sauer hører til de eldste husdyrene [...]. De eldste funnene er fra Nord-Iran [...] og blir datert til ca. 6000 år fvt. Eldste funn i Norge er fra [...] Bergen, tidfestet til 1500-1400 fvt.”⁵. Sauer er flokkdyr som opptre i grupper, noe som forenkler gjeting, men som også gjør dem til utsatte byttedyr. Hunnen kalles sau, søye eller tikke, hannen kalles bukk eller vær, og avkommet kalles lam⁶. Disse faller alle innunder fellesnavnet småfe, som brukes om mindre husdyr i norsk landbruk.

Norge har sauebesetninger over hele landet, og har i tillegg den største saue-tettheten med 229 sauer per 100 hektar dyrket mark.⁶ ”Norsk kvit sau er målt til 73 prosent (2015) av den norske sauepopulasjonen”⁵. ”Av inntektene fra saueholdet kommer 3/4 fra kjøttproduksjonen, 1/4 fra produksjonen av ull”⁵. ”Rundt 2,1 millioner [norske] sauer og lam beiter i utmarka hvert år, og ca. 30 000 blir tatt av rovdyr (de som får erstatning)”⁵.

1.2.3 Oppfølging av sau på beite

Oppgaven avgrenses til slutt inn mot manuell oppfølging av sau på beite, med andre ord å gå oppsynsturer til fots i beiteterranget for å gjøre tilsyn av sauene. Før sauene slippes ut på beite festes bjeller på søyene for å tilvenne både dem og deres lam til lyden av denne. Det er også vanlig at ulla klippes. Transport og sauehold

³<https://lovdata.no/dokument/NL/lov/2009-06-19-97>, besøkt 16.04.21

⁴<https://www.landbruk.no/biookonomi/5-grunner-til-at-vi-trenger-beitedyr-ogsai-fremtiden/>, besøkt 16.04.21

⁵<https://snl.no/sau>, besøkt 7.12.2021

⁶<https://no.wikipedia.org/wiki/Tamsau>, besøkt 16.04.2021

skal samsvare med regelverk om dyrevelferd og dyrehelse, og det er Mattilsynet som skal tilse at regelverket følges. Det innebærer blant annet identifikasjon og sporing av sauene, noe som er med på å forebygge lidelser og utbrudd av sykdom.⁷

Figur 1.1 viser noen av identifikasjonsmetodene, nærmere bestemt bjelle i tillegg til øremerker og slips i en bestemt farge. Forskrift om merking, registrering og rapportering av småfe⁸ krever at sau skal merkes med et visuelt og elektronisk merke senest 30 dager etter fødsel. Øremerkene skal inneholde teksten "MT" (Mattilsynet), "NO" (nasjonalitetsidentifikasjon), et åttesifret produsentnummer eller en sjusifret identifikator, og et fire- eller femsifret individnummer. Fargen på øremerkene er valgfri, med unntak av hvit som er forbeholdt småfe som har blitt forlyttet til et nytt norsk sauehold, og lakserød som er forbeholdt importert småfe. Vi ser på figuren (1.1) at alle øremerkene er grønne, noe som betyr at de sannsynligvis tilhører samme gård. Videre er det vanlig praksis å bruke bjelleslips (også kalt slips) der fargen tilsier hvor mange lam sauen hadde da den ble sluppet ut på beite. En slik fargekodning lønner seg dersom en sauebesetning kan komme til å blande seg med en annen. Bruk av slips er likevel ikke påkrevd, og slipsbetydning kan være ulik mellom bestander. NSG (Norsk sau og geit) har kommet med en anbefaling der rødt slips betyr null lam, blått betyr ett lam, gul (eller ingen farge) betyr to lam, og grønt betyr tre lam. På figuren ser vi at sauen bærer et rødt slips, og det er fristende å anta at rødfargen i dette tilfellet betyr at de to lammene er hennes.

Dyrehelesrerapporten² forteller at det kan være trangt i innefôringsperioden, spesielt når søyene er drektige, og i lammingsperioden. "Høyt lammeantall er en påkjenning for søya både før og etter fødsel, gir økt sannsynlighet for sykdom hos både mor og avkom og kan føre til økt lammetap"². Derfor er avlastningen sauene får på beite viktig. Det finnes en forskrift om velferd for småfe som er ment å tilrettelegge for god helse og trivsel, der §24 sier at "Småfe skal holdes på egnet beite minst 16 uker i året[...]"⁹. Videre sier §19 at "Dyr som holdes på utmarksbeite, skal sees etter minst en gang pr. uke i områder uten særskilt risiko". Det er likevel ikke bare fordeler ved utmarksbeite. Tapet av sauer i beitesesongen er høyt; det antas at tapet av lam på sommerbeite i 2020 var 10,5 prosent.² "De viktigste tapsårsakene er flåttbårne sykdommer (sjodogg), alveld, rovdyr og fluelarver"². Dette gir grunnlag for å holde oppsyn med sauene når de er på beite, som videre omtales som manuell oppfølging av sau på beite.

1.3 Oppgavebeskrivelse

Oppsyn av sau på beite har frem til i dag foregått ved bruk av penn og papir. Sauebonden eller annet oppsynspersonell går tilsyn i terrenget blant sauene og noterer

⁷https://www.mattilsynet.no/dyr_og_dyrehold/produksjonsdyr/sau_og_geit/, besøkt 10.12.2022

⁸<https://lovdata.no/dokument/SF/forskrift/2005-11-30-1356>, besøkt 20.03.2022

⁹<https://lovdata.no/dokument/SF/forskrift/2005-02-18-160>, besøkt 20.03.2022



Figur 1.1: Søye og to lam på beite (foto: Anders H. Rebner)

hvor de selv har gått, hvor de ser sauer og hva de har sett (totalt antall, antall lam og antall av hver ull-, øremerke- og slipsfarge). Ved sesongslutt føres notatene fra alle oppsynsturene som er gått over i en rapport som sendes til myndighetene. Notering med penn og papir er tungvindt i vind og vær, og omgjøringen fra en mengde fysiske notater til digital rapport er tidkrevende. Dette er tid som bonden kunne ha brukt på andre arbeidsoppgaver.

Vi ønsker derfor med dette prosjektet å effektivisere det manuelle oppsynet ved å digitalisere disse arbeidsoppgavene. Det skal utvikles en første versjon av et databasert verktøy som kan tas med ut i felt og brukes til den manuelle oppsynsoppgaven. Det skal også lages en første versjon av et tilhørende system som kan laste inn alle oppsynsturene som har blitt gått og produsere den standardiserte rapporten. For verktøyet som tas med ut i felt er det ønskelig å se på mulighetene for å ta i bruk taleregistrering, som ble undersøkt i et forprosjekt (se vedlegg A). Forprosjektet kom i grove trekk frem til at denne type registrering er hensiktsmessig. Primærårsaken er at det er lett å miste overblikk over sauene dersom man stadig må flytte blikket inn og ut av kikkerten for å notere. Til slutt skal systemet testes og dets kvalitet vurderes.

1.4 Formål

Oppgavens formål er å utvikle en første versjon av et databasert verktøy som tar for seg oppsynsturer av sau i felt, og en tilhørende løsning som produserer en standardisert og digital rapport fra disse.

1.4.1 Effektmål

Prosjektets produkt skal kunne brukes til å vurdere om datateknologi har blitt så god at den egner seg bedre til saueoppsyn enn den tradisjonelle måten å gjøre oppsyn på. Et undermål er at systemet skal gi svar på om dagens taleregistreringsteknologi har kommet langt nok til å være hensiktsmessig å integrere i en slik spesialisert løsning.

1.4.2 Læringsmål

Prosjektmedlemmene skal tilegne seg:

- Grunnleggende forståelse vedrørende hold og oppsyn av sauer.
- Erfaring innenfor å danne seg en felles og presis forståelse av en reell oppgave fra et ukjent domene, samt å komme frem til en habil løsning på denne.
- Praktisk erfaring med å gjennomføre brukertesting av et selvutviklet system.
- Kunnskaper innenfor valgt programmeringsspråk, frontend-rammeverk og kart-bibliotek i tillegg til backend- og databaseteknologi.

1.5 Målgruppe

Målgruppen for systemet som utvikles er sauebønder og annet personell som går oppsyn. Videre kan rapporten være relevant for utviklere som er interessert i kart- og talegjenkjennings-teknologi.

1.6 Bakgrunn og kompetanse

Gruppen består av Nikolai Mork og Anders H. Rebner, som begge er masterstudenter i Informatikk med spesialisering i programvaresystemer ved NTNU i Trondheim. De har begge oppnådd bachelorgrad i ingeniørfag, data ved NTNU i Gjøvik, og har dermed vært gjennom ingeniørfaglige emner. Hovedvekten av studiene omfattet derimot emner som programmering i C++ og Python, algoritmer, operativsystemer, relasjonsdatabaser, brukergrensesnittdesign og systemutvikling. Gruppemedlemmene har i tillegg, gjennom jobberfaringer og valgfag, tilegnet seg kunnskaper innenfor utvikling av web- og mobilapplikasjoner og spill med GUI-biblioteker som JavaFX, React og React Native.

1.7 Rammer

- Prosjektets tidsramme er fra 10.01.2022 til 06.06.22.

1.8 Rapportstruktur

Introduksjon: Introduserer fagområdet prosjektet befinner seg i og legger frem beskrivelse av oppgaven.

State of the art: Gir et innblikk i hva som finnes av eksisterende, relevant teknologi og relaterte systemer.

Kravspesifikasjon: Beskriver kravene som stilles til systemet.

Teknisk design: Beskriver overordnet systemdesign med fokus på database.

Design av brukergrensesnitt: Presenterer systemet fra brukerens perspektiv.

Utviklingsprosess: Beskriver metodikken som ble fulgt under prosjektets utviklingsfase.

Implementasjon: Forklarer hvordan et utvalg funksjonalitet er implementert.

Testing: Beskriver hvordan testing av programvare og brukergrensesnitt er gjennomført, og presenterer resultatene fra brukertesting.

Diskusjon: Diskuterer ulike beslutninger som ble tatt og reflekterer over prosjektets fremgang, mangler og muligheter for videre arbeid.

Konklusjon: Fremlegger og evaluerer prosjektets resultat.

Kapittel 2

State of the art

I dette kapittelet gis en oversikt over situasjonen i fagfeltet ved å gå gjennom systemer som har direkte eller indirekte relevans til systemet som presenteres i denne oppgaven. Først presenteres systemer som digitaliserer dokumentasjon av dyreobservasjoner, deretter presenteres diverse sporingssystemer for dyr på beite, før systemer med tilsvarende funksjonalitet som systemet som presenteres i denne oppgaven beskrives. Til slutt vil kapittelet oppsummeres i en konklusjon.

2.1 Dokumentasjon av dyreobservasjoner

I denne seksjonen vil systemer med funksjonalitet for å dokumentere observasjoner av dyr bli beskrevet. Systemer i denne kategorien er først og fremst relevante fordi de omhandler dokumentasjon av ville dyr, men også fordi de kan ha gode løsninger på å registrere dyr i bevegelse, noe som kan være utfordrende.

2.1.1 Skandobs

Skandobs¹ er en skandinavisk tjeneste for rapportering av bjørn, jerv, gaupe og ulv. Tjenesten er et statlig samarbeid mellom Norge og Sverige der dataen brukes til forskning og bestandsoversikt. Når en registrering er av spesiell interesse for de respektive organer i Norge eller Sverige vil den bli ettergått av feltpersonell. Hvem som helst kan gjøre en registrering via mobilapplikasjon eller nettside, enten på stedet eller i etterkant. Registreringsskjemaet er på syv sider og omfatter hva som er observert (spor, faktisk dyr, kadaver, ekskrement osv.), art, bilde, posisjon, metadata om observasjonen og informasjon om observatøren.

¹<https://www.skandobs.se>, besøkt 05.05.2020

2.1.2 Artsobservasjoner

Arsobservasjoner² er et nettsted som finansieres av Klima- og miljødepartementet³. Hvem som helst kan rapportere et funn, og alle arter innenfor sopp-, plante- og dyreriket kan registreres. Registreringsskjemaet inneholder felter om arten og funnet. Skjemaet er ikke tilpasset mobilskjerm da feltene er små og man må manøvrere både vertikalt og horisontalt for å nå alle feltene. På nettsiden kan man se alle observasjonene som ligger i databasen, med unntak av arter som av ulike grunner er skjernet. Nettsiden brukes av privatpersoner, forskere, forvaltere og foreninger. I Sverige finnes Artportalen⁴ som en tilsvarende tjeneste som Artsobservasjoner, men for Sverige.

2.1.3 Viltappen

Viltappen⁵ er en mobilapplikasjon fra Svenska Jägerförbundet⁶. Applikasjonen har informasjon om dyreartene i svensk natur, hvordan man skal oppføre seg rundt dem og hvordan man kan kjenne igjen deres spor og lyd. Applikasjonen har også en funksjon for å registrere egne observasjoner av dyr i naturen. Registreringsskjermen inneholder et enkelt skjema med felter for navn, sted og tid på observasjonen, samt en lengre beskrivelse av denne, med mulighet for å legge inn lydopptak og bilder.

2.2 Posisjonsrelaterte systemer

Springssystemene som presenteres i denne seksjonen er relevant til oppgaven da de effektiviserer tilsyn ved å gi informasjon om dyrenes posisjon, slik at man ikke trenger å lete etter dem. I tillegg tilbyr systemene varslings ved diverse unormal oppførsel, som gjør at man kan bedre dyrevelferden med supplerende spesifikt tilsyn av enkeltdyr. Med Nofence, som beskrives i seksjon 2.2.1, kan tilsynet effektiviseres ytterligere ved å begrense området dyrene beiter på. Systemene har ulemper som pris og varierende grad og typer av nettverksdekning. Selv om systemene kan effektivisere tilsyn kan de ikke erstatte ukentlig oppsynstur da det er lovpålagt. Videre vil en rekke av denne typen systemer bli presentert.

2.2.1 Nofence

Nofence⁷ er et system bestående av elektroniske klaver⁸ og en mobilapplikasjon. Systemet baserer seg på at de voksne dyrene i bestanden bærer de elektroniske

²<https://www.artsobservasjoner.no/Home/About>, besøkt 05.05.2022

³<https://www.regjeringen.no/no/dep/kld/id668>, besøkt 05.05.2022

⁴<https://artportalen.se>, besøkt 06.05.2022

⁵<https://apps.apple.com/no/app/viltappen/id491035974>, besøkt 06.05.2020

⁶<https://jagareforbundet.se>, besøkt 06.05.2020

⁷<https://www.nofence.no>, besøkt 04.05.2022

⁸Halsbånd for husdyr

klavene og at bonden definerer et beiteområde de skal holde seg innenfor. Om et av dyrene beveger seg i utkanten av det definerte området vil klaven gi fra seg lyd som skal få dyret til å ikke gå videre. Om dyret likevel ikke snur vil det få støt. Bonden får oversikt over hvor klavene befinner seg og varsles dersom dyrenes aktivitet endrer seg. Den elektroniske klaven bruker GPS for posisjonering, men er avhengig av mobilnett for å kommunisere med mobilapplikasjonen. Batteriet er oppladbart og klaven har solcellepanel. Bevegelsessensoren brukes til å monitorere og varsle om dyrets aktivitet. Nofence oppgir ingen enhetspris på deres nettside, de gir istedenfor tilbud ved forespørsel.

2.2.2 NorTrace

NorTrace⁹ har laget produktet "Gjeteren" som er en elektronisk klave for småfe. Klaven bruker Narrowband Internet of Things (NB-IoT) som gjør det mulig å spore sauene ned til én meters nøyaktighet, gir lang batterilevetid og lav kostnad per enhet. Bonden kan gjennom et webgrensesnitt få opp posisjonen til hver enkelt sau og bli varslet hvis de ikke har forflyttet seg over et lengre tidsrom.

2.2.3 Telespor

Telespor¹⁰ fører produktet Radiobjella. I dette avsnittet er det 4. generasjons smartbjelle som er beskrevet, da denne er nyeste modell. Radiobjella bruker GNSS¹¹ for sporing av enhetene og NB-IoT for kommunikasjon. Bjella er vanntett og har Bluetooth. Videre kan den varsle per SMS dersom enheten har lavt batterinivå, ikke har dekning eller ikke har vært i bevegelse på en stund. Kostnaden per enhet er 989 kroner, og for abonnement i 5 måneder koster det 99 kroner. Det anbefales å skifte batteri hvert år, og ekstra batteri kan kjøpes med abonnement for 129 kroner.

2.2.4 Findmy

Findmy¹² sin E-bjelle Model 2 er en sporingsenhet som festes til klaven på sauene. E-bjella bruker satellitteknologi og er ikke avhengig av GSM¹³ eller NB-IoT. Findmy har tilhørende programvare for både mobiltelefon og pc. Systemet har alarmfunksjonalitet ved stress, lite bevegelse, ingen bevegelse eller bevegelse utenfor definerte områder. I tillegg har systemet en rekke karttyper, samt kart med innsikt i historiske posisjoner, tilvekst, beitetrykk og lammetap. Systemet kan også kobles til Sauekontrollen, som er "[...] den landsomfattende husdyrkontrollen og grunnlaget for avlsarbeidet på sau i Norge."¹⁴ Oppsettet av bjellene gjøres med

⁹<https://www.telia.no/magasinet/verdens-storste-iot-pilot>, besøkt 04.05.22

¹⁰<https://telespor.no>, besøkt 04.05.2022

¹¹Global Navigation Satellite Systems

¹²<https://www.findmy.no>, besøkt 04.05.22

¹³Global System for Mobile Communications

¹⁴<https://www.animalia.no/no/Dyr/husdyrkontrollene/sauekontrollen>, besøkt 24.05.2022

mobiltelefon via Bluetooth, som også kan brukes til å spore opp bjellene når man er i nærheten av dem. Pris per enhet er 1 749 kroner og årlig driftsavgift, som gir tilgang til programvare, koster 239 kroner per år.

2.2.5 Smartbjella

Smartbjella¹⁵ er en IoT-enhet for sporing som festes til klaven på sauene. Smartbjella bruker NB-IoT levert av Telenor, og GPS for nøyaktig posisjonering. Produktet har en tilhørende nettside og en mobilapplikasjon hvor man får oversikt over sine bjeller. Systemet har funksjonalitet for å se historiske posisjoner, alarmering om dyret ikke beveger seg over tid, nedlastbare kart og innstillinger for bjellene som sendeplan og hvilke områder det skal varsles om at dyret befinner seg i. Bjella er robust bygget og tåler kulde, vann og røff behandling. Avhengig av hvilke innstillinger man har på bjellen kan batteritiden variere fra 1,5 år til 17 år. Siste versjon av smartbjella, Smartbjella 3, koster 999 kroner per enhet. I tillegg koster det for lisens og datatrafikk. Lisenser kan kjøpes med varierende mengde funksjonalitet. For basislisens (minimum funksjonalitetspakke) i 5 måneder koster det 109 kroner per enhet og for profesjonell lisens (all funksjonalitet) i 12 måneder koster det 149 kroner per enhet.

2.2.6 Oversikt over priser

Hensikten med tabell 2.1 er ikke å gi grunnlag for direkte prissammenligning, men heller å gi en oversikt over hvilke utgifter som kreves for å føre et sporingssystem. Årsaken er at det varierer hva prisene inkluderer når det kommer til funksjonalitet og varighet på enhetene, som forklart i systemenes respektive beskrivelser. Merk at beløpene er kostnadene for å føre én enkelt enhet. Merk også at NorTrace og NoFence ikke er med i tabellen da de ikke har oppgitt priser på sine nettsider. ”Minstepris” er billigste alternativ og ”Makspris” er dyreste alternativ for innkjøp av enhet og abonoment på minimum 5 måneder.

Produsent	Enhetspris	Minstepris	Makspris
Telespor	989	1 088	1 118
Findmy	1 749	1 988	1 988
Smartbjella	989	1 098	1 138

Tabell 2.1: Priser for sporingssystemer

For å supplere prisoversikten i tabell 2.1 vil det her listes opp noen punkter som danner et bilde av pengeflyten i norsk sauedrift.

- Grunnprisen på et levende søyelam er 2 600 kroner.¹⁶
- Grunnprisen på ei levende søye er 2 500-3 000 kroner.¹⁶

¹⁵<https://smartbjella.no>, besøkt 06.05.2022

¹⁶<https://www.nsg.no/sau/livdyrpriser>, besøkt 19.05.2022

- Staten subsidierer 985 kroner per sau om man har 1-150 dyr, og 209 kroner per sau om man har 151 eller flere dyr.¹⁷
- Staten subsidierer 50 kroner per sau og lam som er på beite og 235 kroner per sau og lam på utmarksbeite.¹⁷
- Nortura gir maksimalt 66,94 kroner per kg lam (16,1-25 kg av kjøttfylde E+, fettklasse 1-3 og distrikssone 5).¹⁸
- Normalt ligger prisen for lam på rundt 50 kr per kg.¹⁸
- Nortura gir opp til 24,05 kroner per kg for en voksen sau.¹⁸
- I 2020 ble hver sau som ble tatt av rovvilt i snitt erstattet med 846 kroner.¹⁹
- Norilia gir opptil 58,5 kroner per kg ull.²⁰ En voksen sau kan ha opptil 2 kg ull²¹ og klippes 2 ganger i året²².
- I 2013 anslo en landbruksforsker at "[...] så lite som 5 prosent av sauegårdene kan gi fullt levebrød til én person[...]"²³
- Det er mulig å søke om støtte til innkjøp av elektronisk sporingsutstyr hos næringsfondet, statsforvalteren og regionale miljøprogram.²⁴

2.3 Tilsvarende systemer

Felles for systemene som beskrives i denne seksjonen er at de digitaliserer og forsøker å effektivisere oppsynsturer. Først gis en gjennomgang av den kommersielle mobilapplikasjonen Beitesnap. Videre beskrives tidligere masteroppgaver i kronologisk rekkefølge. Ved å gjøre relevante søk i NTNU Open²⁵ finner man rapporter på at denne masteroppgaven er gjennomført en rekke ganger de siste årene, men med varierende fokusområde. Av de åpent tilgjengelige rapportene ser man at oppgaven ble gjennomført i tre paralleller i 2018, to i 2019, én i 2020 og i fire i 2021. Beskrivelsene av disse gjennomføringene vil fokusere på løsningenes funksjonalitet og utforming, samt hvordan løsningene ble mottatt under brukertesting, slik det kan leses av rapportene. Hver beskrivelse er fattet i korthet og rettfærdiggjør ikke systemene som helhet. Kun hensiktsmessige feil og mangler bemerkes.

¹⁷https://www.landbruksdirektoratet.no/nb/jordbruk/ordninger-for-jordbruk/produksjonstilskudd-og-avlosertilskudd-i-jordbruket/produksjonstilskudd-og-avlosertilskudd--endelige-satser#tilskudd_for_dyr_p%C3%A5_utmarksbeite, besøkt 19.05.2022

¹⁸<https://medlem.nortura.no/prislister/avregningspriser-smafe-vilkar-smafe-article16785-11969.html>, besøkt 19.05.2022

¹⁹<https://www.statsforvalteren.no/innlandet/miljo-og-klima/rovvilt/erstatning-for-tap-til-rovvilt-2020>, besøkt 19.05.2022

²⁰<https://medlem.nortura.no/smafe/priser-vilkar-norilia-har-okt-prisen-pa-den-fineste-cl-ulla-article23456-11787.html>, besøkt 19.05.2022

²¹<https://avisahemnes.no/2017/11/fra-ull-til-garn-2>, besøkt 20.05.2022

²²https://www.bondelaget.no/getfile.php/13626901-1380711626/MMA/Bilder%20NB/Mat/Mat-%20og%20landbrukspolitikk/Skole%2C%20barn%2C%20ungdom/Barneskole/Sau_skoleforedrag.pdf, besøkt 20.05.2021

²³<https://www.nrk.no/innlandet/satser-pa-sau-1.11001539>, besøkt 19.05.2022

²⁴<https://www.findmy.no/nb/produkt>, besøkt 19.05.2022

²⁵<https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/227489>

2.3.1 Beitesnap

Beitesnap²⁶ er en applikasjon lansert i 2017 utviklet av lokalfirmaet Fant AS som består av fire Gudbrandsdøler. Applikasjonen har en rekke funksjonalitet vedrørende observasjon av dyr på beite. Beitesnap har funksjonalitet for å melde inn observasjoner som er tiltenkt alle som ferdes i beiteområder, uavhengig av relasjon til dyrene. Bonden definerer sitt beiteområde ved å markere det som et polygon på et kart i applikasjonen. Etter dette blir bonden varslet når en observasjon gjøres innenfor området. Om det er flere som har overlappende beiteområde vil de få opp varselet og kan kommunisere ved å kommentere på dette. En observasjon kan enten registreres på stedet med bilde og GPS-posisjon, eller i etterkant ved å velge observasjonens posisjon i kartet. Begge metodene fører videre til observasjonsskjemaet vist i figur 2.1. Her fyller man inn beitedyrets art og type observasjon (sett, skade eller død), og eventuelt individnummer, eget navn, eget telefonnummer og en melding for tilleggsinformasjon. Applikasjonen har to typer brukere, betalende og ikke-betalende. Som ikke-betalende bruker kan man registrere observasjoner. Som betalende bruker kan man i tillegg til å definere beiteområde; gå oppsynstur, generere rapport over utført tilsyn, knytte underbrukere til kontoen og laste ned kart. Under oppsynsturen vil GPS-sporet man har gått lagres. I september 2017 hadde applikasjonen omtrent 3 000 ikke-betalende brukere.²⁷

2.3.2 "Lambo", Dysthe og Kjerstad, 2018

Lambo er et system utviklet av Stian Dysthe og Andreas Kjerstad i 2018 som består av en nettside og en mobilapplikasjon. I mobilapplikasjonen kan man laste ned kartutsnitt ved å flytte en rektangulær ramme slik at den overlapper området. Man kan også endre dimensjoner på rammen. Når man går oppsynstur og skal registrere en observasjon må man gjennom et skjema som er felles for alle typer observasjoner. Når en observasjon skal stedfestes må kartet flyttes slik at et fast plassert sikte i midten av skjermen treffer ønsket posisjon. Når en registrering er fullført vises den som en markør på kartet. Ved å klikke på en markør for en registrering får man muligheten til å trykke på en knapp som åpner registreringen. Applikasjonen har ingen funksjonalitet for å registrere uten å se på skjermen. I applikasjonen kan man se tidligere oppsynsturer enkeltvis. På nettsiden kan man se én eller flere oppsynsturer samtidig og filtrere på hvilke typer registreringer man vil se på kartet. Rapporten som genereres baserer seg kun på en enkelt oppsynstur, men man kan velge mellom enkel og detaljert versjon. Av brukertestene kom det frem at deltakerne slet med å forstå at man må bruke siktet for å plassere observasjoner. Det var også én deltaker som slet med å flytte rammen ved nedlastning av kart. [1]

²⁶<https://www.beitesnap.no>, besøkt 03.05.2022

²⁷<https://www.bondevennen.no/aktuelt/nyttig-app-til-beitesesongen>,
04.05.2022

besøkt

ice+ 19:35 89 %

< Observasjon Send

Velg dyreslag:

Sau Geit Storfe Reinsdyr

Velg type observasjon:

Sett Skade Død

Evt. individnr:
123456

Evt. ditt navn:
Ola Nordmann

Evt. ditt telefonnummer:
12345678

Evt. melding (f.eks farge på øremerke, klave eller bjelle og annen relevant informasjon):

Figur 2.1: Registreringsskjerm i Beitesnap

2.3.3 "Pecora", Alm og Gård, 2018

Pecora er en mobilapplikasjon utviklet av Thomas Alm og Alexander Olav Gård i 2018. Innlogging i applikasjonen er integrert med Google og Facebook. Kartnedlastning gjøres ved å navigere kartet slik at området man ønsker å laste ned havner innenfor et fast plassert, dimensjonert og vertikalt orientert rektangel midt på skjermen. Størrelsen på kartutsnittet som lastes ned reguleres med zoom-nivå. Applikasjonen har mulighet for å opprette grupper som kan dele på ansvaret for oppsynsturer. En oppsynstur kan enten utføres for egen gård eller for en gruppe, eksempelvis et beitelag. Man kan forlate oppsynsturskjermen mens en oppsynstur pågår og fortsette den fra menyen. Ved å flytte turer mellom grupper kan man endre dens tilhørighet. I oversikten over turer som er gått har hvert element et sett firkanter med fargekoder som indikerer spesielle hendelser. Applikasjonen bruker et felles skjema for å registrere observasjoner. Skjemaet er delt inn i tre faner hvor "scenario" er tilpasset hva man har sett, eksempelvis sau, rovdyr osv. Alle registreringer gir mulighet til å ta bilder. Plassering av observasjoner i kartet gjøres med et sikte, som i Lambo (seksjon 2.3.2). En observasjon kan kobles til flere posisjoner, man kan med andre ord registrere at man har sett en observasjon fra flere punkter. Det er ingen funksjonalitet for å registrere uten å se på skjermen. Applikasjonen har en tidslinjemodus som lar brukeren bla gjennom tidligere observasjoner i kronologisk rekkefølge. Årsrapporter kan genereres for en gruppe og deles via Bluetooth eller Google Drive. Rapporten har én rad for hver oppsynstur,

men mangler dato. Applikasjonen har også en egen brukermanual. [2]

2.3.4 ”Pecora”, Schmidt-Hanssen, 2018

Pecora er et system bestående av en mobilapplikasjon og en nettside utviklet av Frida Meland Schmidt-Hanssen i 2018. Kart defineres og lastes ned i appen med ”rektangelmetoden”, altså på tilsvarende måte som beskrevet i seksjon 2.3.3. Ved start av oppsynstur må man fylle inn hvem som går turen. Når en observasjon skal plasseres brukes tilsvarende siktemetode som i Lambo (seksjon 2.3.2). Registreringsskjemaet på én side omfavner alle registreringstyper. I skjemaet huker man av det man har sett og fyller inn i feltene som deretter vises. Registreringsprosessen krever bekreftelse på en rekke handlinger. I mobilapplikasjonen kan man i tillegg se tidligere turer, og data synkroniseres med server ved å trykke på en knapp. Det finnes ingen funksjonalitet for å registrere uten å se på skjermen. Etter innlogging på nettsiden lander man på en oversikt over siste oppsynstur. Brukergrensesnittet til nettsiden er utformet slik at kartet opptar hele skjermen, og menyvalgene er noen få knapper i øvre høyre hjørne. Det er mulig å filtrere oppsynsturer på datoer, men det er bare de fem siste turene innenfor de oppgitte datoene som er tilgjengelig. Man kan velge flere oppsynsturer som skal vises i kartet samtidig. For å se metadata om en tur må man klikke på sporet som er gått. En årsrapport med overordnede tall kan genereres. I intervju og brukertest kom det frem at knappene på registreringskjermen var små og at det var vanskelig å laste ned riktig kart i mobilapplikasjonen. På nettsiden var det vanskelig å finne nedlastning av årsrapport da dette lå under innstillinger. [3]

2.3.5 Halvorsen, 2019

I 2019 utviklet Jørgen Aunet Halvorsen en webapplikasjon som skal gi oversikt og innsyn i data fra oppsynsturer. En mobilapplikasjon for oppsynsturer ble utviklet i forprosjektet, men dette er ikke tilgjengelig på NTNU Open. Webapplikasjonen har en mengde filtreringsmuligheter som valg mellom gråtonet eller farget kart, varmekart, størrelsesplot, saueobservasjonsfilter, rovdyrobservasjonsfilter, turrutefilter, beiteområdedata fra NIBIO²⁸, rovdyrdata fra NINA²⁹ og visning av flere turer samtidig. Man kan velge oppsynstur fra en tidslinje over kartet eller en meny på siden. For enkelhets skyld har applikasjonen en rekke forhåndsdefinerte kombinasjoner av filtre og funksjoner som visualiserer datane for en typisk situasjon på best mulig måte. I brukertestene kom det frem at enkelte deler av applikasjonen var for kompleks, eksempelvis mengden og variasjonen data som ble vist på kartet og at de forhåndsbestemte casene ikke var unike og spesifiserte nok. Utformingen av tidslinjen var heller ikke tydelig nok til at alle forstod dens hensikt. Som løsning foreslår Halvorsen blant annet å implementere muligheten

²⁸Norsk institutt for bioøkonomi

²⁹Norsk institutt for naturforskning

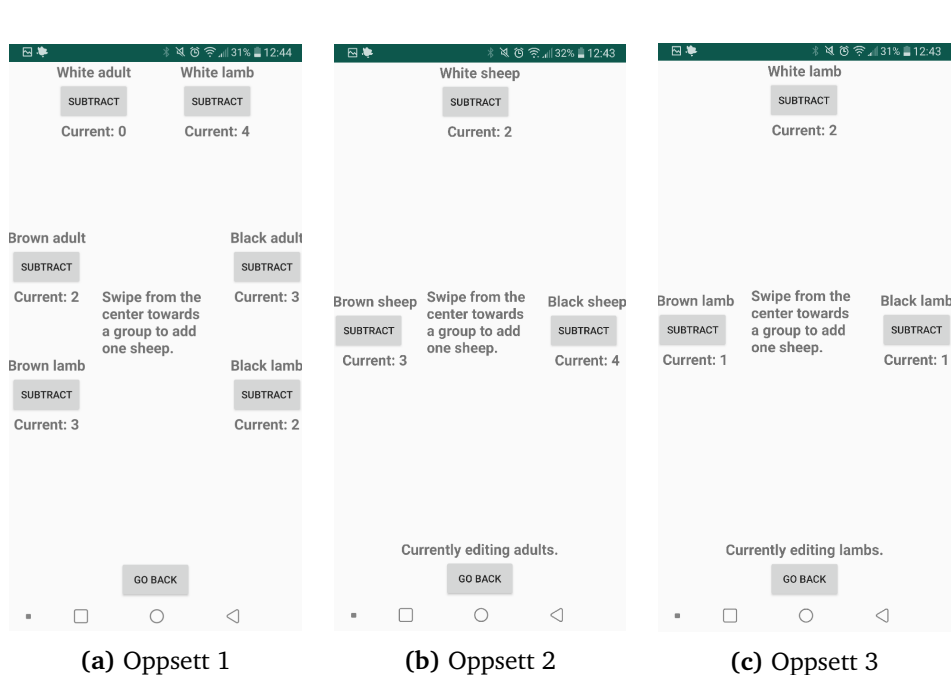
til å skifte mellom enkel visning og ekspertmodus, i tillegg til å personalisere applikasjonen ved å lagre egne kombinasjoner av filtre og oppsett. [4]

2.3.6 Svendsen, 2019

I 2019 utviklet Lærke Svendsen en mobilapplikasjon med fokus på effektivisering av brukergrensesnitt. Applikasjonen har funksjonalitet for å gå oppsynstur og registrere sau, skade, død og annet. I designutkastet foreslås det å trykke på kartet for å registrere hvor en observasjon er gjort, men dette velges bort da det argumenteres for at det ikke er nøyaktig nok og at det krever mange trykk. Løsningen som velges er å bruke et sikte, på tilsvarende måte som i Lambo (se seksjon 2.3.2). Videre i designforslaget foreslås det at knappen for registrering av sau skal være større enn de andre da det vil være flest av denne typen registrering. I tillegg foreslås det at man automatisk skal bli sendt til nær/avstands-registrering basert på avstanden mellom egen posisjon og observasjonspunktet. Ingen av forslagene blir gjennomført. For å muliggjøre registrering uten å se på skjermen kan man sveipe på skjermen for å inkrementere antall sauer. Det kommer frem av brukertestene at det burde bli gitt en form for tilbakemelding med lyd for at brukeren skal forstå at sveipet ble registrert. [5]

2.3.7 "Sheepter", Flytøren, 2020

Sheepter er en mobilapplikasjon utviklet av Petter Flytøren i 2020. Applikasjonens grensesnitt for administrering av kart er et kart i seg selv. Ved å trykke på "save map" kan kartområdet som vises innenfor skjermen lastes ned. Nedlastede kartområder er markert med et grått rektangel på kartet og kan slettes med et langt press og velge "slett". Under en oppsynstur kan man fortløpende velge mellom forskjellige visninger. På oversiktsvisning ser man metadata om turen, som tid og distanse. På kartvisningen vises sporet man har gått samt markeringer for observasjonene som er gjort. På observasjonslistervisning ser man en oversikt over observasjonene som er gjort, og man kan klikke på dem for å få en detaljert visning av dem. Applikasjonen tillater å registrere at en observasjon har blitt sett fra flere punkter. Applikasjonen har flere metoder å registrere sau på, som alle baserer seg på sveiping. Grunnmuren i sveipemetoden er at det i utkantene av skjermen er skrevet hva man inkrementerer ved å sveipe fra midten av skjermen og ut til teksten. I det første oppsettet (figur 2.2a) er brun, svart og hvit ullfarge plassert henholdsvis til venstre, høyre og på toppen av skjermen. På hver av utkantene er voksen og lam plassert med avstand fra hverandre slik at det er en hovedretning per farge og en mer nøyaktig retning for å velge mellom voksen og lam. På andre oppsett (figur 2.2b) er fargene plassert til venstre, høyre og øverst på skjermen, og man sveiper først for voksne før man går videre til en ny skjerm og sveiper for lam. På det tredje oppsettet (figur 2.2c) teller man én farge om gangen og sveiper til høyre for å legge til en sau og til venstre for å trekke fra en sau. For å endre farge sveiper man opp eller ned, og lam telles én gang uavhengig av farge. For å forenkle bruk uten å se på skjermen kan man dobbeltrykke for at applikasjonen



Figur 2.2: De forskjellige oppsettene for opptelling med sveiping i "Sheepster"

mundtlig skal fortelle hvilken farge som telles. Det kommer frem i diskusjonen at antall bør inkluderes i talen, slik at man i tillegg til ullfargen også får vite antallet som er telt. Resultatene fra brukertest viser at desto færre muligheter det var på skjermen desto færre ganger var det behov for å se på den. I intervjuene sa den ene brukeren at den likte oppsett to best mens den andre likte oppsett tre best, selv om det bør effektiviseres. Begge deltakerne slet med å plassere registreringer på riktig sted da disse ble plassert i midten av skjermen uten at det var noen indikator på dette. [6]

2.3.8 "Sauron", Abtahi og Gjeldseth-Borgen, 2021

Sauron er en mobilapplikasjon utviklet av Kimia Dadar Abtahi og Trym Vegard Gjeldseth-Borgen i 2021. I applikasjonen kan kart lastes ned med "rektangelmetoden" som i Pecora (seksjon 2.3.3). Når en oppsynstur startes må et skjema med metadata fylles ut. Før GPS-posisjonering i bakgrunnen ble implementert hadde applikasjonen en strømsparingsmodus der applikasjonen kjørte i forgrunnen med helt sort skjerm med unntak av instruksjoner på hvordan man forlater modusen. Det hevdtes at dette er svært strømsparende, spesielt på OLED-skjermer. Valg av type registrering (sau, skade, død eller rovdyr) gjøres gjennom en pluss-knapp nede til høyre i skjermen. Plassering av observasjonen skjer med et sikte som i Lambo (seksjon 2.3.2). På første skjerm i en saueregistrering må man velge blant forhåndsregistrerte øremerker og/eller legge til nye øremerker. Selve tellingen av sau er utviklet i forprosjektet til oppgaven og er beskrevet i detalj der, men denne forprosjektrapporten er ikke publisert på NTNU Open. Det man likevel kan

forstå om registreringen ut ifra masteroppgaven er at den baserer seg på sveiping der man sveiper til høyre for å legge til én og til venstre for å trekke fra én. Kategorien man teller på endres ved å sveipe opp og ned. Fofatterne beskriver metoden som en kombinasjon av ulike gester og virtuelle knapper, tekst til tale og haptisk tilbakemelding som muliggjør blind registrering av sau. Applikasjonen tillater gjennomgang av tidligere gjennomførte oppsynsturer. Fra brukertestene kommer det frem at ikke alle forstod hvorfor en observasjon ble registrert i midten av skjermen, at knapper og tekst bør være større og at det var små problemer med å forstå sveipemetoden med en gang. [7]

2.3.9 Carlsen, 2021

Jonas Ege Carlsen utviklet i 2021 en mobilapplikasjon med tilhørende server for digitalisering av oppsynsturer. På hjemskjermen til mobilapplikasjonen har man menyvalgene "Mine turer", "Områder", "Lag", "Invitasjoner" og "Gårder". Om en oppsynstur ikke er fullført vil det øverst i menyen gis valg for å fortsette eller forkaste den. Om en tur er avsluttet, men ikke lastet opp vil det være en knapp for å laste den opp øverst på hjemskjermen. Under gårder-menyen kan man søke opp gårder og laste dem ned til enheten. Under invitasjoner kan man bekrefte eller avkrefte invitasjoner til beitelag. Under lag-menyen kan man opprette eller gå inn på lag. Som administrator for et lag kan man sende invitasjoner, overføre administratorrettigheter eller slette laget. Som vanlig lagmedlem kan man se beskrivelse av laget og se turene som er gått for laget. Under menyen "Mine turer" vises en liste over turene man selv har gått samt når de ble gått og status på opplasting. Ved å klikke på en av turene får man en detaljert oversikt over alle registreringene som er gjort på turen. Denne siden har to faner, en for sau og en for annet. Under sauefanen vises alle observasjoner i en liste, der hver oppføring inneholder antall observerte ull- og slipsfarger, samt gårdene og deres øremerker. Under annet-fanen er det en tilsvarende liste, men med informasjon om annet-registreringer. Under områder-menyen kan man se og slette kart som er nedlastet, eller starte en oppsynstur på et område. Det er også fra denne menyen man laster ned nye kart med "rektangelmetoden", som beskrevet i seksjon 2.3.3. For å plassere en registrering i kartet brukes et trådkors, men i motsetning til tidligere oppgaver er ikke trådkorset fast posisjonert, man drar det til posisjonen man ønsker. En registrering er enten av typen sau eller "annet". Registreringsskjema for sau er bolkbasert. Øverste bolk er et enkelt felt for totalt antall. De neste bolkene er utvidbare. For slipsfarger kan man manuelt skrive inn eller bruke sveipe-metoden med oppløsning (beskrives i seksjon 2.3.10). Under gårder-menyen velger man forhåndsdefinerte gårder eller definerer en på stedet. Død sau, sauekadaver og rovdyr er underkategorier av "annet" og må velges på denne siden. Utover dette kan tekstlig beskrivelse og bilde legges til. I brukertestene kommer det frem at flytting av trådkorset var upraktisk fordi fingerene dekket til korsets senter, og funksjonaliteten ble deretter endret til at trådkorset er litt over fingeren. [8]

2.3.10 Hyll og Torgersen, 2021

Magnus Conrad Hyll og Truls Matias Torgersen utviklet i 2021 en mobilapplikasjon for digitalisering av oppsynsturer. For å starte en oppsynstur må man trykke på en stor sauetegning på startskjermen. Man blir etter dette tatt direkte til kartnedlastningsskjermen som bruker ”rektangelmetoden” beskrevet i seksjon 2.3.3. Man må velge mellom å laste ned et kart eller gå videre uten å laste ned et kart, uansett må valget bekreftes. Under en oppsynstur har man mulighet til å vise en tidligere oppsynstur i kartet samtidig. Denne vises da i grått mens pågående tur vises i svart. En registrering stedfestes i kartet med en fast plassert sauemarkør som i Lambo (seksjon 2.3.2). En registrering er enten av typen sau, skadet sau, død sau eller rovdyr. Ved registrering av sau kan man velge mellom fanene ”Ser slips” og ”Ser ikke slips”. Ved registrering av sau har applikasjonen en sveipemodus der man sveiper opp for å legge til én, sveiper ned for å trekke fra én og sveiper til høyre eller venstre for å endre hva som telles (som i figur 2.2c, men med motsatte akser). Hva som registreres leses opp fortløpende. Skjema for rovdyr, skadd sau og død sau er enkle og man kan ta bilde i de to sistnevnte. Man kan redigere eller slette en observasjon etter at den er lagt til. Ved avslutning av en oppsynstur vises en oppsummering av det som har blitt registrert på turen. Applikasjonen kan i tillegg vise tidligere turer, men da bare én om gangen. Under oppsynsturer kan man fortløpende velge mellom offline og online kart. Fra brukertesten kommer det frem at det var uklart hvordan man lastet ned kart og hvorfor man kom til denne skjermen når man startet en oppsynstur. Ikke alle forstod hvordan man skulle plassere en registrering. Utover dette var det flere funksjonaliteter som ikke var tilstrekkelig kommunisert gjennom brukergrensesnittet, som blant annet at man kan sveipe på skjermen for sveipetelling. [9]

2.3.11 Styve, 2021

Svein Olav Styve utviklet i 2021 en mobilapplikasjon for dataregistrering av beitede sau. Kartnedlasting gjøres med ”rektangelmetoden”, som beskrevet i seksjon 2.3.3, men kartutsnittet som lastes ned er det som er innenfor skjermen og ikke det som er innenfor et rektangel. For nedlastede kart kan man se en oversikt over hva som er registrert på kartet. Fra hovedskjermen kan man starte en ny oppsynstur eller fortsette en pågående oppsynstur. En registrering plasseres ved å holde inne på skjermen og man kan deretter velge mellom typene: flokk, død, skadet, rovdyr eller miljø. Oversikt over hva som er observert på en tur kan ses i listeform. Det er mulig å se tidligere gjennomførte turer og å eksportere data fra en tur. Ved registrering av sau kan man enten bruke pluss- og minusknapper for å øke eller minke antallet i en kategori med én, skrive inn antall med tastatur eller bruke sveipemotode som i Hyll og Torgersens applikasjon (se seksjon 2.3.10). Under menyvalget ”Rapport” kan man eksportere data for ett eller alle år til database-filer, JSON eller tekstrapport, i tillegg til å kun eksportere observasjonsbilder. Videre kan disse deles via Gmail eller Google Drive. Brukertester viste at førstegangsbrukere trenger tid til å forstå at en observasjon stedfestes med et langt press. Det tok også

litt tid å forstå konseptet med sveiperegistrering. Når brukeren etter ett forsøk var inneforstått med hvordan disse metodene fungerte var resultatene gode. [10]

2.4 Konklusjon

I dette kapittelet har vi gjennomgått systemer som har likheter med systemet som har blitt utviklet i dette prosjektet. Ved å utforske eksisterende arbeid har vi økt vår kunnskap på området og kan utnytte dette til videre arbeid. Vi vet nå at det pågår omfattende arbeid rundt dokumentasjon av arter og da spesielt rovdyr. Vi vet også at det finnes et bredt utvalg av systemer med tilhørende programvare for overvåking av dyr på beite. Disse koster mellom ett og to tusen per enhet i oppstartskostnader og hundre til tre hundre kroner per enhet i årlige driftskostnader. En diskusjon om sauebønder som marked kan leses i seksjon 9.4. I tillegg vet vi hva som finnes av gode og dårlige løsninger i systemer som digitaliserer oppsynsturer, noe vi tar med oss videre i utviklingen av dette systemet. Vi ser at mange av systemene har like løsninger. Eksempelvis har de fleste applikasjonene et sikte i kartet for å stedfeste observasjoner, og de fleste har et fast plassert rektangel som markerer kartutsnittet som skal lastes ned. Det er også gjort en del iterasjoner på bruk av sveiping for å telle uten å måtte se på mobilskjermen. En diskusjon av våre valg kan leses i seksjon 9.5.1. I tillegg til det helhetlige inntrykket vi har fått ser vi på følgende punkter som gode idéer:

- Langt trykk på kartet for å velge plasseringen til en observasjon
- Oppsummering av oppsynsturen ved avslutning av turen
- Mulighet for å sende inn observasjoner og bli varslet om observasjoner på eget beiteområde (som i Beitesnap, seksjon 2.3.1)
- Bruk av tekst-til-tale-teknologi for å kommunisere til brukeren
- Bruk av bilder for dokumentasjon
- Mulighet til å se tidligere oppsynsturer i mobilapplikasjon
- Se hva som er registrert på et punkt ved å trykke på en markør
- Se flere oppsynsturer i kartet samtidig i webapplikasjon
- Bruk av grupper/beitelag for å fordele oppsynsturer
- Koble en observasjon til flere punkter
- Brukermanual i mobilapplikasjon
- Mulighet for å vise relevante eksterne data
- Bruk av diverse fremstillingsmetoder i kart for å gi innblikk i saueflokk
- Enkel og rask tilgang til saueregistrering fremfor andre registreringstyper
- Dynamisk registreringsskjema basert på avstand til observasjon
- Bruk av sveiping for å registrere antall uten å se på skjermen
- Tanken om å spare strøm
- Mulighet til å bruke andre funksjonaliteter i applikasjonen samtidig som det går oppsynstur
- Fortløpende endring av offline og online kart
- Endre og slette observasjoner som er gjort under pågående oppsynstur

Videre ønsker vi å tilføre feltet ny kunnskap i form av å prøve andre løsninger enn det som har vært vanligst tidligere. Dette innebærer blant annet bruk av talegjenkjenning for registrering av sau, som er helt nytt i fagområdet. Samtidig ønsker vi å lage et komplett system, som beskrives videre i rapporten.

Kapittel 3

Kravspesifikasjon

Dette kapittelet presenterer kravene som stilles til systemet. Kravene har blitt utarbeidet underveis i samarbeid med veileder, som i denne settingen hadde rollen som kunde. Kravspesifikasjonen er delt inn i funksjonelle og ikke-funksjonelle krav.

3.1 Funksjonelle krav

De funksjonelle kravene til systemet presenteres som use cases, med enten sauebonde eller oppsynspersonell som aktør. Her er det et viktig poeng at sauebonden kan tre inn i rollen som oppsynspersonell.

3.1.1 Use cases for webapplikasjon

Use case:	Opprette brukerkonto
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne opprette en brukerkonto.

Tabell 3.1: Use case: Opprette brukerkonto

Use case:	Logge inn
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne logge seg inn på egen brukerkonto.

Tabell 3.2: Use case: Logge inn

Use case:	Administrere eget oppsynspersonell
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne administrere eget oppsynspersonell.

Tabell 3.3: Use case: Administrere eget oppsynspersonell

Use case:	Se gjennomførte oppsynsturer
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne se alle gjennomførte oppsynsturer og registreringer som er gjort på vegne av egen gård.

Tabell 3.4: Use case: Se gjennomførte oppsynsturer

Use case:	Laste ned årsrapport
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne laste ned en årsrapport med oversikt over alle oppsynsturene fra et valgt år.

Tabell 3.5: Use case: Laste ned årsrapport

Use case:	Definere beiteområder
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne definere egne beiteområder.

Tabell 3.6: Use case: Definere beiteområder

Use case:	Definere øremerker
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne definere øremerkefarger og om fargene tilhører egen eller andres gård.

Tabell 3.7: Use case: Definere øremerker

Use case:	Definere slips
Aktør:	Sauebonde
Beskrivelse:	Sauebonde skal kunne definere slipsfargene som brukes på egne søyer, og hvor mange lam de representerer.

Tabell 3.8: Use case: Definere slips

3.1.2 Use cases for mobilapplikasjon

Use case:	Opprette brukerkonto
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne opprette en brukerkonto.

Tabell 3.9: Use case: Opprette brukerkonto

Use case:	Logge inn
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne logge seg inn på egen brukerkonto.

Tabell 3.10: Use case: Logge inn

Use case:	Gjennomføre oppsynstur
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne gjennomføre oppsynstur både med og uten nettverksforbindelse.

Tabell 3.11: Use case: Gjennomføre oppsynstur

Use case:	Laste ned kart
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne laste ned kart for et avgrenset område definert av sauebonde.

Tabell 3.12: Use case: Laste ned kart

Use case:	Se kart
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne se, manøvrere og zoome i nedlastede kart.

Tabell 3.13: Use case: Se kart

Use case:	Se egen posisjon i kart
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne se markering for egen posisjon på kartet.

Tabell 3.14: Use case: Se egen posisjon i kart

Use case:	Se eget bevegelsesmønster på kart
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal på kartet kunne se hvor de har gått under pågående oppsynstur.

Tabell 3.15: Use case: Se eget bevegelsesmønster på kart

Use case:	Se posisjon på registrering på kart
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal på kartet kunne se hvor det er gjort registreringer.

Tabell 3.16: Use case: Se posisjon på registrering på kart

Use case:	Se sikotlinje til registrering på kart
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal på kartet kunne se sikotlinje fra egen posisjon, da registreringen ble gjort, til registreringsposisjonen.

Tabell 3.17: Use case: Se sikotlinje til registrering på kart

Use case:	Registrere sauer
Aktør:	Oppsynspersonell
Beskrivelse:	<p>Oppsynspersonell skal kunne registrere én eller flere sauer. Registreringen skal være mulig å gjennomføre uten å se på mobilskjermen. Hvilke data som skal registreres avhenger av avstanden mellom oppsynspersonell og sau(ene). Ved avstand over 50 meter (avstandsregistrering) skal følgende registreres:</p> <ul style="list-style-type: none"> • Sauen(e)s posisjon • Antall: <ul style="list-style-type: none"> ◦ Sauer og lam totalt ◦ Lam ◦ Av hver ullfarge (hvit/brun/svart/svart hode) <p>Ved avstand under 50 meter (nærregistrering) skal det i tillegg registreres:</p> <ul style="list-style-type: none"> • Antall: <ul style="list-style-type: none"> ◦ Av hver øremerkefarge ◦ Av hver slipsfarge

Tabell 3.18: Use case: Registrere sau

Use case:	Registrere skadd sau
Aktør:	Oppsynspersonell
Beskrivelse:	<p>Oppsynspersonell skal kunne registrere følgende tilknyttet skadd sau:</p> <ul style="list-style-type: none"> • Alvorlighetsgrad (moderat/alvorlig) • Type skade (beinskade/blodutredning/hodeskade/annen) • Øremerke-ID • Slipsfarge (valgfritt) • Sauens posisjon

Tabell 3.19: Use case: Registrere skadd sau

Use case:	Registrere død sau
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne registrere følgende tilknyttet sauekadaver: <ul style="list-style-type: none"> • Øremerke-ID • Slipsfarge • Fotografi • Kadaverets posisjon

Tabell 3.20: Use case: Registrere død sau

Use case:	Registrere rovdyr
Aktør:	Oppsynspersonell
Beskrivelse:	Oppsynspersonell skal kunne registrere følgende tilknyttet rovdyr: <ul style="list-style-type: none"> • Art (bjørn/ulv/jerv/gaupe/kongeørn/havørn) • Rovdyrets posisjon

Tabell 3.21: Use case: Registrere rovdyr

3.2 Ikke-funksjonelle krav

Denne seksjonen beskriver de ikke-funksjonelle kravene til systemet, altså krav som ikke direkte sikter til programvarefunksjonalitet.

3.2.1 Generelle krav

- Språket i brukergrensesnittet skal være norsk bokmål. Det kan gjøres unntak for taleregistrering, denne kan være engelsk om nødvendig.
- Kart som brukes i applikasjonene skal være topologiske kart levert av Norgeskart i minimum detaljnivå 1:25 000.
- Årsrapporten skal inneholde informasjon fra hver enkelt oppsynstur som ble gått innenfor året som velges, og være i en tilstand som kan leveres til myndighetene.

3.2.2 Operasjonelle krav

Det operasjonelle kravet som stilles til systemet er at det skal være mulig å gjennomføre oppsynsturer i områder uten nettverksforbindelse, så lenge kartområdet det skal utføres oppsyn i har blitt lastet ned på forhånd. Dette innebærer at også taleregistrering skal kunne brukes uten nettverksforbindelse, så lenge enheten støtter dette, er konfigurert riktig og språkpakke er nedlastet på forhånd.

3.2.3 Krav til leveranse

Følgende skal leveres:

- Systemets kodebase
- Instruksjoner for bruk av systemet
- En rapport som detaljert begrunner, beskriver, vurderer og diskuterer prosjektet i sin helhet

3.3 Mål for brukervennlighet

Systemet har som mål å:

- hjelpe brukeren å unngå å gjøre feil
- gi brukeren status når det jobber
- gi tilbakemeldinger som brukeren forstår
- være effektivt å bruke ute i felt

Kapittel 4

Teknisk design

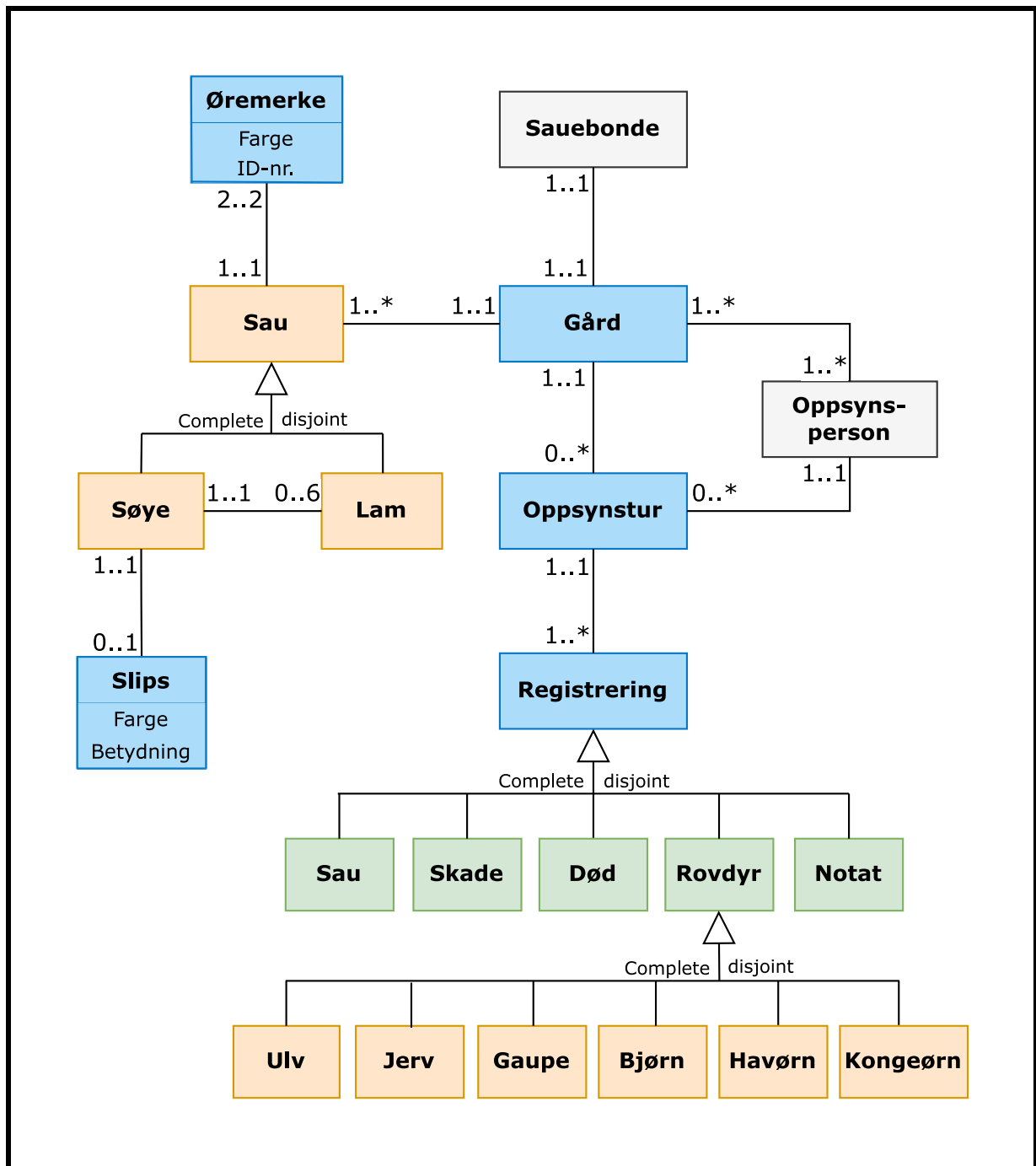
Dette kapitlet beskriver overordnet teknisk utforming av systemet og inneholder en domenemodell, en kort beskrivelse av arkitekturen og til slutt en utdypende beskrivelse av databasedesignet.

4.1 Domenemodell

For å gi en bedre og noe mer teknisk forståelse av domenet har vi valgt å modellere dette i et UML 2.5 klassediagram¹, som vist i figur 4.1. Modellen er ikke et typisk klassediagram med informasjon som klassenavn og variabeltyper, men fungerer heller som en konseptuell modell av de relevante delene av virkeligheten. Modellen vil brukes som et utgangspunkt for systemet som skal utvikles. Den har en fargeinndeling der blå representerer objekter/ting, grå representerer menneskeroller og grønn representerer registreringstyper.

For å holde modellen oversiktlig har vi gjort noen forenklinger ved å ekskludere enkelte assosiasjoner. Dette gjelder særlig for registreringstypene "Sau", "Skade" og "Død", som vil være assosiert med søyer, lam, slips og øremerker i henhold til kravspesifikasjonen (se tabell 3.18-3.20). Det er verdt å nevne at en sauebonde i praksis kan ha mer enn én gård, og at ei søye kan ha flere enn 6 lam, men vi anser dette som uvanlig. Det kan også være tilfelle at sauebonden trer inn i rollen som oppsynsperson og at gården dermed ikke har noe eksternt oppsynspersonell, modellen tar ikke hensyn til dette.

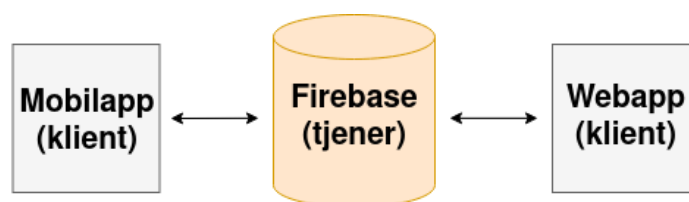
¹<https://www.uml-diagrams.org/class-diagrams-overview.html>, besøkt 25.03.2022



Figur 4.1: Domenemodell

4.2 Overordnet arkitektur

Applikasjonen består i hovedsak av tre komponenter; en mobilapplikasjon, en webapplikasjon og en skytjeneste. Den overordnede arkitekturen er klient-tjener hvor mobilapplikasjonen og webapplikasjonen er klienter og Firebase [11] er tjeneren, som vist i figur 4.2. Klientene er tykke og tjeneren er slank. Tjeneren holder databasen og utfører logikk for autentisering. Resterende logikk håndteres av klientene.

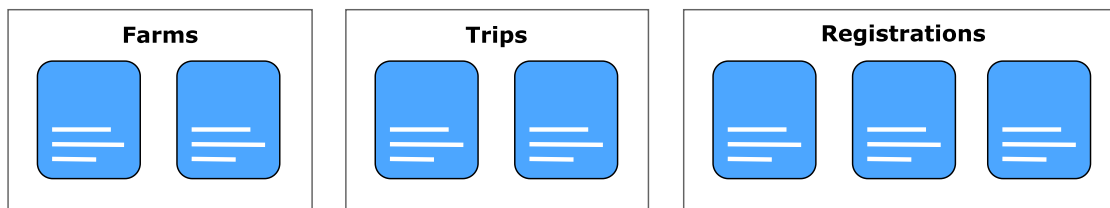


Figur 4.2: Klient-tjener-arkitektur

4.3 Database

Systemet bruker Firebase [11] sin skalerbare NoSQL dokument-database kalt Cloud Firestore². For å strukturere data har man i all hovedsak tre alternativer; documents (dokumenter), collections (samlinger) og dokumenter med subcollections (undersamlinger) i seg. Det er fordeler og ulemper ved hver av de tre datastrukturene, og hva som er den beste løsningen er svært applikasjonsavhengig. Beslutningen for datastruktur er gjerne en balansegang mellom antall database-reads (lesinger), mengde data per lesing, forventet programflyt og tilgangskontroll. Antall lesinger er sentralt fordi man betaler per dokument-lesing, uavhengig av dokumentets størrelse. Man kan derfor tro det er lurt å legge store mengder data i enkeltdokumenter for å spare penger, men det har i hovedsak tre potensielle ulemper. Den første er at dokumenter er begrenset i størrelse (1MB) og i antall felter (20 000), så skalerbarheten svekkes dersom man har behov for å stadig legge til data i samme dokument. Andre ulempe er at det ikke er mulig å laste ned deler av et dokument, noe som tvinger klienten til å laste ned (mye) mer data enn den har bruk for med mindre hele dokumentet skal brukes. Tredje og siste ulempe er at klienten blir nødt til å utføre filtrering eller sortering av data dersom brukergrensesnittet tilbyr dette, noe som er ansett som dårlig praksis med store datasett. Derfor er det i mange tilfeller best å fordele data på flere dokumenter, og muligens bruke samlinger og/eller undersamlinger. Vi vil i de neste avsnittene ta for oss datastrukturene vi vurderte, diskutere fordeler og ulemper ved hver av disse, og begrunne vår beslutning.

²<https://firebase.google.com/docs/firestore>, besøkt 24.03.2022



Figur 4.3: Firestore datastruktur: Alternativ 1

4.3.1 Hovedstruktur

Alternativ 1

Figur 4.3 viser "Farms", "Trips" og "Registrations" som tre topp-nivå samlinger. Hver gård, oppsynstur og registrering har altså sitt eget dokument innenfor disse. Med denne datastrukturen kan man tro at søking gjennom "Registrations"-samlingen blir tregt etterhvert som antallet dokumenter øker, men det trenger man ikke. Tiden det tar å gjennomføre en query (spørring) er proporsjonal med antall resultater man får, uavhengig av hvor mange dokumenter det søkes gjennom². Denne datastrukturen kunne dermed vært en habil løsning dersom det var relevant å utføre spørringer som å hente ut de ti lengste oppsynsturene, eller de ti registreringene med høyest antall sauer på tvers av gårder. Da det ikke er relevant å hente ut oppsynsturer eller registreringer på tvers av gårder, valgte vi å gå bort fra denne strukturen.

Alternativ 2

Vi innså at det med datastrukturen i Figur 4.3 ville ha kostet mange lesinger å hente ut alle registreringer fra en oppsynstur (proporsjonalt med antall registreringer), og ønsket dermed å se på muligheter for å redusere disse. Figur 4.4 viser den andre datastrukturen som ble vurdert, hvor alle registreringer er flyttet fra individuelle dokumenter til å være ett felt innunder et oppsynstur-dokument. Den klare fordelene med dette er at man vil bruke én enkelt lesing på å laste ned en oppsynstur og dens 30 registreringer, sammenlignet med 31 lesinger med datastrukturen i alternativ 1 (vist i figur 4.3). Samtidig har denne strukturen en ulempe ved at klienten potensielt laster ned større datamengder enn den har bruk for, fordi man som nevnt må lese hele dokumenter. Det vil være relevant å gi sauebonden muligheten til å se gjennom oppsynsturer som er gjennomført, sannsynligvis i form av en liste der hver oppføring viser dato og navn på beiteområde. Dersom listen viser 20 oppsynsturer om gangen vil det resultere i 20 lesinger. Da har klienten lastet ned *alle* registreringene til 20 turer, som eksempelvis blir 1 000 registreringer med 50 registreringer i gjennomsnitt per tur. Det er lite sannsynlig at bonden kommer til å klikke seg inn på alle de 20 turene, og man har dermed lastet ned unødvendig mye data. Dersom det brukes en lazy loading liste og bonden ønsker å scrolle ned til den eldste oppsynsturen, vil klienten ende opp med å laste ned *alle*



Figur 4.4: Firestore datastruktur: Alternativ 2

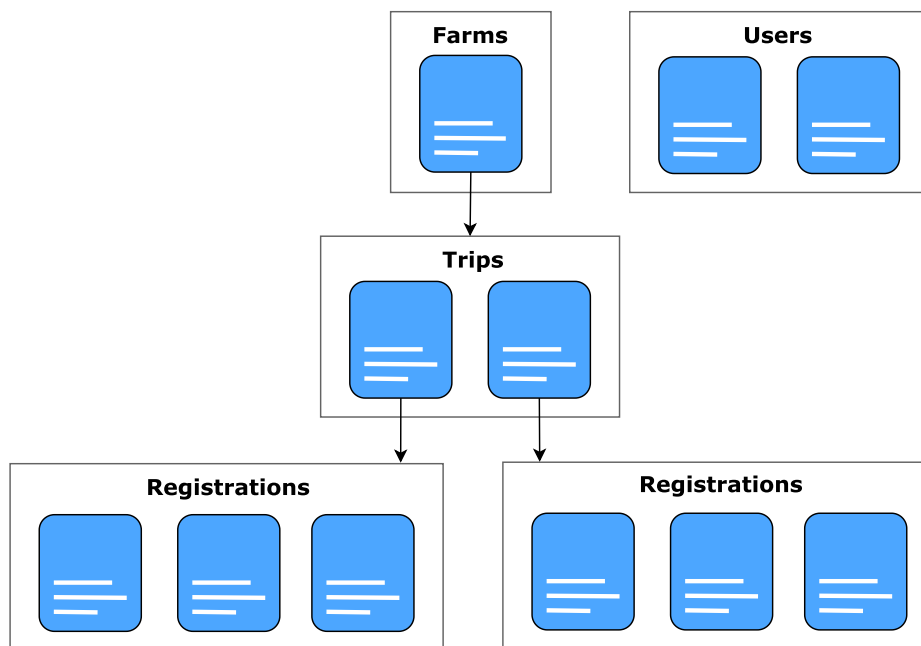
registreringer som noensinne er gjort for gården, noe som heller ikke er hensiktsmessig. Med denne datastrukturen hadde vi oppnådd få lesinger, men med noe dårlig design fordi klienten må laste ned unødvendig mye data. Færre lesinger er noe man bør vurdere og kostnadsestimere dersom man skal publisere systemet. Siden dette ikke var særlig relevant for prosjektet ønsket vi heller å gå for godt design, og vurderte en siste datastruktur.

Alternativ 3

På bakgrunn av vurderingene som ble gjort av de to foregående alternativene, landet vi til slutt på datastrukturen vist i figur 4.5. Her er "Farms" en topp-nivå samling med dokumenter som inneholder "Trips" som undersamling, og hvert trip-dokument inneholder "Registrations" som undersamling. Figuren inkluderer også topp-nivå-samlingen "Users" for komplettethet. Denne strukturen gjør det enkelt å hente ut oppsynsturer per gård, og registreringer per oppsynstur, som er gunstig. For "Trips" og "Registrations" unngår vi i stor grad å utføre spørringer, og vi trenger for disse ikke å tenke på id-felter da alle "Trips"-dokumenter ligger innunder tilhørende "Farms"-dokument, og alle "Registrations"-dokumenter ligger innunder tilhørende "Trips"-dokument. Siden samme person kan være sauebonde og oppsynsperson på samme tid, valgte vi å ikke skille mellom brukere av mobilapplikasjonen (oppsynspersoner) og brukere av webapplikasjonen (sauebønder).

4.3.2 Dokumentstruktur

Denne seksjonen illustrerer og forklarer innholdet i dokumentene, nærmere bestemt hvilke felter de ulike dokumentene inneholder, hvilken type disse er av, samt en forklaring av verdiene deres.



Figur 4.5: Firestore datastruktur: Alternativ 3

Users

Feltene i "Users"-dokumentene vises i kodeliste 4.1. Dokumentene inneholder fullt navn, e-postadresse og telefonnummer til en bruker. De inneholder ikke passord fordi systemet bruker Firebase Authentication, som tar seg av hashing og lagring av passord.

Kodeliste 4.1: Felter i Users-dokumentet

```

{
  "name": string,
  "email": string,
  "phone": string,
}
  
```

Farms

"Farms"-dokumentene blir tilegnet sauebonde-brukeren sin uid³ som id. Dokumentets felter vises i kodeliste 4.2. Feltene "name" og "address" inneholder gårdsnavn- og adresse. Etterhvert som sauebonden definerer øremerker, slips, beiteområder og oppsynspersonell legges tilsvarende felter til i dokumentet. "eartags" er en map med heksadesimal fargeverdi som nøkkel og bool som verdi. True betyr at øremerkefargen tilhører egen gård, false betyr at den tilhører noen andres gård.

³Unique Identifier

"ties" er en tilsvarende map, men verdiene er tall som gjenspeiler slipsfargens betydning, altså antall lam. "maps" er en map med navn på beiteområdet som øverste nøkkel, og en map med koordinatene til områdets nordvestlige og sørøstlige hjørner som verdi, som videre er delt inn i en map med nøklene "latitude" og "longitude" og tall som verdi. Det siste feltet "personnel" er en liste som inneholder e-postadressen til brukere som er registrert som oppsynspersonell for gården. I tillegg har hvert "Farms"-dokument en undersamling med "Trips"-dokumenter, som beskrives nærmere i neste avsnitt.

Kodeliste 4.2: Felter i Farms-dokumentet

```
{
  "name": string,
  "address": string,
  "farmNumber": string,
  "eartags": {
    string : bool,
    ...
  },
  "ties": {
    string: number,
    ...
  },
  "maps": {
    string: {
      "northWest": {
        "latitude": number,
        "longitude": number
      },
      "southEast": {
        "latitude": number,
        "longitude": number
      }
    }
  },
  ...
},
"personnel": [string, ...]
}
```

Trips

Feltene i "Trips"-dokumentene vises i kodeliste 4.3. "mapName" forteller hvilket beiteområde oppsynsturen foregikk på, "personnelEmail" inneholder hvem som gikk oppsynstur, "startTime" og "stopTime" er henholdsvis start- og sluttidspunkt, og "track" er en liste med enhetens posisjoner i løpet av turen. Hvert "Trips"-

dokument har også en subcollection med "Registrations"-dokumenter, som beskrives i neste avsnitt.

Kodeliste 4.3: Felter i Trips-dokumentet

```
{
  "mapName": string,
  "personnelEmail": string,
  "startTime": timestamp,
  "stopTime": timestamp,
  "track": [
    {
      "latitude": number,
      "longitude": number
    },
    ...
  ]
}
```

Registrations

Feltene i "Registrations"-dokumentene vises i kodelistene 4.4 - 4.6. Her er det fem ulike dokument-varianter; sau, skade, kadaver, rovdyr og notat. Felles for variantene er et "type"-felt for å skille de fra hverandre og et "timeStamp"-felt for registreringstidspunkt, i tillegg til feltene "devicePosition" og "registrationPosition", som vist i kodeliste 4.4. Posisjons-feltene inneholder henholdsvis enhetens posisjon da registreringen ble gjort, og posisjonen til det som ble registrert i form av en map med nøklene "latitude" og "longitude" og tall som verdier. Variantene av registrerings-dokumenter beskrives nærmere i de neste avsnittene.

Kodeliste 4.4: Felter som er felles for alle Registrations-dokumenteter

```
{
  "type": string,
  "timestamp": timestamp,
  "devicePosition": {
    "latitude": number,
    "longitude": number
  },
  "registrationPosition": {
    "latitude": number,
    "longitude": number
  },
}
```

Alle vanlige saueregistrerings-dokumenter (se kodeliste 4.5) inneholder feltene "sheep", "lambs", "white", "brown", "black" og "blackHead", som beskriver antall sau og lam totalt, antall lam og antall av de ulike ullfargene. "type"-feltet vil for denne typen registrering ha verdien "sheep". For nærregistreringer vil dokumentene i tillegg inneholde felter som gjenspeiler øremerke- og slipsfargene som er definert av sauebonden. Eksempler på disse er "blueEar" og "redTie", med verdier som forteller antallet av hver av disse.

Kodeliste 4.5: Unike felter i Registrations-dokument for registrering av sau

```
{  
  "sheep": number,  
  "lambs": number,  
  "white": number,  
  "brown": number,  
  "black": number,  
  "blackHead": number,  
  "blueEar": number,  
  "redTie": number  
}
```

Dokumenter for registrering av skadd sau inneholder feltene vist i kodeliste 4.6. "type"-feltet vil for denne registreringstypen ha verdien "injuredSheep". "eartag"-feltet gjenspeiler øremerke-id på formatet "XX-XX-YYYYYYY-ZZZZZ", der X er landskoden (typisk MT-NO), Y er gårds-identifikatoren på syv eller åtte siffer, og Z er individ-nummeret på fire eller fem siffer. Dokumentene inneholder også feltene "tieColor" med slipsfargens heksadesimale verdi, "injuryType" som enten er "Beinskade", "Blodutredning", "Hodeskade" eller "Annen", "severity" som enten er "Moderat" eller "Alvorlig", og "note" med eventuell tilleggsinformasjon som oppsynspersonen la til under registreringen.

Kodeliste 4.6: Unike felter i Registrations-dokumentet for registrering av skadd sau

```
{  
  "eartag": string,  
  "tieColor": string,  
  "injuryType": string,  
  "severity": string,  
  "note": string  
}
```

Dokumenter for registrering av kadaver inneholder feltene vist i kodeliste 4.7. "type"-feltet har for denne registreringstypen verdien "cadaver". Feltene "eartag", "tieColor" og "note" inneholder samme data som for skadd sau. "photos"-feltet inneholder null til tre lenker til eventuelle bilder som oppsynspersonen har tatt av kadaveret.

Kodeliste 4.7: Unike felter i Registrations-dokumentet for registrering av kadaver

```
{  
  "eartag": string,  
  "tieColor": string,  
  "photos": [string],  
  "note": string  
}
```

Dokumenter for registrering av rovdyr inneholder feltene vist i kodeliste 4.8. Verdien til "type"-feltet er "predator" og verdien til "species"-feltet er enten "Bjørn", "Ulv", "Jerv", "Gaupe", "Havørn" eller "Kongeørn". "quantity"-feltet beskriver hvor mange rovdyr som ble sett, og "note" er et eventuelt notat.

Kodeliste 4.8: Unike felter i Registrations-dokumentet for registrering av rovdyr

```
{  
  "species": string,  
  "quantity": number  
  "note": string  
}
```

"type"-feltet i dokumenter for registrering av notat har verdien "note" og feltet "note" inneholder notatet som ble skrevet.

Kodeliste 4.9: Unike felter i Registrations-dokumentet for registrering av notat

```
{  
  "note": string  
}
```

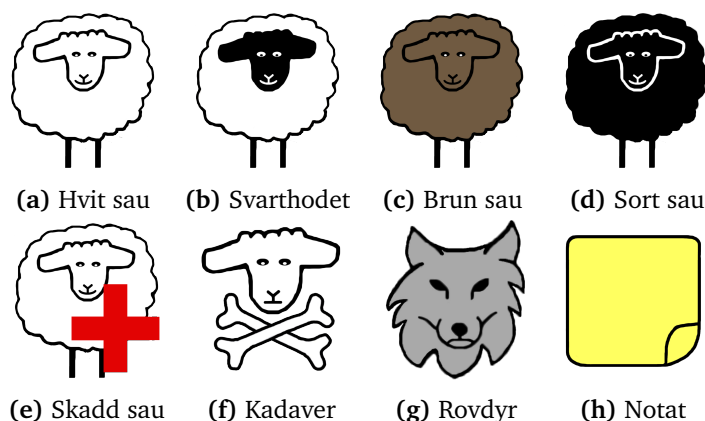
Kapittel 5

Design av brukergrensesnitt

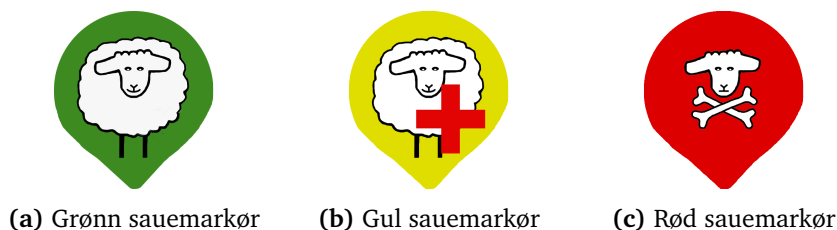
Dette kapittelet presenterer brukergrensesnittene til webapplikasjonen og mobilapplikasjonen, og hvordan de henger sammen. Designet tok utgangspunkt i papirprototypene i vedlegg B for webapplikasjonen og vedlegg C for mobilapplikasjonen, som begge ble tegnet i forkant av utviklingen. Avslutningsvis begrunnes overordnede designvalg som er gjort på tvers av systemet.

5.1 Egne ikoner

For å gi webapplikasjon og mobilapplikasjon et særpreg tegnet vi ikonene som vises i figur 5.1. Ikonene er hyppigst brukt i mobilapplikasjonen, men brukes også på kartet i webapplikasjonens oppsynstur-fane som beskrives i seksjon 5.2.3. For å plassere ikonene på kartet ble det også laget kartmarkører, som vist i figur 5.2, der markøren er grønn, gul eller rød basert på registreringens alvorlighetsgrad. Markøren er grønn for registreringer av typene sau og notat, gul for skadd sau og rød for kadaver og rovdyr.



Figur 5.1: Egne ikoner



Figur 5.2: Egne kartmarkører med utvalg av ikoner

5.2 Webapplikasjon

De følgende seksjonene presenterer de ulike sidene i webapplikasjonen og beskriver deres funksjonalitet. For å ta mindre plass i rapporten er nettleservinduet nedskalert i noen av bildene, og disse gjengir dermed ikke en presis brukeropplevelse fordi det er trangere mellom widgets og generelt mindre oversiktlig.

5.2.1 Innlogging & opprettelse av brukerkonto

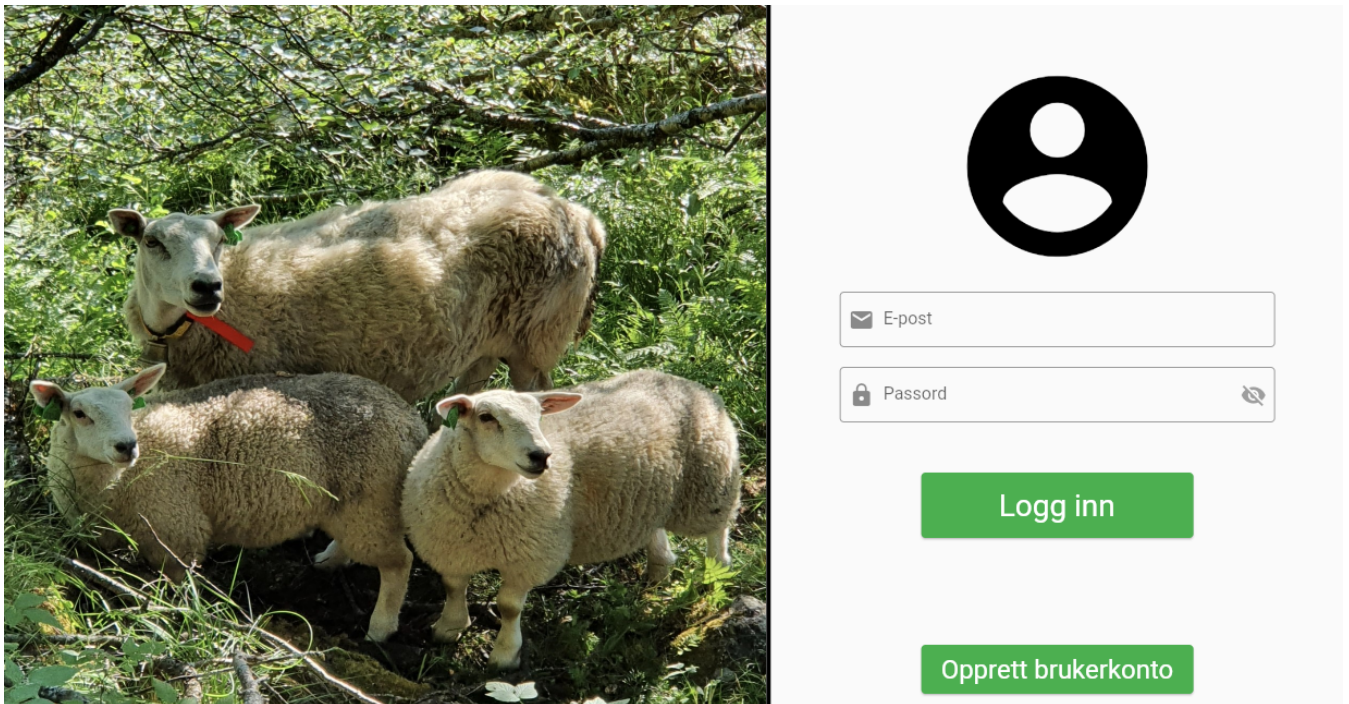
Skjermene for innlogging og brukerregistrering vises i figur 5.3 og 5.4. Gjennom passord-feltens øye-ikon har man mulighet til å vise passordet i klartekst om man ønsker det, som vist i figur 5.5a. Alle felter har tilhørende validerings-funksjon. Dersom man har skrevet inn noe ugyldig og trykker på enter-tasten, eller klikker på knappen for innlogging eller registrering, vil det gis tilbakemelding på hva som er galt. Tilbakemeldingene er enkle og presise, med unntak av tilbakemeldingen for ugyldig kombinasjon av e-post og passord ved innlogging. Den er obskur av sikkerhetsmessige årsaker, og røper ikke om en bruker med e-posten allerede eksisterer eller om det bare er passordet som er feil. Dette er generelt god praksis, selv om brukerregistrerings-skjermen riktignok kan røpe at en e-post allerede er i bruk. Tilbakemeldingene på ugyldig input er rødfarget, og kantene rundt de aktuelle feltene det gjelder for markeres også rødt, som vist i figur 5.5. Når man påbegynner retting av input, vil feltene valideres i sanntid, slik at feedback-meldingene og rødfargen forsvinner så fort det som er skrevet inn blir gyldig. Etter en suksessfull innlogging eller brukerregistrering lander man på hovedsiden, som er delt inn i de tre fanene "Årsrapporter", "Oppsynsturer" og "Min side", som vist i figur 5.6.

5.2.2 Min side

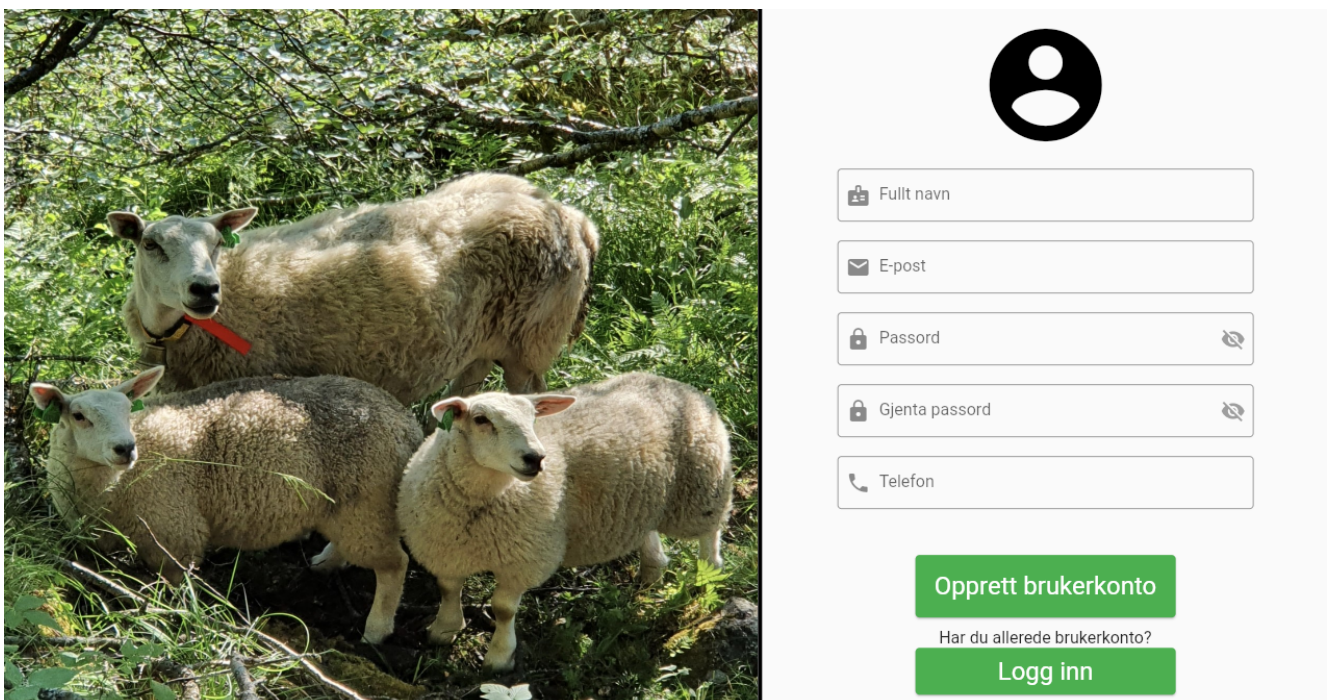
Denne seksjonen tar for seg fanen "Min side", som består av en meny med valgene "Gård", "Beiteområder", "Øremerker", "Slips" og "Oppsynspersonell".

Min gård

Menyvalget "Gård" åpner siden vist i figur 5.6, der sauebonden kan legge inn gårdens navn, adresse og gårdsnummer. I likhet med feltene på innlogging- og



Figur 5.3: Webapp: Innlogging



Figur 5.4: Webapp: Opprette brukerkonto

(a) Ugyldig brukerregistrering

(b) Ugyldig innloggingsdetaljer

Figur 5.5: Webapp: Tilbakemelding ved ugyldig input

registreringssiden, valideres også disse når brukeren klikker på lagre-knappen eller trykker på enter-tasten. Valideringsmeldingene er "Skriv gårdsnavn" og "Skriv gårdsadresse" dersom de to førstnevnte feltene har færre enn to bokstaver eller tall. Valideringsmeldingene for gårdsnummer-feltet er "Skriv gårdsnummer" dersom det er tomt, "7-8 siffer" dersom antall siffer er utenfor gyldig område, og "Gårdsnummer kan kun bestå av siffer" dersom det ikke kun inneholder siffer. Vi ønsket ikke å validere at gårdsnummer er unike fordi man kan skrive feil, og fordi hvem som helst kan opprette en brukerkonto og okkupere et nummer. Da hadde det vært uheldig å hindre gården som faktisk har gårdsnummeret i å bruke det. I tillegg, som beskrevet i seksjon 4.3.1, er det ingen relasjon mellom gårdene i databasen (eller i grensesnittet), så like gårdsnummer vil teknisk sett ikke være et problem. Dersom alle feltene er gyldig fylt inn, gis tilbakemeldingen "Gårdsinformasjon lagret" i grønn tekst. Gårdsnavn og gårdsnummer vil bli brukt i mobilapplikasjonen under oppsynstur, dette blir beskrevet nærmere i seksjon 5.3.4 og 5.3.6. Alle feltene på denne siden brukes i årsrapporten, mer om denne i seksjon 7.8.

Mine beiteområder

Menyvalget "Beiteområder" åpner siden vist i figur 5.7, der sauebonden kan definere beiteområder som oppsynspersonell skal gå oppsynstur i. På skjermen ser man en generell beskrivelse av hva som kan gjøres på siden, samt en tabell som

Årsrapporter	Oppsynsturer	Min side

Min gård

Navn

Gårdsnavn

Adresse

Gårdsadresse

Nummer

Gårdsnummer

Figur 5.6: Webapp: Min gård

inneholder utsnitt, navn og koordinater til to tidligere definerte områder. Utsnittene vises for å gjøre det enklere å gjenkjenne det aktuelle beiteområdet.

I tabellen med beiteområder kan man klikke på søppelspannet for å slette et område og endre navn på området i navn-feltet. Dersom man begynner en navneendring, vil søppelspann-ikonet erstattes med knapper for å lagre eller avbryte endringen. Figur 5.8 viser skjermen etter å ha klikket på "Legg til"-knappen i nedre høyre hjørne. De eksisterende kartutsnittene blir mindre for å gi plass til det interaktive kartet som nå vises. Sauebonden får instruksjoner om å klikke og holde på nordvestlig hjørne, og deretter sørøstlig hjørne av området som skal defineres. For å sørge for at sauebonden får med seg instruksjonene, er de viktigste delene ("Klikk og hold", "nordvestlig hjørne" og "sørøstlig hjørne") uthevet med fet skrift. Dersom sauebonden forsøker å plassere sørøstlig hjørne feil i forhold til nordvestlig hjørne gis det beskjed om dette. Når hjørnene er plassert, rammes beiteområdet inn i en rød firkant som vist i figur 5.9. Sauebonden er nødt til å gi beiteområdet et navn før kartet kan lagres ved å klikke på "Lagre"-knappen. Det er også mulig å avbryte definering av et nytt beiteområde. Når et nytt beiteområde lagres, eller prosessen avbrytes, går siden tilbake til sin opprinnelige tilstand, dvs. at kartet forsvinner, og størrelsen på eksisterende (og eventuelt nytt) kartutsnitt øker som i figur 5.7.

Mine øremerker

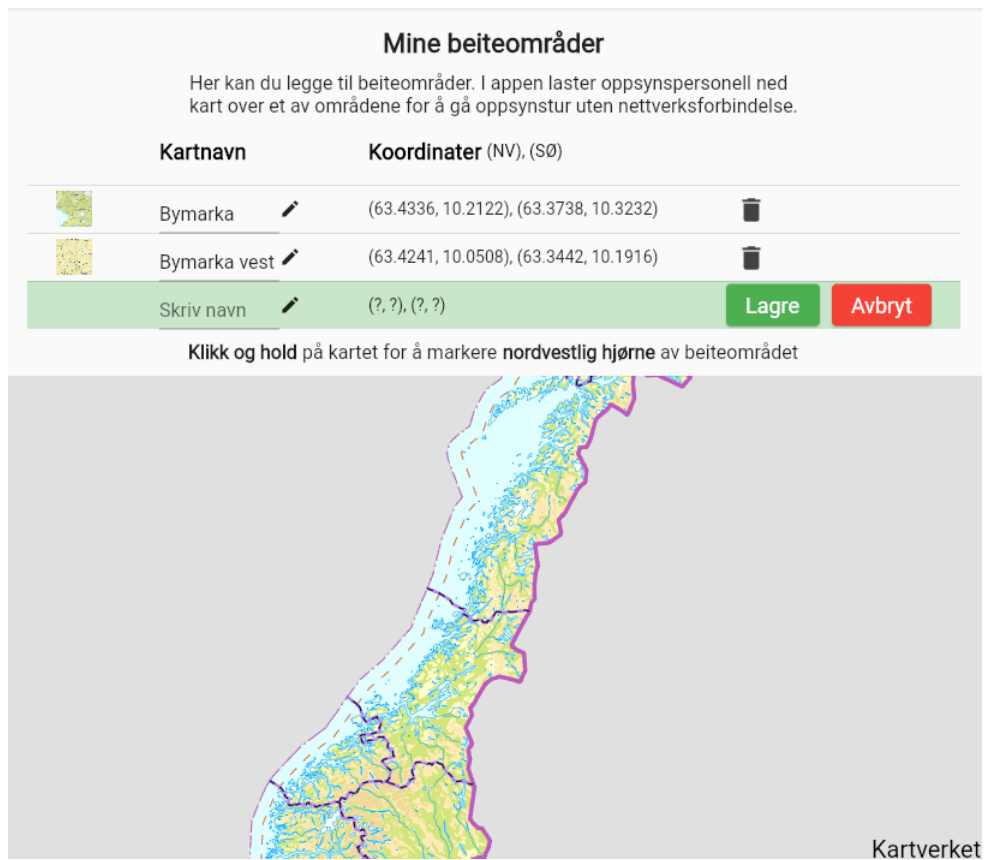
Menyvalget "Øremerker" åpner siden vist i figur 5.10, der sauebonden kan definere øremerkefarger som oppsynspersonell kan komme til å møte på, både sine egne øremerker og andres. På skjermen ser man informasjon om hva som kan gjøres på siden, i tillegg til en tabell med eksisterende øremerkefarger. I kolonnen "Øremerker" er det en nedtrekksmeny der hvert alternativ består av et farget



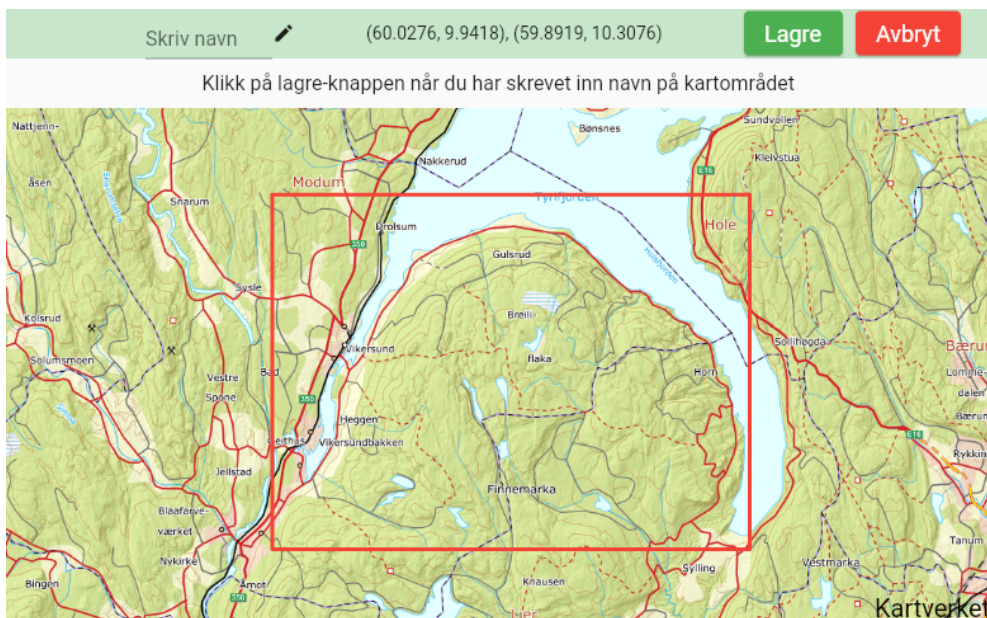
Figur 5.7: Webapp: Mine beiteområder

øremerke-ikon og fargens navn i tekstform. Kolonnen "Eier" inneholder en bryter med alternativene "Meg" og "Annen", som viser om sauer med den aktuelle øremerkefargen tilhører egen eller andres gård. Alternativet som er valgt har grønn bakgrunnsfarge for "Meg" og oransje bakgrunnsfarge for "Annen". Valgt alternativ har i tillegg fet tekst for å tydeliggjøre at det er valgt. Alternativet som ikke er valgt har grå bakgrunnsfarge. Siste kolonne inneholder også her et klikkbart søppelspenn-ikon for å slette øremerkefargen.

Figur 5.11 viser prosessen med å endre øremerkefarge fra grønn til gul. Figur 5.11a viser nedtrekksmenyen i åpen tilstand, og man ser her at grønn farge er valgt fordi den har grå bakgrunn. Når man klikker på gul, vil nedtrekksmenyen lukkes og siden ser ut som på figur 5.11b, der endringen er markert med grønn bakgrunnsfarge. Bakgrunnsfargen til brytere i "Eier"-kolonnen blir også grønn ved bytte av eier. En endring lagres ved å klikke på "Lagre" eller angres ved å klikke på "Avbryt". Alle endringer kan angres, inkludert sletting av slips. Dersom sauebonden er i ferd med å definere to eller flere øremerker med samme farge, gis det beskjed om at dette ikke er tillatt, og lagre-knappen blir grå og deaktivert. Når dette er rettet opp blir lagre-knappen grønn og klikkbar igjen. Når alt er i orden og "Lagre"-knappen klikkes på, gis det tilbakemelding på at endringene er lagret i form av grønnfarget tekst.



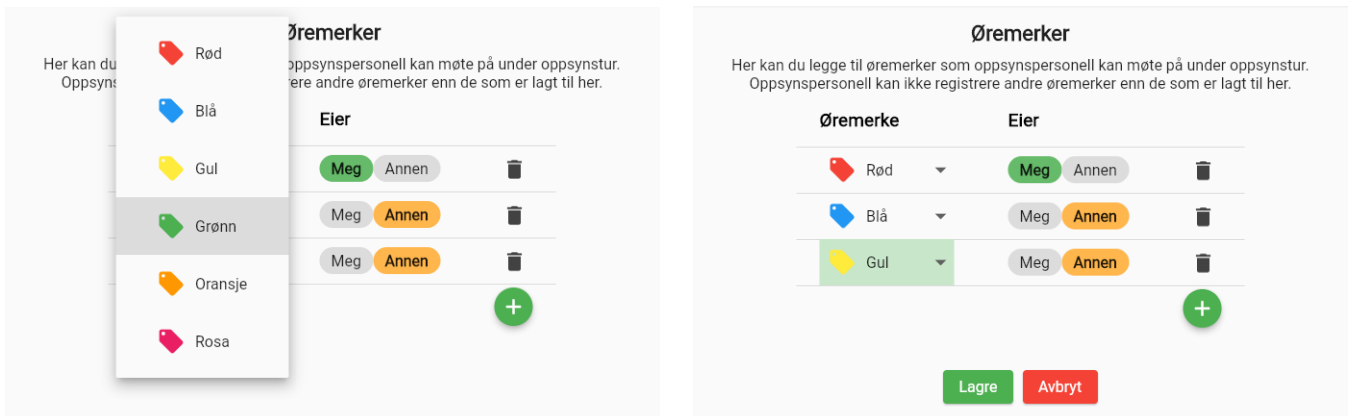
Figur 5.8: Webapp: Legg til nytt beiteområde



Figur 5.9: Webapp: Legg til nytt beiteområde, markert



Figur 5.10: Webapp: Mine øremerker



(a) Endre øremerkefarge

(b) Endret øremerkefarge er markert

Figur 5.11: Webapp: Endre øremerkefarge



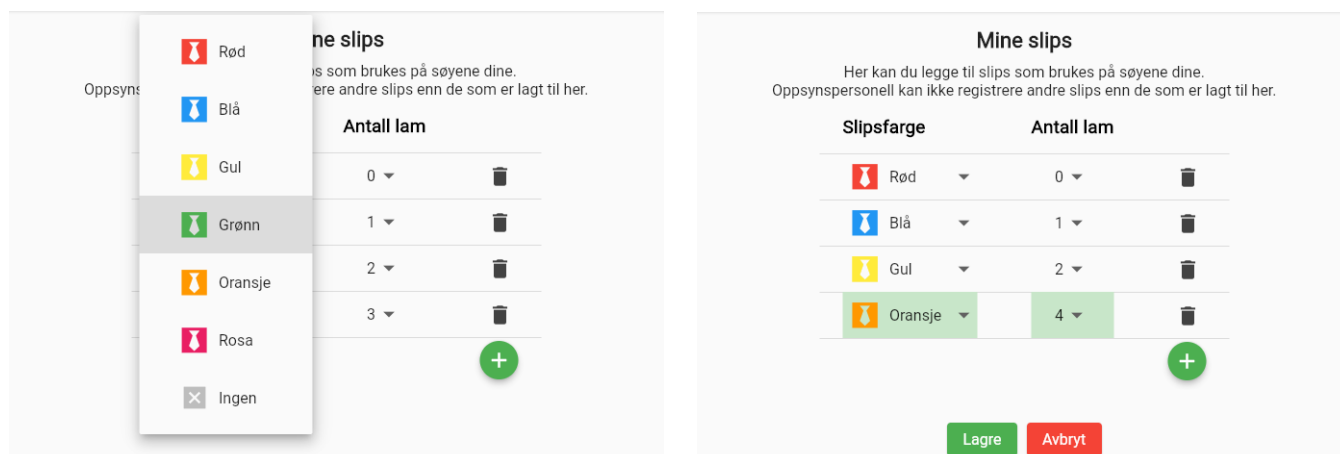
Figur 5.12: Webapp: Mine slips

Mine slips

Menyvalget "Slips" åpner siden vist i figur 5.12, der sauebonden kan legge til slipsfargene som sauene bærer, og hvor mange lam fargen representerer. Denne siden er svært lik øremerke-siden og fungerer i all hovedsak på samme måte, men med to avvik. Første avvik er at brukerinntput foregår gjennom to nedtrekksmenyer. Andre avvik er at det er mulig å definere at "ingen" slips har en betydning, fordi dette er praksis for noen sauebønder. Figur 5.13a viser at det siste alternativet i nedtrekksmenyen for slipsfarge har et "X"-ikon, som representerer søyer uten slips. Figur 5.13 viser endring av slipsfarge og tilhørende antall lam. Endrede tabellceller utheves også her med grønn bakgrunnsfarge. Dersom sauebonden er i ferd med å definere to eller flere slips med samme farge, eller to eller flere like antall lam, gis det tilbakemelding på at dette ikke er tillatt, på samme måte som på "Øremerke"-siden.

Mitt oppsynspersonell

På siden som åpnes etter å ha klikket på menyvalget "Oppsynspersonell" kan man administrere gårdens oppsynspersonell. Man legger til personell ved å klikke på plussknappen nede til høyre, og skjemen vil da se ut som i figur 5.14. Deretter må e-postadressen som er knyttet til oppsynspersonellets brukerkonto fylles inn før man klikker på lagre. Dersom brukerkontoen ikke eksisterer, gis det beskjed om dette. Hvis kontoen eksisterer blir e-postadressen lagt til i tabellen sammen med fullt navn og telefonnummer. Vi har gjort et bevisst valg om at sauebonden selv ikke vises i denne tabellen, selv om bonden har mulighet til å gjøre oppsyn for egen gård. Vi har implementert det på denne måten for å hindre at sauebonden fjerner seg selv fra lista, glemmer dette, og senere ikke får til å gjøre oppsyn selv.



(a) Endre slipsfarge

(b) Endret slipsfarge og betydning er markert

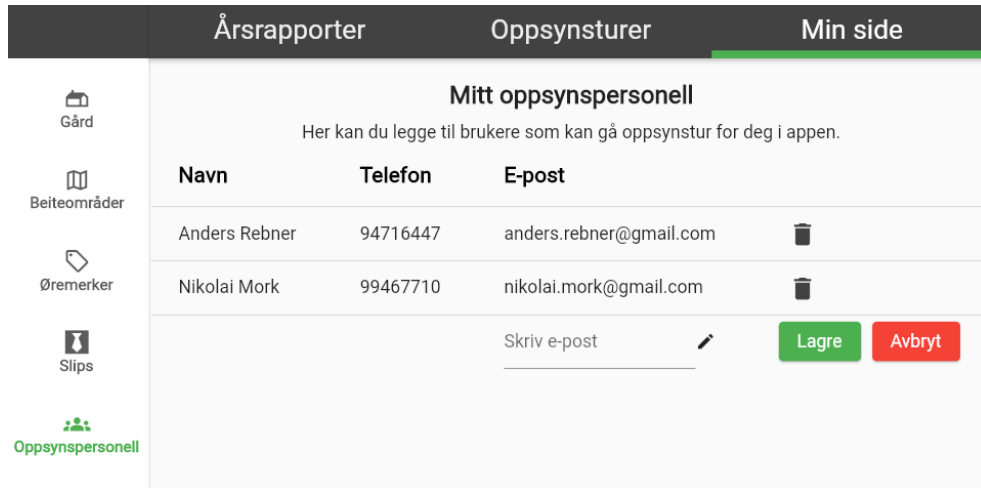
Figur 5.13: Webapp: Slipsendring

Dersom sauebonden likevel forsøker å legge til sin egen brukerkonto, vil det gis beskjed om at dette ikke er nødvendig.

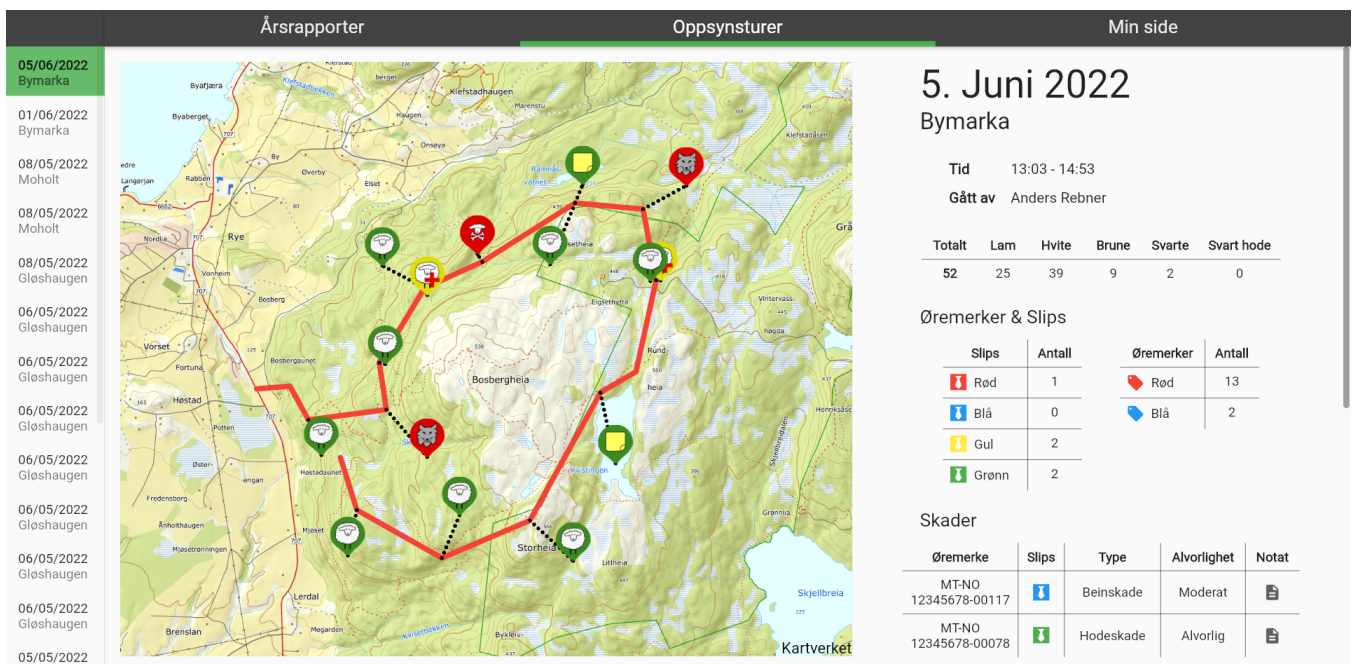
5.2.3 Oppsynsturer

I den midterste fanen "Oppsynstur" vil alle oppsynsturene som har blitt gått på vegne av gården vises. Dersom det ikke er noen fullførte turer, gis det beskjed om dette. Hvis det derimot har blitt gått oppsyn, vil hver tur vises i margen på venstre side. Hvert element i margen viser datoen oppsynsturen ble gjennomført og navnet på beiteområdet det ble gått i. Når sauebonden klikker på et element, vil siden presentere informasjon fra oppsynsturen, som vist i figur 5.15. Kartet viser ruten oppsynspersonen har gått (rød linje), posisjonen til de ulike registreringene (fargede markører med ikon) og siktlinjer fra oppsynspersonens posisjon da en registrering ble gjort til posisjonen til det som ble registrert (sort stiplet linje). Kartmarkører er farget grønne, gule eller røde etter alvorlighetsgrad. Kartet er interaktivt, så det er mulig å zoome og å navigere rundt. Alle markørene i kartet er klikkbare og åpner popup-vinduer som presenterer hva som ble registrert, inklusivt registreringstidspunkt, som vist i figur 5.17. Den høyre delen av oppsynstursiden gjengir dato i stor tekst, navn på beiteområdet, tidsrommet turen ble gått i og navnet på personen som gikk oppsyn. Videre viser den en hovedtabell som inneholder summert antall sauer, lam og antall av hver ullfarge. I tillegg har den én tabell som viser antall av hver slipsfarge, og én tabell som viser antall av hver øremerkefarge.

Hvis det har blitt registrert skadd sau, kadaver, rovdyr eller notat, vises disse i egne tabeller lengre ned på høyre del av siden (se figur 5.16), som da vil være scrollbar om ikke alle tabellene får plass på skjermen. Hver rad i disse tabellene



Figur 5.14: Webapp: Mitt oppsynspersonell



Figur 5.15: Webapp: Se oppsynstur

Årsrapporter
Oppsynsturer
Min side

05/06/2022
Bymarka

01/06/2022
Bymarka

08/05/2022
Moholt

08/05/2022
Moholt

08/05/2022
Gløshaugen

06/05/2022
Gløshaugen

06/05/2022
Gløshaugen

06/05/2022
Gløshaugen

06/05/2022
Gløshaugen

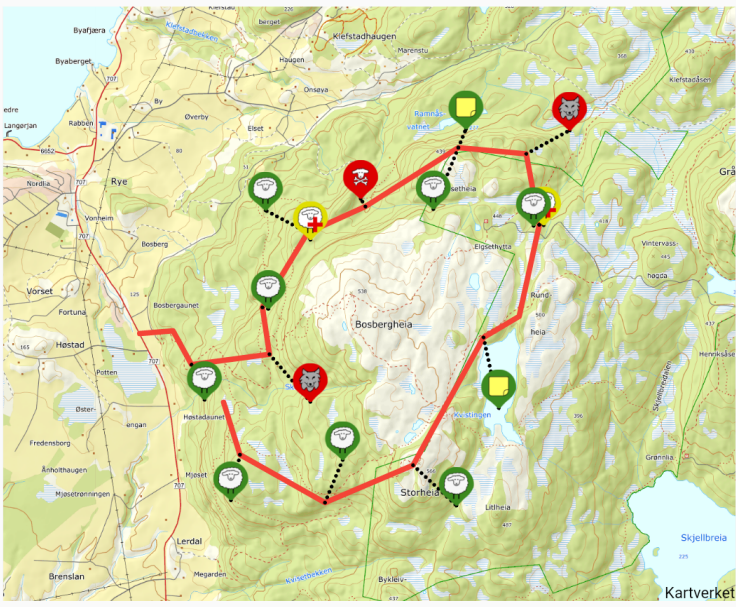
06/05/2022
Gløshaugen

06/05/2022
Gløshaugen

06/05/2022
Gløshaugen

06/05/2022
Gløshaugen

05/05/2022



Skader

Øremerke	Slips	Type	Alvorlighet	Notat
MT-NO 12345678-00117		Beinskade	Moderat	
MT-NO 12345678-00078		Hodeskade	Alvorlig	

Kadaver

Øremerke	Slips	Notat & bilder
MT-NO 12345678-00159		

Rovdyr

Art	Antall	Notat
Bjørn	3	
Jerv	1	

Notater

Mange som bader på sørenden av Kvistingen i dag. Ser ingen sauer ved vaninkanten.

Deler av gjerdet mot Ramnåsvannet ligger langs bakken. Stolpene må slås ned på nytt med slegge.

Figur 5.16: Webapp: Se oppsynstur (nedscrollet)

viser all informasjon fra registreringen, med unntak av notat og bilder for skade- og kadaverregistreringer, fordi varierende notatlengde og varierende antall kadaverbilder ville ha gjort siden svært lite oversiktlig. Derfor inneholder siste kolonne i de to tabellene et klikkbart dokument-ikon, som åpner samme popup-vinduer som markørklikk i kartet gjør (se figur 5.17). Alle popup-vinduene gir tekstlig tilbakemelding dersom det ikke ble registrert noe notat, eller det for kadaverregistrering ikke ble tatt noen bilder.

Oppsynstursiden er i sin helhet ment å gi sauebonden et så godt som mulig overblikk over hver enkelt oppsynstur. Høyre del av siden har som hensikt å gi en oppsummering av turen. Her kan sauebonden lese linje for linje og få med seg alt fra turen ferdig oppsummert. Det hadde krevd en del klikk, og vært mer utfordrende, å skaffe seg overblikk gjennom kartet alene. Kartet gir likevel en annen type oversikt i form av et visuelt perspektiv gjennom posisjon og grafikk. Det som ikke gjengis av den høyre delen av siden kan sauebonden finne i kartet, nemlig ruten som ble gått, siktelinjer mellom oppsynsperson og registreringer, og posisjonen til hver enkelt registrering.






Nærregistrert sau

 9 Sauer & Lam	 2 Blå
 5 Sauer	 0 Grønne
 4 Lam	 7 Rosa
 2 Hvite	 2 Blå
 3 Brune	 0 Gule
 0 Svarte	 0 Grønne
 4 Svarte hoder	 1 Rosa

20:03

(a) Nærregistrering sau

Avstandsregistrert sau

 7 Sauer & Lam
 4 Sauer
 3 Lam
 5 Hvite
 0 Brune
 2 Svarte
 0 Svarte hoder

20:01

(b) Avstandsregistrering sau

Saueskade
MT-NO
0015782-00235

Slips	 Blå
Type	Beinskade
Alvorlighet	Moderat

Ganske stygt skadet, halter bortover. Må hentes inn.

20:05

(c) Saueskade

Rovdyr

Art	Bjørn
Antall	3

Brunbjørn med to unger

20:07

(d) Rovdyr

Kadaver
MT-NO
0015782-00235 |  Grønn

Ingen tegn til rovdyrangrep, dødsårsak ukjent.




18:05

(e) Kadaver

Notat

Gjerdet rundt hytta på toppen av åsen har tegn til at et dyr satt fast en stund. Vi bør oppfordre til bruk av strømgjerde. Jeg oppnådde ikke kontakt med hytteeier, hun var ikke til stede.

20:12

(f) Notat

Figur 5.17: Webapp: Popups for å se enkeltregistreringer



Figur 5.18: Webapp: Generer årsrapport

5.2.4 Årsrapporter

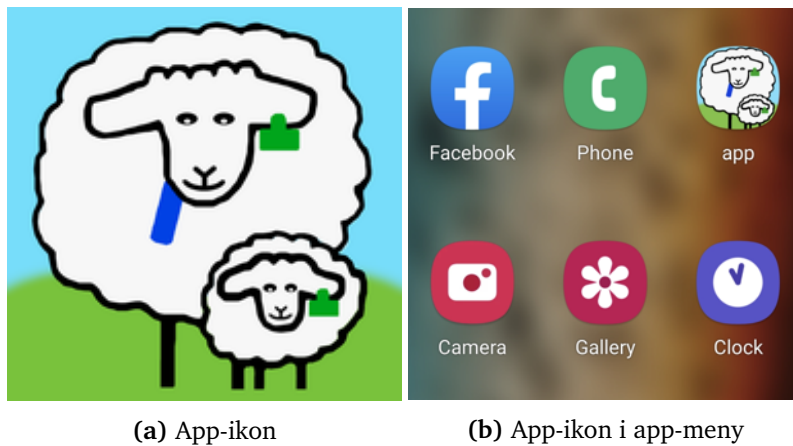
I fanen "Årsrapporter", vist i figur 5.18, kan sauebonden generere årsrapporter. Når man har valgt hvilket år man ønsker rapport for gjennom nedtrekksmenyen og trykker på "Last ned årsrapport"-knappen, vil rapporten bli generert og lagret lokalt på filsystemet. Alternativene i nedtrekksmenyen er kun de årstallene det har blitt gått oppsynstur i. Dersom det ikke har blitt gått noen oppsynsturer blir nedtrekksmenyen og knappen erstattet av en tilbakemelding om dette. Eksempel på årsrapport gis i vedlegg D, og generering av denne beskrives i seksjon 7.8.

5.3 Mobilapplikasjon

De følgende seksjonene presenterer mobilapplikasjonens brukergrensesnitt og beskriver funksjonaliteten som tilbys på hver skjerm. Seksjonen innledes med en kort presentasjon av applikasjonsikonet og en oversikt over programflyten.

5.3.1 App-ikon

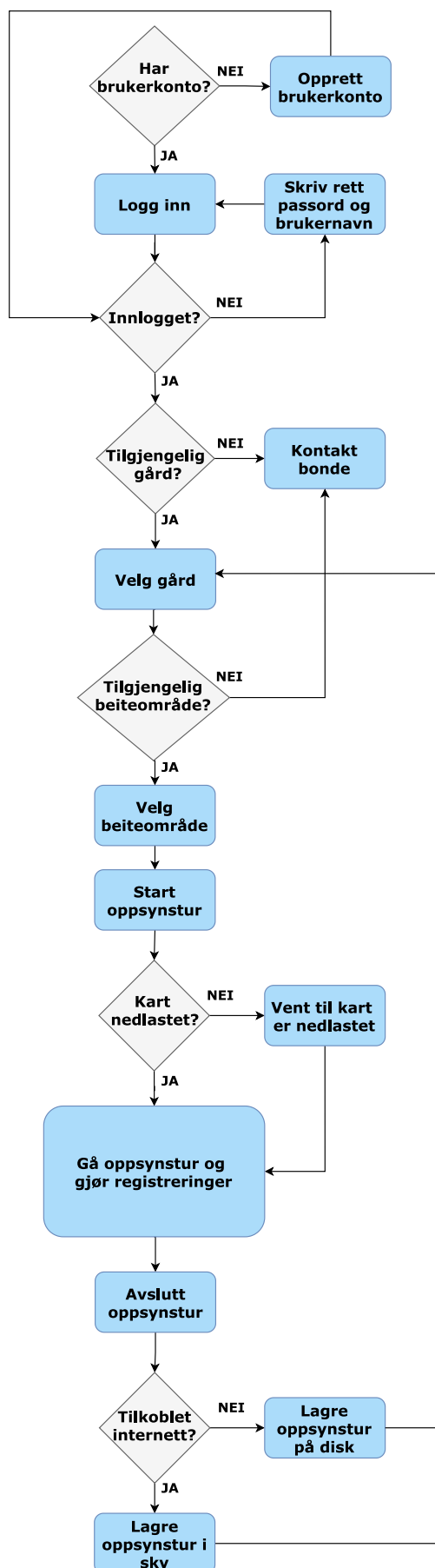
Vi ønsker med app-ikonet i figur 5.19a å representere det applikasjonen handler om. Derfor tegnet vi ei søye med et lam. Søyen har et blått slips og begge har grønne øremerker, da dette er sentralt ved registrering av sau. Kontrasten mellom slips- og øremerkefarge mot hvit ullfarge er bevisst høy for å tydeliggjøre øremerker og slips. Mange mobile enheter vil gjøre enhetsspesifikke modifiseringer på ikonet når det vises blant andre apper i app-menyen, eksempelvis avrunding av hjørner. Figur 5.19b viser hvordan ikonet set ut i app-menyen på en Samsung Galaxy S10+.



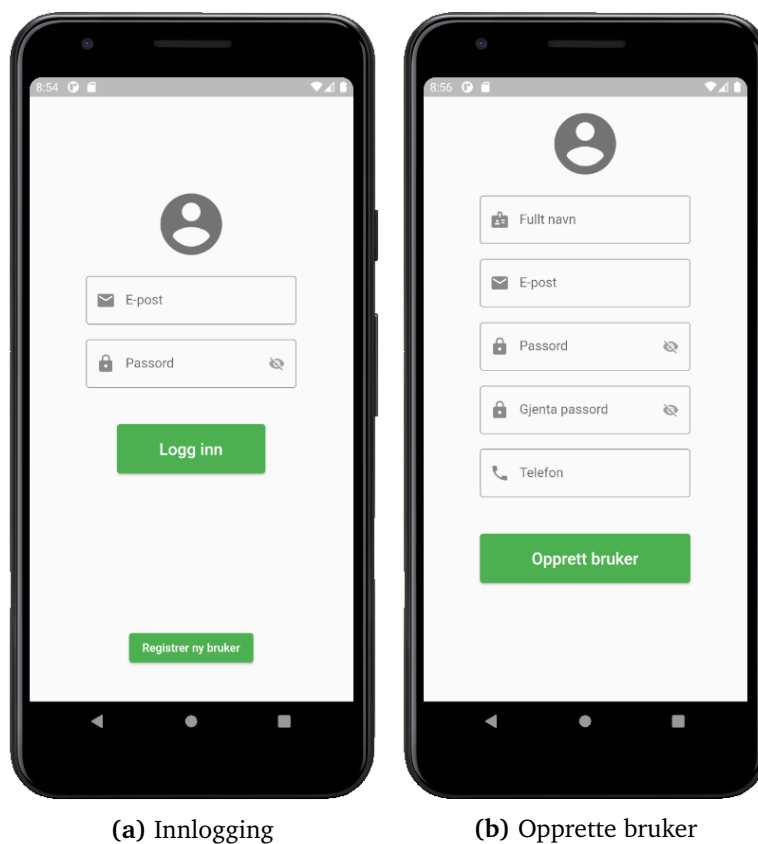
Figur 5.19: Mobilapp: App-ikon

5.3.2 Programflyt

Diagrammet i figur 5.20 viser mobilapplikasjonens programflyt fra brukerens perspektiv. Valg systemet gjør og prosesser som skjer i bakgrunnen er utelatt fra dette diagrammet. Boksen "Gå oppsynstur og gjør registreringer" er større enn de andre boksene fordi den omfavner en stor del av handlingene brukeren utfører. Det vil under en oppsynstur bli gjort et ubestemt antall registreringer av varierende type, og å inkludere flyten for hver av disse vil gjøre diagrammet unødvendig uoversiktlig. Hensikten med diagrammet er å vise samhandlingen mellom bruker og applikasjon, og at denne i all hovedsak følger én lineær flyt.



Figur 5.20: Mobilapp: Programflyt fra brukerens perspektiv



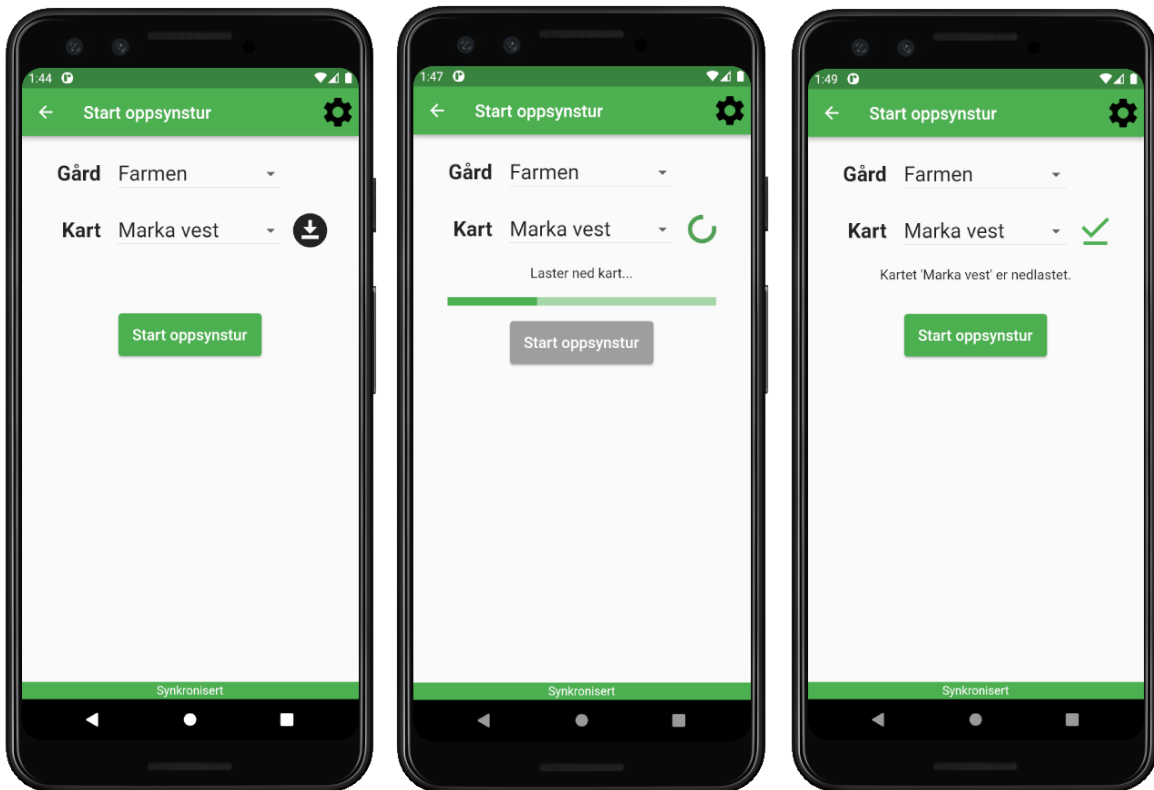
Figur 5.21: Mobilapp: Innlogging og brukerregistrering

5.3.3 Innlogging & opprettelse av brukerkonto

Figur 5.21 viser skjermene for innlogging og brukerregistrering. Disse fungerer på samme måte som i webapplikasjonen, som beskrevet i seksjon 5.2.1. Som konsekvens av at mobiltelefoner har mindre skjerm er det små forskjeller. Sidene er scrollbare fordi tastaturet tar en stor del av skjermen. Knappen nederst i høyre hjørne på tastaturet gjør det samme som å trykke på knappene for å logge inn eller opprette brukerkonto.

5.3.4 Start oppsynstur

Start oppsynstur-skjermen vises i figur 5.22. Dersom brukeren er gårdseier (sauebonde) eller er registrert som oppsynspersonell hos én eller flere gårder, vil gården(e) og tilhørende navn på kartområde(r) vises i nedtrekksmenyer. Man kan laste det ned ved å trykke på nedlast-ikonet til høyre for kartnavnet. Trykker man på knappen for å starte oppsynstur uten at det valgte kartet er lastet ned, vil kartet automatisk lastes ned før oppsynsturen starter.



(a) Start oppsynstur

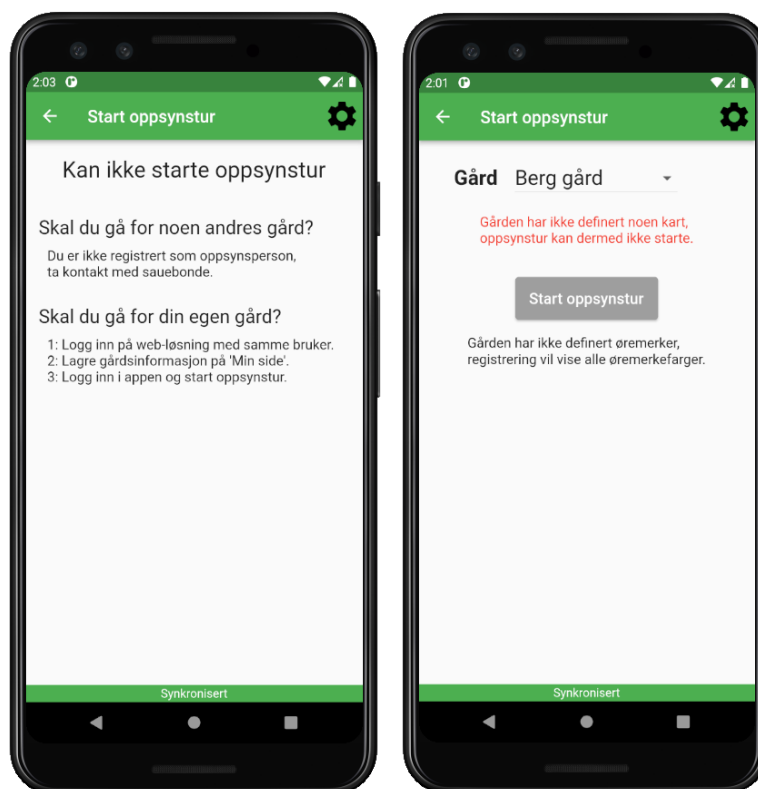
(b) Nedlasting av kart

(c) Kart nedlastet

Figur 5.22: Mobilapp: Start oppsynstur

Knappen for å starte oppsynsturen blir deaktivert mens kartet lastes ned. Siden nedlasting av kart kan kreve tid, gis det i tillegg tilbakemelding i form av en fremdriftsindikator. Indikatoren kan bevege seg sakte for store kartutsnitt, og det vises derfor en animert spinnende sirkel samtidig, som vist i figur 5.22b. Animasjonen er ment å betrygge brukeren om at det jobbes i bakgrunnen, og man kan unngå at brukeren avslutter applikasjonen basert på en feilaktig teori om at den har krasjet. Nedlasting av kart er en del av forberedelsene til oppsynsturer, og derfor er det mulig å laste ned kart uten å starte oppsynsturen. Om turen skal foregå i et område uten nettverksforbindelse må nedlastningen gjøres i forkant av turen på et sted der enheten har nettverksforbindelse. Når et kart er nedlastet, forblir det på enheten til applikasjonen avinstalleres.

Når brukeren bytter gård eller kart, sjekkes det automatisk om kartet er nedlastet. Det signaliseres at kartet er lastet ned med en grønn hake til høyre for det valgte kartet, som vist i figur 5.22c. Om kartet ikke er nedlastet vises det sorte nedlastings-ikonet, som vist i figur 5.22a.



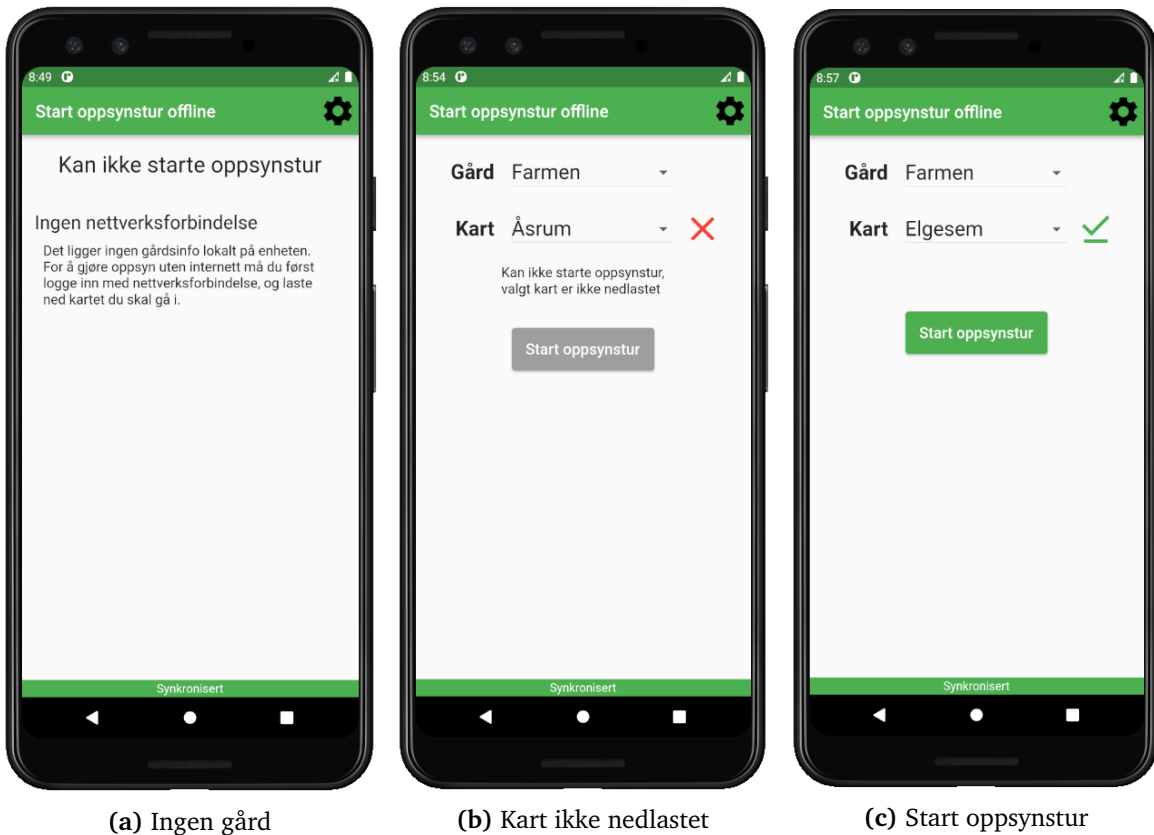
(a) Ingen gård

(b) Ingen kartområder

Figur 5.23: Mobilapp: Kan ikke starte oppsynstur

Start oppsynstur-skjermen håndterer alle spesielle tilfeller som kan oppstå. Figur 5.23a viser hvordan skjermen ser ut for oppsynspersoner som ikke har blitt lagt til hos noen gård, eller for sauebonden om gårdsinformasjon ikke har blitt registrert i webapplikasjonen. Teksten som vises gir instruksjoner om hva man må gjøre både dersom man skal gå for noen andres gård, eller sin egen. For at man skal kunne gå oppsynstur er det et krav at gården må ha definert minst ett beiteområde. Figur 5.23b viser hvordan skjermen ser ut dersom kravet ikke er oppfylt. I dette tilfellet har "Berg gård" heller ikke definert øremerkefarger, noe det er verdt å informere oppsynspersonen om. I praksis betyr dette bare at alle farger kan registreres i de ulike registreringsskjermene som beskrives senere i kapitlet.

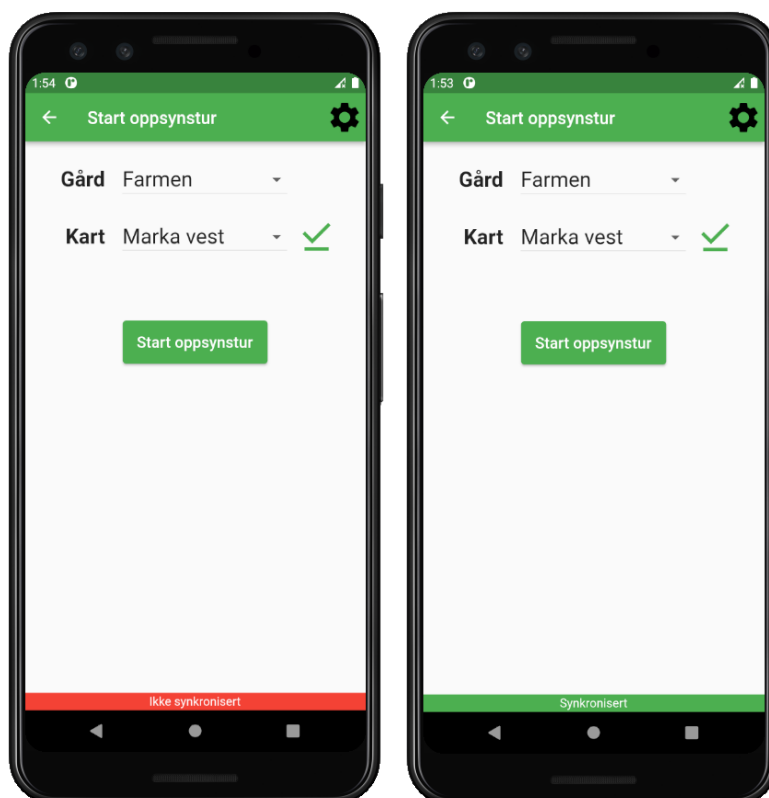
Siden det skal være mulig å gjennomføre oppsynstur i områder uten nettverksforbindelse vil applikasjonen starte direkte på en av skjermene vist i figur 5.24 dersom enheten ikke har nettverksforbindelse. Man kan bare gå oppsyn uten nettverk som den siste innloggede brukeren, mer om dette i seksjon 7.6. Applikasjonen starter på skjermen i figur 5.24a dersom ingen brukere har vært innlogget på enheten tidligere, eller dersom den sist innloggede brukeren ikke er definert som oppsynspersonell hos noen gård. På denne skjermen gis det en kort forklaring



Figur 5.24: Mobilapp: Start oppsynstur uten nettverksforbindelse

til hvorfor oppsynsturen ikke kan starte. Skjermen i figur 5.24b vises dersom en brukerkonto, som er registrert som oppsynspersonell, har vært innlogget tidligere, og valgt kartområde ikke er nedlastet. Til høyre for det valgte kartet vises et rødt kryss fordi kart ikke kan lastes ned uten nettverksforbindelse. I tillegg gis det informasjon om hvorfor oppsynsturen ikke kan gjennomføres. Figur 5.24c vises dersom oppsynstur uten nettverksforbindelse kan startes.

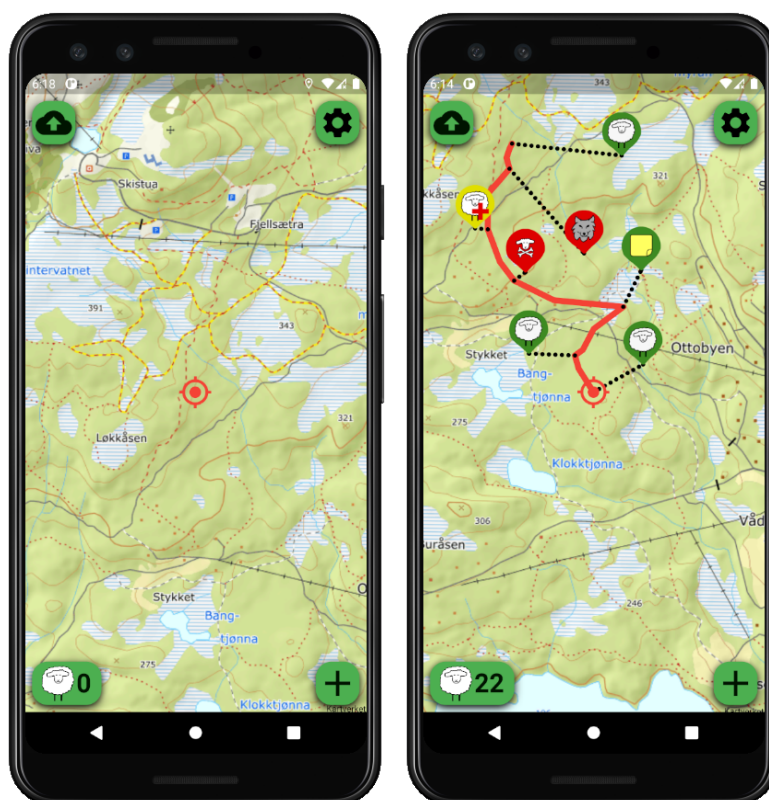
På startskjermen kan man se om enheten er synkronisert eller ikke, med andre ord om alle oppsynsturer er lastet opp. Dersom én eller flere oppsynsturer ble avsluttet uten nettverksforbindelse, vil denne/disse ligge lokalt på enheten. Status gis i dette tilfellet av en rød statusbar nederst på siden med teksten "Ikke synkronisert", som vist i figur 5.25a. Oppsynsturen(e) lastes opp automatisk fra denne siden når enheten får nettverksforbindelse, og fargen på statusbaren blir grønn med teksten "Synkronisert", som vist i figur 5.25b.



(a) Ikke synkronisert

(b) Synkronisert

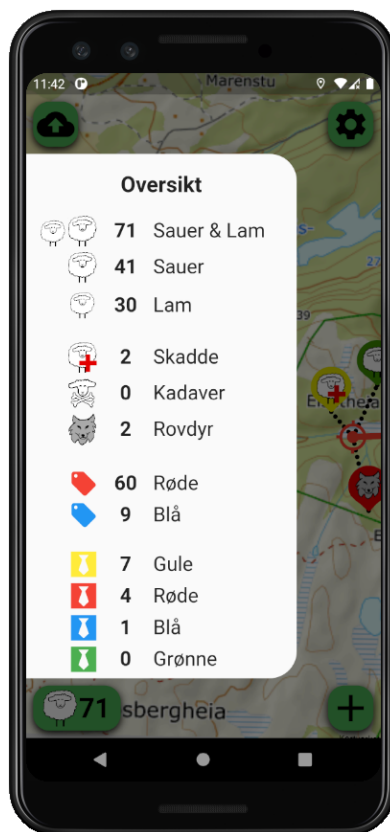
Figur 5.25: Mobilapp: Synkroniseringsstatus



(a) Hovedskjerm start

(b) Hovedskjerm underveis

Figur 5.26: Mobilapp: Hovedskjerm



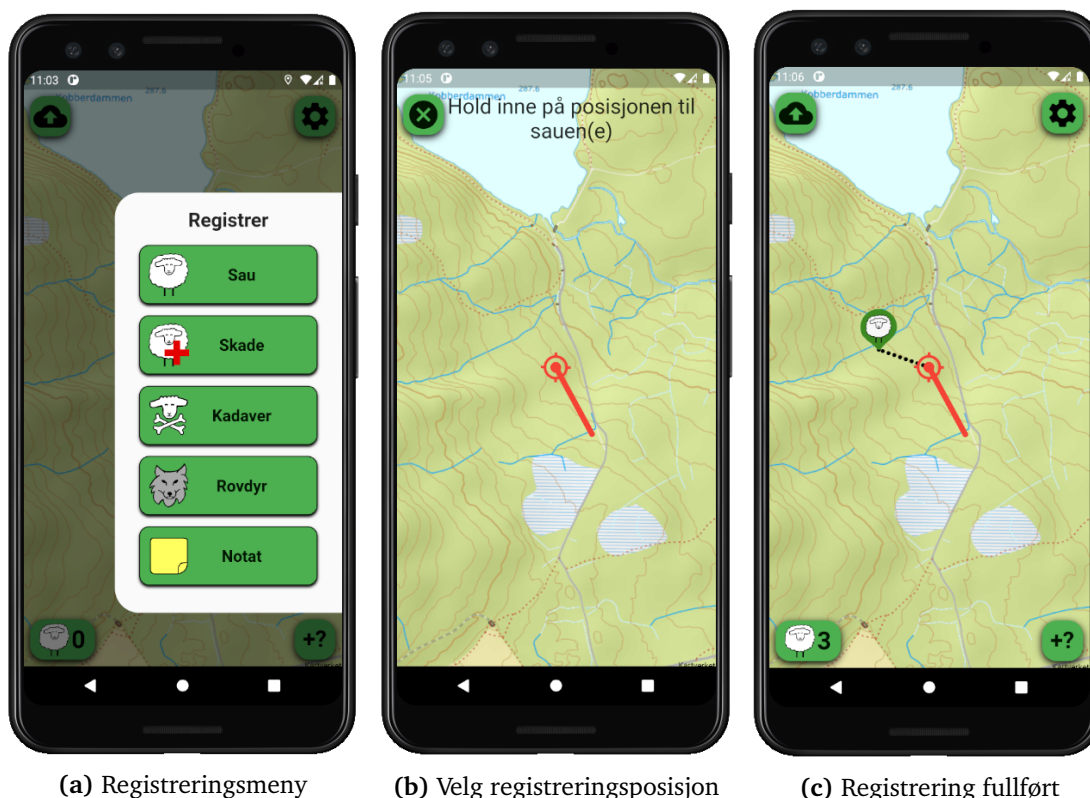
Figur 5.27: Mobilapp: Sommerturoversikt

5.3.5 Hovedskjerm

Applikasjonens hovedskjerm, vist i figur 5.26, inneholder blant annet et interaktivt kart som viser enhetens posisjon (rødt posisjons-ikon), stien som har blitt gått (rød linje), siktlinjer mellom enhetens og registreringenes posisjon (sort stiplet linje) og registreringenes posisjon (registreringsmarkører med ikon).

Når oppsynsturen nettopp har startet, vil skjermen se ut som i figur 5.26a. Når oppsynspersonen har fullført en registrering, vil siktlinje og markør med registreringsikon legges til på hovedskjermen. Når turen er godt underveis, ser skjermen ut som i figur 5.26b. Kartmarkørene er klikkbare og åpner popup-vinduer som viser hva som ble registrert på markørens posisjon. Disse vinduene er tilnærmet identiske som i webapplikasjonen, og er vist i figur 5.17.

Skjermen har i tillegg én knapp i hvert hjørne. Knappen i øvre venstre hjørne brukes til å avslutte og laste opp oppsynsturen (evt. forkaste), og beskrives i seksjon 5.3.8. Knappen øverst til høyre leder til et popup-vindu med innstillinger og illustreres i seksjon 5.3.9. Knappen nede til høyre leder til en meny som sklir inn fra høyre med knapper for hver registreringstype, og beskrives i seksjon 5.3.6. Knappen i nedre venstre hjørne viser antall sauer og lam som er registrert på oppsynsturen så langt (kadaver medregnes ikke), og åpner et vindu som sklir inn fra venstre. Vinduet, vist i figur 5.27, gir en grundigere turoversikt med summert antall sauer og lam, skadde, kadaver, rovdyr, øremerkefarger og slipsfarger. Dette vinduet blir scrollbart dersom innholdet blir for høyt. Begge vinduene som sklir inn fra sidene (altså summert turoversikt og registreringstyper) kan i tillegg til å åpnes ved å trykke på knappene åpnes ved å swipe innover fra skjermkanten.



Figur 5.28: Mobilapp: Registrering

5.3.6 Registreringsskjermer

Da oppsynspersonen stort sett kommer til å utføre vanlige sauregistreringer, åpnes sauregistrerings-skjermen (som beskrives i neste seksjon) ved å holde inne på sau(e)s posisjon på kartet. Dette er gjort for å spare tid og antall trykk. For å gjøre en annen type registrering trykker man på knappen i nedre høyre hjørne av hovedskjermen og menyen vist i figur 5.28a sklir inn fra høyre. Når ønsket registreringstype er valgt, går applikasjonen tilbake til en modifisert hovedskjerm der knappene i hjørnene er erstattet av en pulserende instruksjon og en knapp for å avbryte denne tilstanden, som vist i figur 5.28b. Instruksjonen er tilpasset valgt registreringstype, og ber oppsynspersonen om å holde inne på posisjonen til det som skal registreres. Når oppsynspersonen enten har holdt inne på kartet for å registrere sauer, eller holdt inne på kartet etter å ha valgt type registrering fra menyen, vil den aktuelle registreringsskjermen åpnes. Dersom oppsynspersonen trykker på registreringsskjermenes tilbake-ikon i øvre venstre hjørne, eller enhets tilbake-knapp, åpnes et popup-vindu som spør om registreringen skal avbrytes, for å unngå at data i registreringen slettes ved uhell. Når registreringsskjemaet er ferdig utfylt, vil siktlinje og registreringsmarkør legges til på hovedskjermen, som vist i figur 5.28c.

Registrering av sau

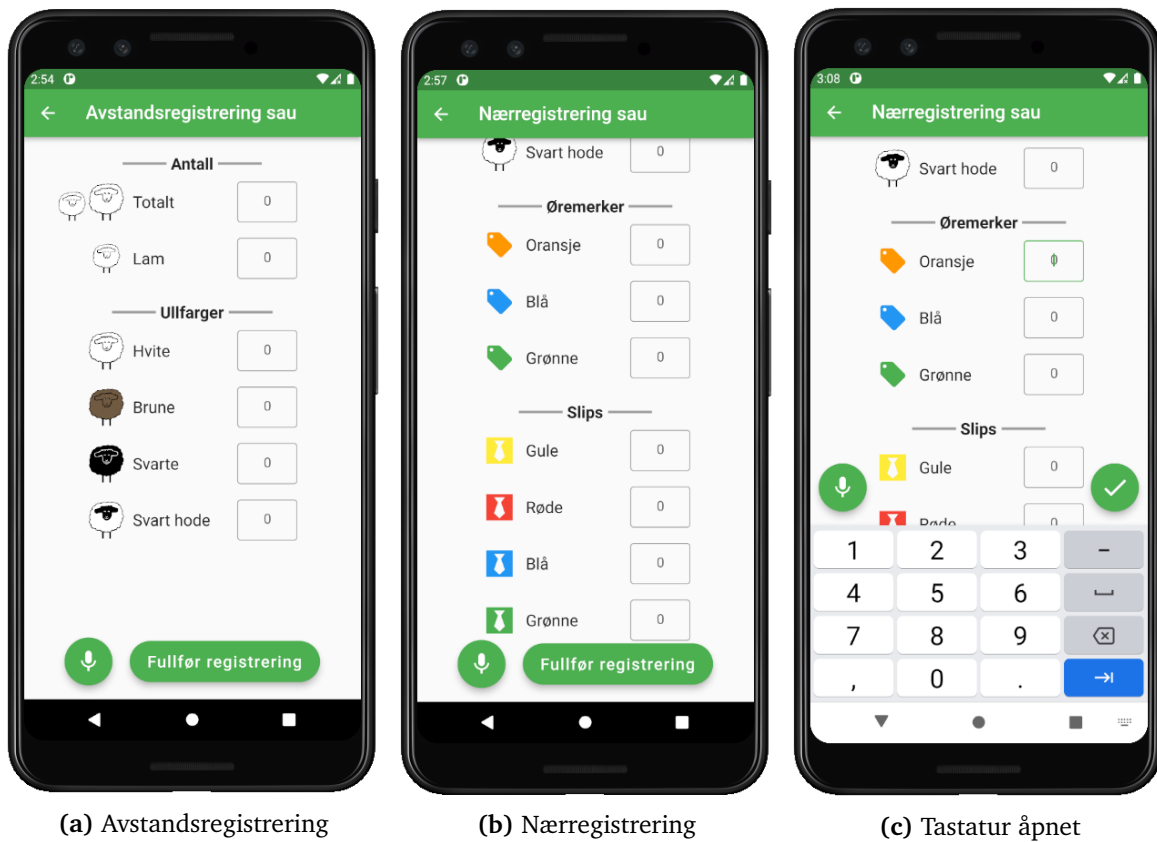
Figur 5.29 viser skjermene for registrering av sau, herunder avstandsregistrering og nærregistrering. Applikasjonen beregner avstanden mellom enheten og sau(en)e som skal registreres, og velger nær- eller avstandsregistrering deretter. På avstand kan det være utfordrende å se detaljer som øremerker og slips, og derfor inneholder skjemaet for avstandsregistrering kun totalt antall sauer og lam, antall lam og antall av hver ullfarge. Skjemaet for nærregistrering inneholder, i tillegg til feltene fra avstandsregistrering, antall av de ulike øremerke- og slipsfargene som er definert av sauebonden i webapplikasjonen (som beskrevet i seksjon 5.2.2). Dersom sauebonden ikke har definert disse enda, brukes alle øremerkefarger, samt NSG sin offisielle anbefaling til slips¹.

Feltkombinasjonen ”Totalt” og ”Lam” er valgt fremfor ”Voksne” og ”Lam” av hensyn til effektivitet. Med den valgte kombinasjonen utfører oppsynspersonen én grovtelling og én fintelling. Med grovtelling menes at man teller *alle* sauene man ser, og med fintelling menes at man skiller mellom voksen sau og lam under tellingen. Vi mener at en grovtelling går raskere enn en fintelling, og at det ikke alltid er enkelt å skille voksen sau fra lam (spesielt på avstand). Det er på dette grunnlaget vi gikk bort fra feltkombinasjonen ”Voksne” og ”Lam” da den ville ha krevd to fintellinger fremfor én.

Det er to måter å fylle inn feltene på. Den første måten er med tastatur, som vist i figur 5.29c. I stedet for å trykke på ett og ett felt, kan oppsynspersonen heller trykke på den blå ”neste”-knappen nede til høyre på tastaturet for å aktivere neste felt, noe som forenkler og effektiviserer denne prosessen noe. Applikasjonen scroller automatisk etterhvert som man jobber seg nedover feltene.

Den andre måten å legge inn data på er gjennom taleregistrerings-dialogen, som startes ved å trykke på knappen med et mikrofon-ikon. Da vil applikasjonen muntlig stille spørsmål i rekkefølgen feltene står i. Applikasjonen scroller automatisk etter hvert som dialogen stiller spørsmål og fyller tilhørende felter med svarene. Det samme kjente brukergrensesnittet for sauregistrering benyttes uavhengig om man bruker tastatur eller taledialog. En detaljert beskrivelse av taleregistreringen gis i seksjon 5.3.7.

¹<https://www.nsg.no/beitebruk/utmarksbeite/merking-av-smafe/bjelleslips/>, besøkt 08.04.2022



Figur 5.29: Mobilapp: Registrering av sau



Figur 5.30: Mobilapp: Eksempler på validering av saueregistrering

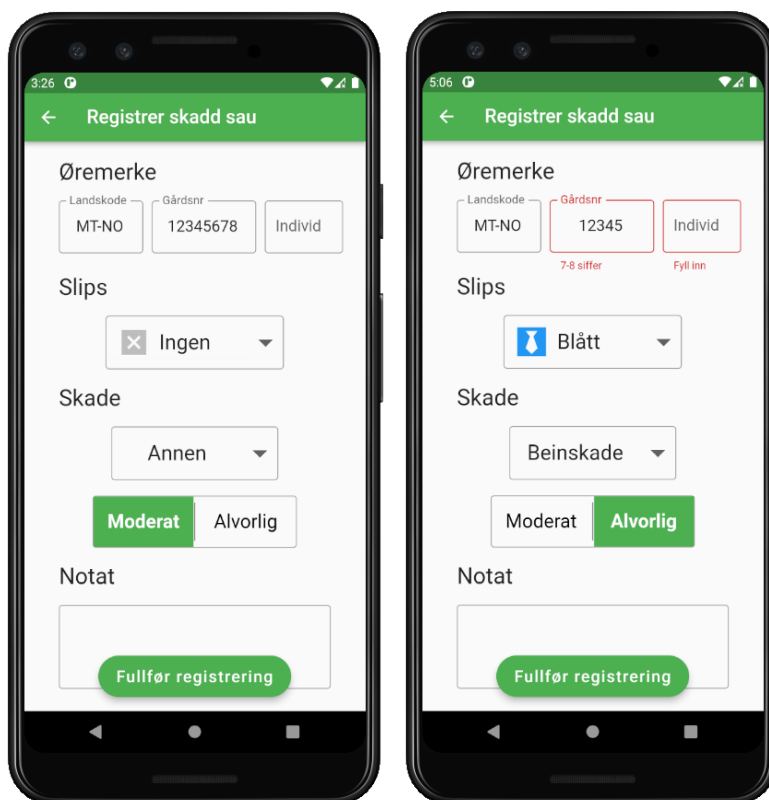
For at en registrering skal kunne fullføres må innflylt data være gyldig. Det er valgt å gå for en myk restriksjon fremfor en hard. Med myk restriksjon menes at registreringen er ugyldig dersom oppsynspersonen har telt for få sauer sammenlignet med de andre tellingene. Registreringen vil dermed være ugyldig dersom (1) det er flere lam enn totalt antall sauer og lam, (2) summen av ullfarger er høyere enn totalt antall sauer og lam, (3) summen av øremerkefarger er høyere enn totalt antall sauer og lam, eller (4) summen av slipsfarger er høyere enn totalt antall sauer og lam. Hard validering ville i tillegg ha inkludert å kontrollere at det er like mange øremerker som det er sauer, og at summen av de ulike slipsfargene ikke er høyere enn antall voksne sauer. Man må regne med at oppsynspersonen ikke får med seg absolutt alle øremerker og slips ved hver eneste registrering. Det er nemlig utfordrende å få med seg slike detaljer både på avstand, i ulike værforhold og grunnet sauenes bevegelse eller mangel på denne. Derfor hadde en hard restriksjon vært et lite brukervennlig valg. Hvordan skjermen ser ut for de to første restriksjonene vises i figur 5.30. Valideringsmelding gis i en rød boks som er fast plassert over de resterende elementene på skjermen, slik at den alltid er synlig selv om man scroller nedover. De aktuelle feltene som valideringsmeldingen gjelder fremheves med den samme rødfargen. Valideringen utføres i sanntid for input med tastatur, og etter endt dialog for input med tale.

Registrering av skadd sau

Figur 5.31 viser skjermen for å registrere skade; figur 5.31a før utfylling og figur 5.31b med valideringsfeil. Øremerke-input er delt opp i tre felter fordi antall siffer i gårdsnummer og individnummer kan variere fra gård til gård. For å effektivisere registreringen er landskode og gårdsnummer forhåndsutfyllt med den norske "MT-NO" og gårdens eget nummer, med mulighet for å endre disse siden man kan møte på sauer som tilhører andre gårder, og utenlandsk sau i grenseområder. I tillegg til øremerke registres slipsfarge (evt. ingen slips), type skade og om skaden er moderat eller alvorlig. Dersom oppsynspersonen mener det er hensiktsmessig å skrive et notat kan dette gjøres nederst på skjermen.

Registrering av sauekadaver

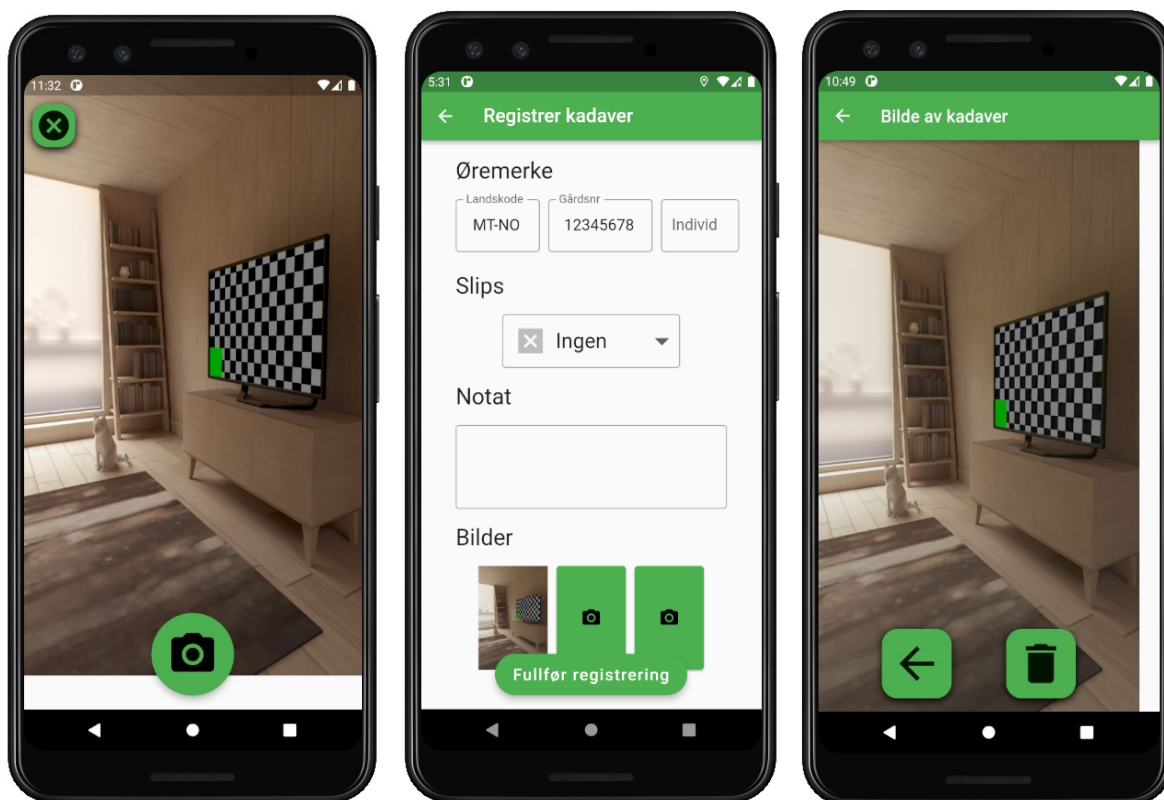
Figur 5.32b viser skjermen for registrering av sauekadaver etter at et bilde har blitt tatt. Øremerke-, slips- og notat-input fungerer på samme måte som i registrerings-skjemaet for skadd sau. Ved å trykke på en av de tre knappene med kamera-ikon kan man ta bilde, som vist i figur 5.32a. Man kan avbryte fotografering med knappen i øvre venstre hjørne, eller ta bilde med den sentrerte knappen nederst på skjermen. Når et bilde er tatt går applikasjonen tilbake til registrerings-skjermen, og bildeknappen er nå erstattet av bildet som ble tatt, som vist i figur 5.32b. Hvis man ønsker å vurdere bildet kan man trykke på det, og skjermen i figur 5.32c vises. Her er det mulig å slette bildet ved å trykke på søppelspann-knappen.



(a) Før utfylling

(b) Valideringsfeil

Figur 5.31: Mobilapp: Registrer skadd sau

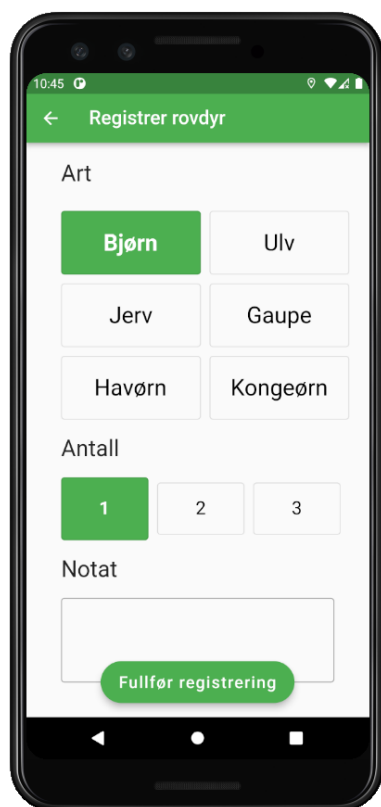


(a) Ta bilde

(b) Bilde tatt

(c) Vurder bilde

Figur 5.32: Mobilapp: Registrering av sauekadaver



Figur 5.33: Mobilapp: Registrering av rovdyr



Figur 5.34: Mobilapp: Registrering av notat

Registrering av rovdyr

Figur 5.33 viser skjemaet for å registrere rovdyr. Input består av radio-knapper for art og antall, i tillegg til et notat-felt.

Registrering av notat

Skjermen for registrering av notat vises i figur 5.34, og inneholder kun et større tekstfelt. Tekstfeltets høyde gjør at man ser hele feltet selv om tastaturet er åpent. Hensikten med notatregistreringen er å dekke alt oppsynspersonen ønsker å dokumentere, som ikke går under de andre registreringstypene. Noen eksempler er at et gjerde er ødelagt, og at et stort tre har falt over en sti.

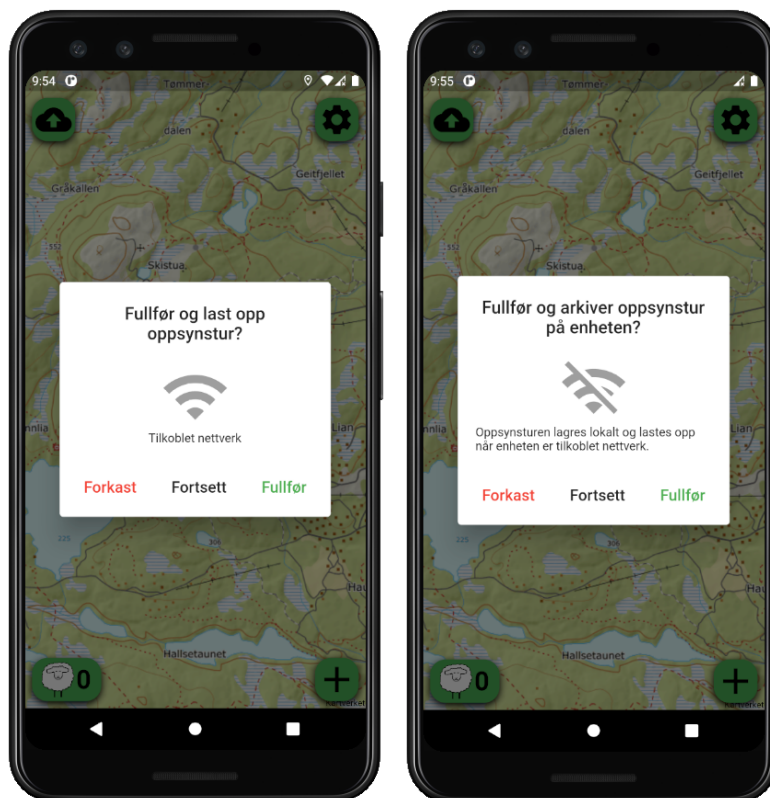
5.3.7 Registrering med tale

Dialogbasert taleregistrering er implementert for saueregistrering fordi det er for denne registreringstypen det kan være mest utfordrende å beholde oversiktsbildet (fordi man må telle antall sau, lam, øremerkefarger og slipsfarger imens saue-ene beveger seg). Dialogspørsmålene er forhåndsdefinert, og applikasjonen velger

spørsmål basert på om registrering er på nært hold eller på avstand, og basert på hvilke øremerke- og slipsfarger som er definert for gården. Applikasjonen stiller muntlige spørsmål i input-feltenes rekkefølge, og begynner med ”Hvor mange sauer og lam?”. Når applikasjonen er klar for å ta imot tale-input, signaliserer den dette med en lyd. Etter lydsignalet svarer man på spørsmålet med et tall på engelsk. Hvis innstillingen ”Les tilbake” er skrudd på, vil enheten si det som ble tolket før den stiller neste spørsmål. Dersom svaret ble feiltolket og/eller man ønsker å endre svaret, kan man bruke talekommandoene ”back” eller ”previous” istedenfor å svare på det neste spørsmålet som blir stilt, og applikasjonen stiller forrige spørsmål på nytt. Disse talekommandoene kan gjentas mange ganger etter hverandre. Dersom applikasjonen ikke forstod hva som ble sagt (dvs. at tolkningen ikke er et tall), vil den si ”Jeg forstod ikke”, og deretter stille spørsmålet på nytt. Dersom oppsynspersonen ikke sier noe, vil enheten (hvertfall Android-enheter) i løpet av få sekunder slutte å lytte etter svar og signalisere dette med et eget lydsignal, og dialogen settes på pause. Når man er klar til å fortsette, trykker man på mikrofonikonet, og dialogen gjenopptas på spørsmålet den ikke fikk svar på. Dette er nyttig i tilfeller der oppsynspersonen trenger tid til å finne svaret på spørsmålet. Dessverre er det på Android-enheter ikke mulig å øke lyttetiden når mikrofonen ikke oppfatter noe tale. Vi anser derfor denne løsningen, der dialogen blir satt på pause og kan gjenopptas, som den beste. Under dialogen vil applikasjonen automatisk ta seg av scrolling nedover på registreringssiden ettersom spørsmålene stilles, og all input valideres når registreringen fullføres. All denne funksjonaliteten har blitt brukertestet, hvor godt den fungerer beskrives i seksjon 8.2.

5.3.8 Opplasting av oppsynstur

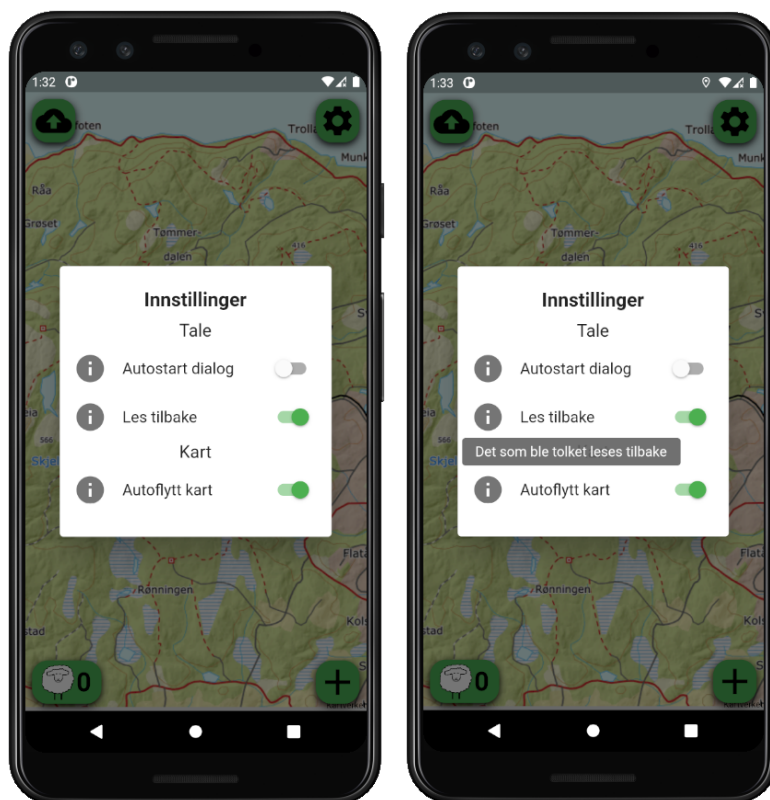
Når man er ferdig med en oppsynstur trykker man på knappen i øvre venstre hjørne av hovedskjermen. Knappen åpner én av to mulige popup-varianter; den ene dersom enheten har nettverksforbindelse, den andre om enheten ikke har nettverksforbindelse, se figur 5.35. Popup-vinduet fungerer som en sikring mot ikke-tiltenkte trykk ved at det dobbeltsjekker om oppsynspersonen faktisk vil fullføre og laste opp turen. Dette bekreftes ved å trykke på alternativet ”Fullfør”. På denne måten vil oppsynsturen ikke avsluttes ved uhell. Dersom man ønsker å fortsette oppsynsturen trykker man på ”Fortsett”, utenfor popup-vinduet eller bruker enhetens tilbake-knapp. Man har i tillegg muligheten til å avbryte oppsynsturen gjennom alternativet ”Forkast”.



(a) Med nettverkforbindelse

(b) Uten nettverksforbindelse

Figur 5.35: Mobilapp: Opplasting av oppsynstur



(a) Innstillinger

(b) Aktivt tooltip

Figur 5.36: Mobilapp: Innstillinger

5.3.9 Innstillinger

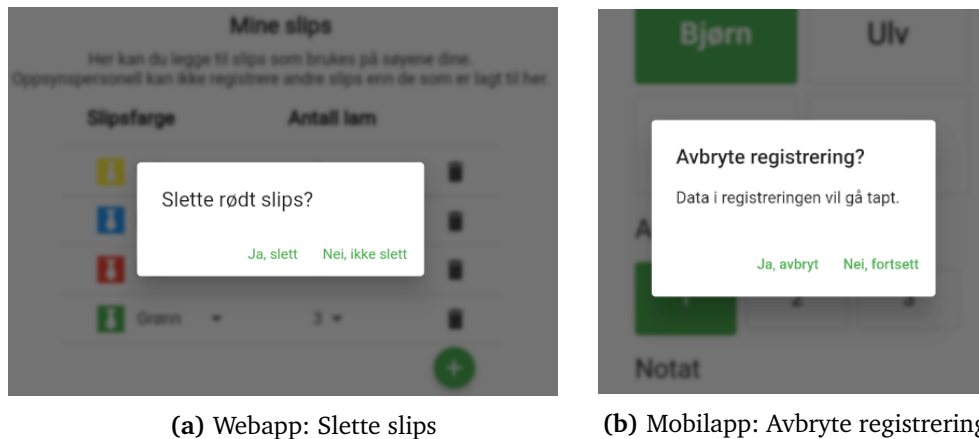
Figur 5.36 viser applikasjonens innstillinger, som inkluderer taleinnstillinger og en innstilling for kart. Hver innstilling har et info-ikon som gir en kort forklaring av innstillingen, som vist i figur 5.36b. Dersom innstillingen "Autostart dialog" er skrudd på, vil taleregistreringsdialogen starte automatisk når saueregistreringer påbegynnes. Om den derimot er skrudd av, kan man starte dialogen ved å trykke på knappen med mikrofon-ikon. Dersom innstillingen "Les tilbake" er aktivert, vil appen gjengi det den tolket før neste dialog-spørsmål stilles. Dette gir en ekstra forsikring om at registreringen ble korrekt uten at man trenger å se på skjermen, og feiltolkninger kan oppdages og rettes på ved bruk av talekommandoer. I tilfeller der enheten ikke støtter taleregistrering uten nettverksforbindelse, vil mikrofon-knappen skjules, og det gis beskjed om dette under taleinnstillinger med forklaring av hvordan enheten kan konfigureres riktig. Taleregistreringsfunksjonaliteten forklares nærmere i seksjon 5.3.7, og implementasjonen i seksjon 7.4. Hvis kartinnstillingen "Autoflytt kart" er aktivert, vil kartet sentrere seg på enhetens posisjon hver gang denne hentes (hvert 15. sekund). På den måten slipper oppsynspersonen å flytte kartet etter seg, og kan heller holde blikket opp. Det er likevel mulig at noen foretrekker å flytte kartet selv, og det er derfor naturlig å ha dette som en innstilling.

5.4 Designvalg

Det er lagt mye tid og fokus i systemets programflyt og grafiske utforming. Dette er gjort fordi det er avgjørende for brukeropplevelsen at systemet er så behagelig som mulig å bruke. Vi har forsøkt å utvikle en intuitiv programflyt. Derfor vil de viktigste designvalgene fremheves i denne seksjonen.

5.4.1 Felles

Det har vært fokus på brukervennlighet på tvers av systemet. Der det er hensiktsmessig gir de ulike sidene på webapplikasjonen beskrivelser som forklarer hva som kan gjøres, hvordan det gjøres og hvordan det henger sammen med mobilapplikasjonen. Videre fokuserer begge applikasjonene på å dobbeltsjekke at brukeren faktisk ønsker å utføre handlinger som ikke kan angres, eksempelvis ved sletting av beiteområde, øremerke, slips eller oppsynspersonell i webapplikasjonens defineringssider, og ved avbryting av registrering eller ved avslutning av oppsynstur i mobilapplikasjonen (se figur 5.37). I tillegg gis det tilbakemelding dersom noe er feil, der feilen utheves med visuelle effekter og forklares i tekst, som vist i enkelte figurer tidligere i kapittelet.



Figur 5.37: Eksempler på popup-vinduer for å bekrefte handling



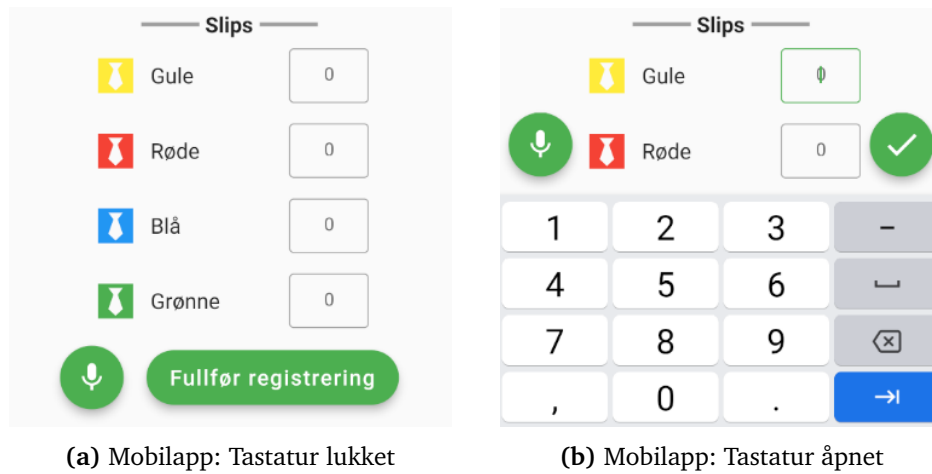
Figur 5.38: Webapp: Spinning circle under generering av årsrapport

For å sikre et stilrent og standardisert utseende av applikasjonene er Material Design² benyttet. Temafargen grønn er valgt for å gi assosiasjoner til naturen, som er miljøet applikasjonen skal brukes i. Knapper er grønne med hvit tekst, valgt fane har grønn understrek, valgt menyvalg på ”Min side” er markert med grønn, spinning circle er grønn, input-felter er grønne når de har fokus.

Bryter-knappene har, på tvers av applikasjonene, bakgrunnsfarge og fet tekst for å fremheve alternativet som er valgt. Fet tekst på grønn bakgrunnsfarge brukes for valgt alternativ, og ikke-fet tekst med hvit eller grå bakgrunnsfarge brukes for alternativer som ikke er valgt. Fargeskiftet gjøres for å tydeliggjøre endring, og fet tekst brukes for å utheve det valgte alternativet.

Imens en side lastes inn eller har prosesser i bakgrunnen, gis brukeren tilbakemelding om dette gjennom en animert spinning circle med tilhørende tekst, som vist i figur 5.38 under generering av årsrapport.

²<https://material.io/>



(a) Mobilapp: Tastatur lukket

(b) Mobilapp: Tastatur åpnet

Figur 5.39: Mobilapp: Plassering av knapper på registrerings skjermer

5.4.2 Webapplikasjon

Det er alltid en synlig scrollbar på webapplikasjonen dersom innholdet ikke får plass på skjermen. Denne vises for eksempel på siden for å definere beiteområde om flere beiteområder er definert, og på siden for å vise oppsynsturer dersom det er registrert saueskade, kadaver, rovdyr eller notat.

5.4.3 Mobilapplikasjon

Plassering av knapper

På skjermen for oppsynsturer er knappene på kartet plassert i hvert av skjermens hjørner. Dette er av to årsaker, både at de skal ta så lite fokus som mulig og at et så stort som mulig område i sentrum av skjermen skal være åpent for å gjøre registreringer. Knappen for ny registrering er plassert nede til høyre i skjermen da det gjør den mest tilgjengelig og det er denne som vil bli brukt oftest. Tilsvarende er knappen for å avslutte oppsynstur plassert øverst til venstre da dette gjør den minst tilgjengelig og det er denne knappen som skal benyttes færrest ganger i løpet av en oppsynstur.

Alle registrerings skjermene tar hensyn til at tastaturet dekker en stor del av skjermen. Med lukket tastatur er knappen for å fullføre registrering (og mikrofon-knappen ved sauregistrering) plassert nederst på skjermen, som vist i figur 5.39a. Med åpent tastatur flyttes den (og mikrofon-knappen ved sauregistrering) ovenfor tastaturet og ut til siden(e) for å holde så mange felter som mulig synlige, som vist i figur 5.39b. Knappen går i tillegg fra å være en stor knapp med tekst til å være en mindre og sirkelformet knapp med en hake i når tastaturet åpnes.

Effektivitetshensyn

Felles for registreringsskjemaene i mobilapplikasjonen er at de har fokus på effektivitet. Alle skjemaene har nedtrekksmenyer og store bryter-knapper fremfor tekstfelt der det lar seg gjøre, samt store overskrifter, for å gjøre registreringen så effektiv som mulig. I skjemaet for registrering av sau er det ikke nødvendig å fylle inn de tellingene det er null av, dette er signalisert ved å ha "0" som labelText. LabelText er valgt fremfor faktisk tekst fordi brukeren da vil slippe å fjerne "0" før et tall legges inn. I skjemaene for registrering av skade og kadaver kan det argumenteres for at slipsvalg hadde vært noe mer effektivt med radioknapper istedenfor nedtrekksmeny. Likevel veide hensynet til oversiktighet tyngre fordi de aktuelle registreringsskjemaene har en del input-widgets (det hadde blitt opptil syv radioknapper for å dekke alle syv slipsfarger). Øremerkets landskode og gårdsnummer er forhåndsutfylt for å øke effektiviteten. Videre ble det for registreringsskjermene valgt å bruke en flytende fullfør-knapp, som gjør at man slipper å lukke tastaturet for å fullføre registreringen.

Avvik fra felles designvalg

De aller fleste knapper er som nevnt grønne med hvit tekst eller ikon, men det finnes noen få avvik fra dette i mobilapplikasjonen. På hovedskjermen er ikonene bevisst sorte (fremfor hvite), fordi dette gir høyere kontrast mot kartet. Fargen på ikonene i forbindelse med fotografering er også sort av samme årsak, siden det også her vil være mange farger involvert.

Kapittel 6

Utviklingsprosess

Dette kapittelet presenterer prosessen i form av utviklingsmetodikk og tilhørende sermonier, samt prototyping av design.

6.1 Utviklingsmetodikk

Utviklingsmetodikken som er valgt for dette prosjektet er smidig og bruker en del elementer fra Scrum. Metodikken legger vekt på de fire grunnprinsippene i smidig utvikling [12]; ”Inkrementell design og iterativ utvikling”, ”Inspiser og tilpass”, ”Arbeid i samarbeid” og ”Kontinuerlig kundeinvolvering”. Teamet består av to studenter med kompetansen beskrevet i seksjon 1.6. Siden prosjektet er en masteroppgave, som innebærer en satt tidsramme, er enkelte deler av prosessen planbasert, slik som at design av prototyper ble gjort før prosjektperioden. Videre følger en nærmere beskrivelse av sermoniene som ble inkludert i metodikken og hvordan de ble brukt.

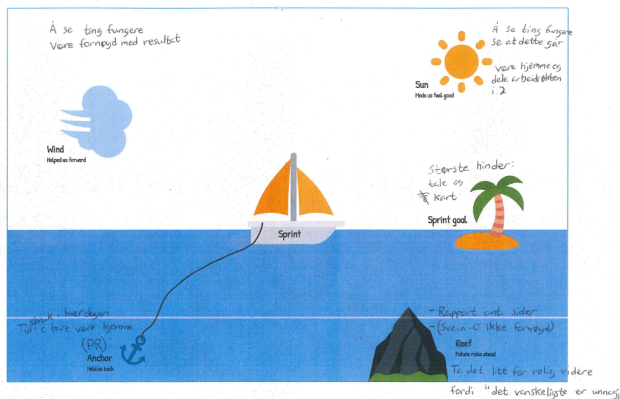
6.1.1 Daily Standup/Morgenmøte

Hver morgen ble det holdt et kort møte der man fortalte hva man har gjort og hva man skal gjøre. På slutten av møtene ble det lagt opp til å ta opp ufordringer eller temaer det var behov for å diskutere. Disse møtene tok vanligvis ikke mer enn fem minutter, men kunne strekke seg til en time ved lengre diskusjoner. Ved å holde et slikt møte hver morgen sørget vi for at hele teamet var oppdatert på hva som foregikk med prosjektet og dets fremgang.

6.1.2 Sprint retrospective/Evalueringsmøte

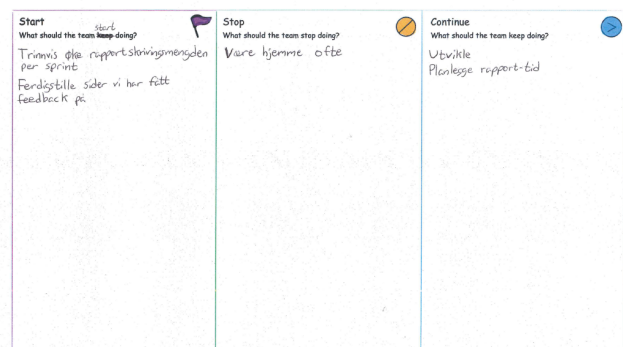
Etter hver sprint hadde vi et sprint retrospective møte. Målet med dette møtet var å evaluere sprinten i fellesskap slik at erfaringer kunne påvirke neste sprint til det bedre. I begynnelsen av prosjektperioden prøvde vi forskjellige formater for å overføre tankene våre til papiret, som for eksempel seilbåtmetoden vist i figur 6.1a. Til slutt endte vi opp med å bruke start-stop-continue, vist i figur 6.1b,

Sprint



(a) Seilbåtmetoden

Start / Stop / Continue



(b) A Start-Stop-Continue

Figur 6.1: Eksempler på sprint retrospective

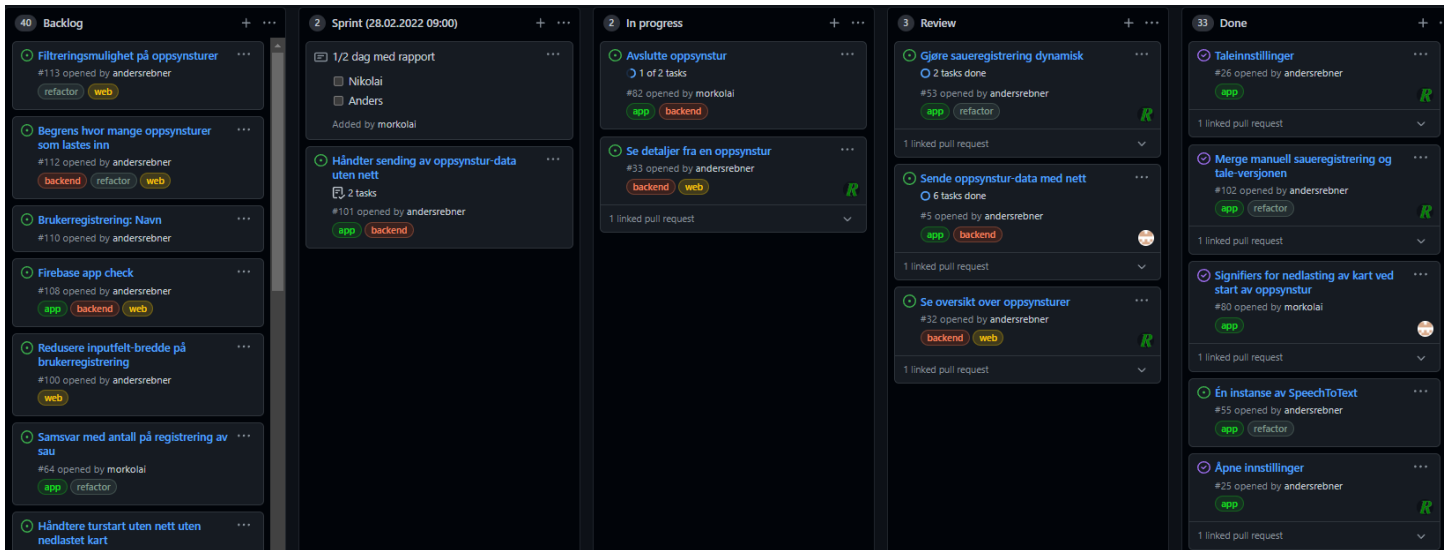
da denne metoden er enkel og konsis. I start-stop-continue skriver man ned hva teamet bør begynne med, slutte med og fortsette med i påfølgende sprint. Vi fylte inn skjemaene individuelt før vi i fellesskap diskuterte punktene vi hadde notert og kom frem til enkelte punkter vi skulle fokusere på. For å få inn et teambyggende mellommenneskelig element tegnet vi et hjerte på baksiden av skjemaet. Tanken var at man kunne skrive ned det man hadde på hjertet som nødvendigvis ikke passet inn i start, stop eller continue.

6.1.3 Sprint planning/Planleggingsmøte

Hver sprint ble planlagt i et sprint planning-møte i begynnelsen av hver sprint. I møtet ble det valgt oppgaver fra "Backlog"-kolonnen på utviklingstavlen vist i figur 6.2. Hver av de valgte oppgavene ble diskutert og estimert i antall dagsverk. Oppgaver som ville bidra til å oppfylle kravspesifikasjonen (kapittel 3) ble prioritert. Om det fortsatt var tid til overs i sprinten, etter at oppgavene i den var fullført, ville den bli supplert med oppgaver. Hvis den resterende tiden ikke var tilstrekkelig til en av de prioriterte oppgavene ville en mindre viktig oppgave bli valgt for å fylle ut tiden.

6.1.4 Kundemøte

Siden veileder også fungerte som produkteier og satt på domenekunnskap, ble veiledningsmøter i praksis brukt som kundemøter. Spesielt i begynnelsen av utviklingsperioden da domenet var ukjent for oss og det var mye som måtte kartlegges. I disse møtene var det spesielt viktig fra vår side å komme forberedt og stille gode spørsmål slik at vi fikk oversikt over domenet så raskt som mulig. Det var gjennom disse møtene kravspesifikasjonen ble utarbeidet. Utformingen av kravspesifikasjonen foregikk på en smidig måte ved at den var under utvikling gjennom hele



Figur 6.2: Utviklingstavle

prosjektperioden, fordi vi stadig avdekket nye krav og behov.

6.1.5 Prosjekttavle og GitHub

GitHub ble brukt for å administrere prosjektet, noe som var en fordel da både prosjektstyringsverktøy og kodebase er på samme plattform. Prosjekttavlen, vist i figur 6.2, delte vi inn i fem kolonner. Første kolonne er "Backlog" hvor fremtidige oppgaver ligger. Andre kolonne er "Sprint" hvor oppgavene i inneværende arbeidsperiode er oppført, og perioden varer til datoen som er skrevet i kolonnenavnet. Tredje kolonne er "In progress" hvor oppgaver som løses er plassert. Fjerde kolonne er "Review" hvor oppgaver som er til vurdering er plassert. Siste kolonne er "Done" hvor fullførte oppgaver er oppført. Tavlen er delvis automatisert og hver oppgave er et *issue*. Når det opprettes et nytt *issue* tilknyttet prosjektet vil det automatisk legges i "Backlog", og når en *pull request* knyttet til et *issue* lukkes flyttes *issue*et til "Done".

6.1.6 Code review

For å sikre kvalitet i kodebasen evaluerte vi hverandres kode. Dette var en påkrevd handling før koden kunne merges inn i main-branchen. I tillegg til å øke kvaliteten ga en slik evalueringsprosess oss bedre oversikt over kodebasen. Med et team som består av kun to personer vil en code review ta opp halvparten av teamets kapasitet. Samtidig vil den gi avkasting i form av at hele teamet får innblikk i kode de ikke har skrevet selv.

6.2 Prototyping av Design

For å konkretisere konseptet og egne idéer bestemte vi oss for å prototype webapplikasjonens og mobilapplikasjonens design ved prosjektets oppstart. Designene skulle ikke være endelig, men var heller ment å gi grunnlag for videre utvikling. Vi var klar over at det, med en smidig tankegang, var tidlig å prototype på daværende tidspunkt grunnet faren ved å låse seg til design og programflyt, men det var spesielt to årsaker til at vi likevel gjorde det. Første årsak var at fagområdet oppgaven befinner seg i var ukjent for oss begge. Ved å prototype ville vi utarbeide mer konkrete spørsmål, og med svar på disse ville vi oppnå bedre forståelse. Den andre årsaken er at våre erfaringer fra tidligere prosjekter tilsier at man gjerne har ulike mentale modeller av hvordan systemet skal bli, og at prototyping kan føre til bedre overensstemmelse mellom disse.

Vi satt derfor i gang med å tegne lo-fi prototyper på papir hver for oss før vi i fellesskap diskuterte dem og til slutt endte opp med prototypene vist i vedlegg B for webapplikasjonen og vedlegg C for mobilapplikasjonen.

Kapittel 7

Implementasjon

I dette kapitlet presenteres implementasjonen av systemet. Kapitlet redegjør ikke for implementasjonen i sin helhet, men går heller i dybden på de delene vi anser som viktige eller interessante. Tabell 7.1 viser antall linjer kildekode og testkode som vi har skrevet, fordelt på webapplikasjonen og mobilapplikasjonen. Antallene som vises ekskluderer tomme linjer. All kode er tilgjengelig på GitHub under linken: <https://github.com/rebner-mork/master-thesis>, og webapplikasjonen er i skrivende stund tilgjengelig under linken: <https://master-backend-93896.web.app/>.

	Linjer kildekode	Linjer testkode
Web	5 237	481
App	6 214	755

Tabell 7.1: Antall kodelinjer, beregnet i Powershell

7.1 Valgte teknologier og biblioteker

Denne seksjonen gir en grov beskrivelse av de viktigste teknologiene og bibliotekene som brukes i implementasjonen. En oversikt over alternativene som ble vurdert gis i seksjon 9.1.

7.1.1 Flutter & Dart

Systemet er utviklet i Dart¹ med rammeverket Flutter. Dart er et Google-støttet programmeringsspråk med åpen kildekode som er optimalisert for raske og plattformuavhengige applikasjoner. Flutter ble lansert av Google i 2018, og er et kryssplattform rammeverk som brukes for å utvikle applikasjoner for mobil, web og desktop fra én enkelt kodebase. Dette betyr at man eksempelvis kan utvikle en applikasjon for både Android og iOS med samme kildekode, istedenfor å utvikle én

¹<https://dart.dev/>

versjon i Kotlin/Java for Android og en annen versjon i Swift/Objective-C for iOS. Flutter gjør dette ved å native-kompilere kodebasen, som betyr at høynivå-språket Dart kompileres til det plattformspesifikke språket til enheten applikasjonen kjøres på. Dette gjør at Flutter-applikasjoner, i motsetning til applikasjoner utviklet med andre kryss-plattform rammeverk, ikke har særlig redusert ytelse. [13]

7.1.2 Firebase

Firebase er en skyplattform levert av Google. Den tilbyr en rekke tjenester som brukerautentisering, database, skylagring, push-varsling osv. Tjenestene hostes og vedlikeholdes av Google, og man kan enkelt og raskt skalere applikasjonen sin i takt med økende ressursbehov. Man betaler kun basert på mengden ressurser man bruker utover det som er gratis (antall autentiseringer og database-lesinger, lagret datamengde osv.). Med andre ord slipper man å utvikle en egen backend for standard funksjonalitet. [11]

Videre gis en kort beskrivelse av de Firebase-tjenestene som brukes av vårt system.

Firebase Authentication

Firebase Authentication² er brukt for å autentisere brukere med e-post og passord. Denne støtter i tillegg funksjonalitet som multi-faktor autentisering, nullstilling av passord og innlogging gjennom SMS eller Google/Apple/Facebook osv.

Cloud Firestore

Cloud Firestore³ er en rask NoSQL dokumentdatabase. Hvordan denne fungerer og hvordan vi har designet den til vårt bruk er beskrevet i seksjon 4.3.

Cloud Storage

Cloud Storage⁴ er en skylagringsløsning. Siden bilder kan overskride den maksimale dokumentstørrelsen i Cloud Firestore (1MB), bruker vi Cloud Storage til å lagre bilder som tas av kadaver da denne tjenesten er beregnet for lagring av større filer. Når bilder blir tatt i mobilapplikasjonen lagres de først på enheten. Når de lastes opp til Cloud Storage, lagres det en referanse til hver av disse i kadaverdokumentet i Cloud Firestore. Når bildene er lastet opp blir de slettet fra enheten. Idet de skal hentes ut brukes referansen til å hente en nedlastningslenke for hvert bilde.

²<https://firebase.google.com/docs/auth>

³<https://firebase.google.com/docs/firestore>

⁴<https://cloud.google.com/storage>

Firestore Hosting

Firestore Hosting⁵ tilbyr hosting av blant annet webapplikasjoner, og gjør at man med en enkelt-kommando i terminalvinduet kan utplassere en applikasjon til en global CDN (content delivery network). Vi bruker denne hostingen, som gjør at webapplikasjonen i skrivende stund kan nås direkte på `https://master-backend-93896.web.app/`.

7.1.3 Biblioteker

Kart

”flutter_map” er en implementasjon av kartklienten Leaflet i Dart som kan brukes i Flutter-applikasjoner⁶. Dette gir i tillegg til funksjoner som å se og manøvrere i kart muligheten til å tegne på det og finne lengde- og breddegrader til punkter. Biblioteket brukes i både webapplikasjon og mobilapplikasjon.

Talegjenkjenning

Talegjenkjenningsbiblioteket `Speech_to_text`⁷ er ment å brukes til kommandoer eller korte setninger, og brukes derfor i taledialogen ved sauregistrering. Biblioteket benytter enhetens underliggende talegjenkjenning, noe som betyr både at den fungerer uten nettverk dersom enheten er konfigurert for dette, og at den fungerer på tvers av plattformer.

PDF-behandling

Generering av årsrapport gjøres med Dart-biblioteket `pdf`⁸. Biblioteket er todelt, der én av delene har et system som er tilnærmet likt som Flutter sitt Widget-system. Dette forenkler genereringsprosessen en god del fordi man i stor grad kan behandle hver rapportside som en statisk Flutter-skjerm.

Andre biblioteker

Bruken av de resterende bibliotekene er enkel og nevnes derfor kort her. Biblioteket:

- **flutter_tts** brukes til å stille dialog-spørsmål gjennom enhetens høyttaler⁹.
- **email_validator** brukes til å validere format på e-postadresser¹⁰.
- **location** brukes til å hente enhetens GPS-posisjon¹¹.

⁵<https://firebase.google.com/docs/hosting>

⁶https://pub.dev/packages/flutter_map

⁷https://pub.dev/packages/speech_to_text

⁸<https://pub.dev/packages/pdf>

⁹https://pub.dev/packages/flutter_tts

¹⁰https://pub.dev/packages/email_validator

¹¹<https://pub.dev/packages/location>

- **camera** brukes til å fotografere med enhetens kamera¹².
- **fluttericon** brukes til å vise ikoner av sau og slips¹³.
- **provider** brukes til håndtering av helhetlig applikasjonstilstand¹⁴.
- **path_provider** brukes til å hente filbanen til applikasjonens dokumentmappe på enheten.¹⁵

7.2 Grov oversikt over widget-hierarki

Flutter er inspirert av React, og brukergrensesnittet består dermed av ”Widgets”¹⁶. Widgets beskriver hvordan de selv skal se ut med den eventuelle tilstanden de måtte ha (det finnes stateful og stateless widgets). Når tilstanden til en widget endrer seg, vil den forrige widget-beskrivelsen sammenlignes med den nye for å finne de minimale endringene i widget-treet som skal til for å gjennomføre overgangen til en ny tilstand. Flutter har en rekke standard-widgets, eksempelvis Text, Row, Column, Container og Stack, men det finnes mange widgets innenfor kategorier som layout, input, animasjon, styling, interaksjon, bilder osv. Det er vanlig å lage widgets til eget bruk, der man benytter seg av en kombinasjon av disse.

Flutter har to routing-mekanismer som kan brukes alene, eller i kombinasjon, for å navigere fra en skjerm til en annen. Den første er en Router-widget og den andre er en Navigator-widget. Vår implementasjon benytter seg av Navigator, som pusher og popper de ulike skjermene (widgetene) inn og ut av stacken. Det er kun den øverste skjermen i stacken som vises, så når den poppes av stacken vil skjermen under vises.

Figur 7.1 viser en grov oversikt over webapplikasjonens widget-hierarki. Det er kun de sentrale topp-nivå widgetene som vises i figuren, disse omfavner flere nivåer med både widgets vi har utviklet selv og standard widgets. Figuren gjenspeiler Navigator-stacken som kun består av to widgets, der LoginOrSignUpPage er startsidene og MainPage pusher på stacken etter innlogging eller opprettelse av brukerkonto. Videre består MainPage av en TabBar og et TabBarView, der fanene inneholder widgetene ReportsPage, TripsPage og MyPage. ReportsPage-boksen er tom fordi denne widgeten kun består av standard widgets. TripsPage benytter en Row-widget for å dele seg i to kolonner. Første kolonne inneholder en widget kalt TripOverviewList, som er en liste med alle oppsynsturer. Den andre kolonnen inneholder en widget kalt DetailedTrip, som viser data fra en valgt oppsynstur, bl.a. ved hjelp av kart-widgeten MapOfTrip. MyPage bruker en NavigationRail-widget for å lage de fem menyvalgene som viser widgets for definering av gård, øremerker, slips, oppsynspersonell og beiteområder.

¹²<https://pub.dev/packages/camera>

¹³<https://pub.dev/packages/fluttericon>

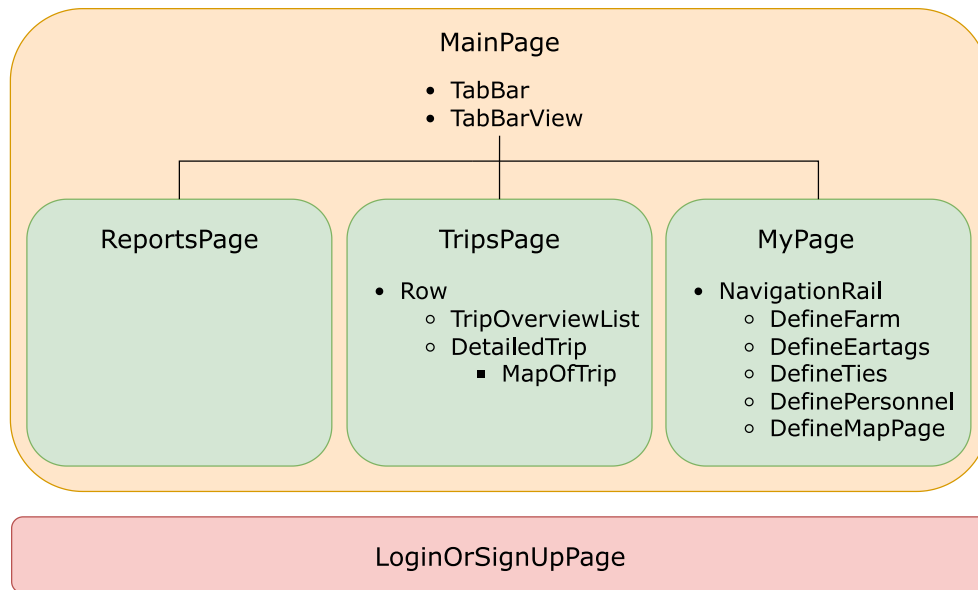
¹⁴<https://pub.dev/packages/provider>

¹⁵https://pub.dev/packages/path_provider

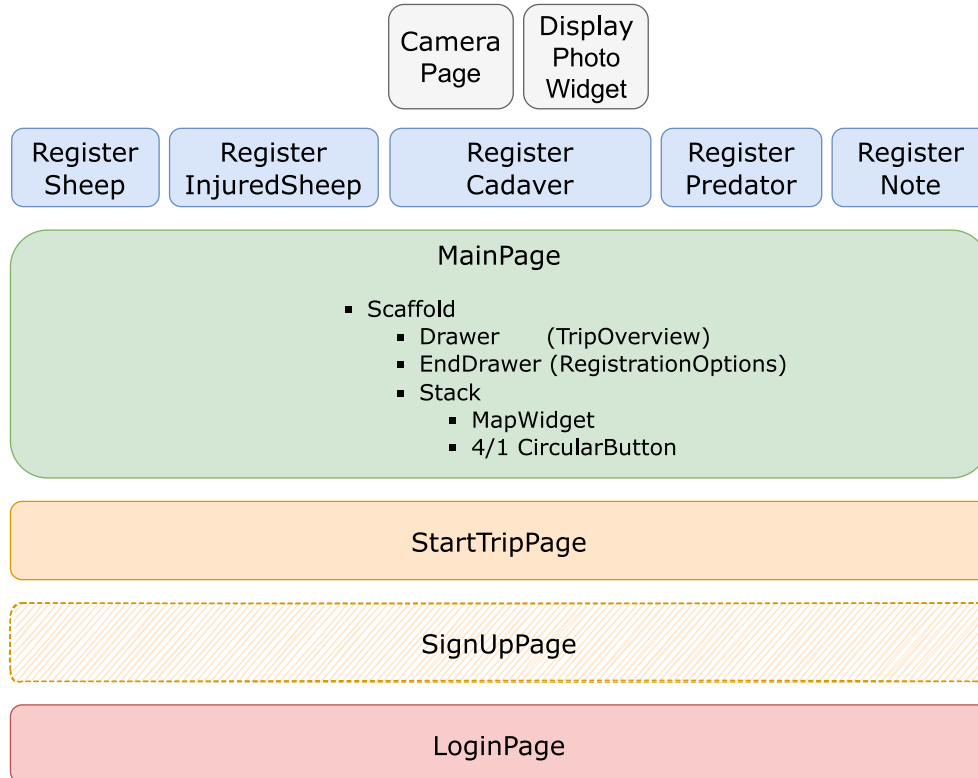
¹⁶<https://docs.flutter.dev/development/ui/widgets-intro>

Figur 7.2 gir en grov oversikt over mobilapplikasjonens widget-hierarki, på samme måte som figuren for webapplikasjonen. Navigator-stacken kan for mobilapplikasjonen bestå av opptil seks nivåer, der fotograferingsskjermen eller skjermen for bildevisning ligger på øverste nivå. Boksen rundt SignUpPage er skravert og har stiplede linjer fordi denne kun ligger i stacken dersom man oppretter brukerkonto. De blå boksene i figuren, som representerer registrerings skjermene, ligger på samme nivå fordi det kun er én av disse som kan ligge i stacken om gangen. MainPage pusher altså kun én av registrerings skjermene. Videre er det kun skjermen for registrering av kadaver som pusher én av de to mulige bilde-widgettene oppå seg.

MainPage består av en Scaffold-widget, som gir drawer-funksjonalitet. Med andre ord kan turoversikten (TripOverview) dras inn fra venstre side av skjermen og menyen med registreringsalternativer (RegistrationOptions) dras inn fra høyre side. Disse kan også åpnes ved bruk av knapper. Videre består MainPage av en Stack-widget, som gjør at widgets kan ligge foran og bak hverandre i skjermens dybde-retning. Kartwidgeten MapWidget ligger i bunnen og én eller fire hjørneknapper (CircularButton) ligger ovenpå. Stacken i figuren er noe forenklet da den ikke viser popup-vinduene for registreringsdetaljer, innstillinger eller avbrytelse av registrering, selv om disse riktignok er en del av stacken når de vises.



Figur 7.1: Webapp: Widget-hierarki



Figur 7.2: Mobilapp: Widget-hierarki

7.3 Kart

Kartklienten leser inn kartfilene fra en bestemt struktur som baserer seg på hvordan kartet er bygget opp. Kartet har en rekke zoomnivåer som øker detaljgraden. Hvert nivå er bygget opp av en rekke kvadratiske bildefiler kalt fliser. Hver flis har en oppløsning på 256*256 piksler. For høyere zoomnivåer er det økt detaljgrad som fører til at flere fliser vises i en skjerm. Antall fliser i et nivå øker eksponentielt og kan beregnes med 2^z , der z representerer zoomnivået og går fra 0 til feks. 18. På zoomnivå 0 er det én flis og hele kartet vises på denne, mens på zoomnivå 18 (høyeste detaljgrad) består kartet av 262 144 fliser som ligger i et rutenett. Flisene lagres i en mappestruktur hvor det på øverste nivå er én mappe per zoomnivå. I hvert zoomnivå kan man finne en spesifikk flis med en x - og y -verdi siden flisene bygger kartet som et rutenett. Under mappene for zoomnivå ligger det derfor en mappe for hver x -indeks som inneholder alle flisene for denne kolonnen, og hver flis i denne mappen er navngitt med y -indeksen den representer. Man kan med andre ord finne en spesifikk flis med verdier for z , x og y .

`flutter_map` brukes ofte i sammenheng med en WMS (Web Map Service) der de flisene som trengs fortløpende hentes fra nettet. Det var et krav at applikasjonen skal bruke Norgeskart sitt topografiske kart (se seksjon 3.2.1). Statkart har en åpen WMS som tilbyr dette via url-malen: `https://opencache{s}.statkart.no/gatekeeper/gk/gk.open_gmaps?layers=topo4&zoom={z}&x={x}&y={y}`. Et annet krav for applikasjonen var at kartet skal kunne brukes uten nettverkstilkobling (se seksjon 3.1.2), som betyr at kartet må lastes ned i forkant av oppsynsturen. `Flutter_map` sin dokumentasjon foreslår her to løsninger, enten å laste inn flisene og lagre dem i cache, eller å laste ned flisene og pakke dem med applikasjonen når den bygges. Verken den første eller den andre løsningen passer da man på forhånd ikke vet hvilke fliser brukeren ønsker og cache kun er ment for midlertidige filer. Derfor var det for mobilapplikasjonen best å lagre flisene i applikasjonens dokument-mappe. Etter nedlasting blir kartflisene liggende på enhetens filsystem helt til applikasjonen avinstalleres.

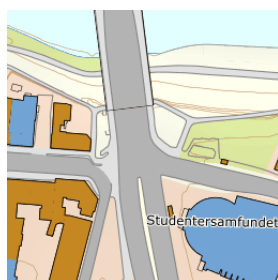
Kartutsnittet som skal lastes ned defineres av to punkter med lengde- og breddegrad, et for nordvestlig hjørne og et for sørøstlig hjørne. Disse to punktene danner en firkant som rammer inn alle flisene kartutsnittet er bygget opp av. For å laste ned disse flisene må lengde- og breddegradene regnes om til flisindekser. Matematikken bak omregningen baserer seg i korte trekk på at jorda er rund, men representeres som en flate i kartet. Her finnes det forskjellige projeksjoner med ulike fordeler og ulemper. Projeksjonen som er aktuell her er World Geodetic System 1984¹⁷ (WGS84). Formlene for å regne ut x - og y -indeksene er som følger:

¹⁷https://no.wikipedia.org/wiki/World_Geodetic_System

$$x = \left\lfloor \frac{lon + 180}{360} \cdot 2^z \right\rfloor$$

$$y = \left\lfloor \left(1 - \frac{\ln \left(\tan \left(lat \cdot \frac{\pi}{180} \right) + \frac{1}{\cos \left(lat \cdot \frac{\pi}{180} \right)} \right)}{\pi} \right) \cdot 2^{z-1} \right\rfloor$$

I ligningen representerer z zoomnivå, lon representerer lengdegrad og lat representerer breddegrad. Når disse indeksene er beregnet henter applikasjonen inn alle flisene via HTTP-kall og lagrer dem som bilder i applikasjonens dokumentmappe. Bildene blir navngitt og strukturert slik at de kan hentes tilbake og brukes i kartklienten. For eksempel vil enhetsspesifikk filbane til applikasjonens dokumentmappe etterfulgt av `/maps/17/69320/35430.png` gi flisen som vises i figur 7.3. Denne flisen representerer lengdegrad 63.422443 og breddegrad 10.395491 med indekser 69320,35430.



Figur 7.3: Kartflis på zoomnivå 17 som viser Studentersamfundet i Trondheim

Størrelsen på filene som lastes ned varierer da det er forskjellige mengder data i dem. En flis på lavt zoomnivå i et urbant strøk kan inneholde rikelig med former og farge og er derfor vanskelig å komprimere, mens en flis på høyeste zoomnivå av en vidde kan være tilnærmet ensfarget. Det er derfor ikke mulig å si nøyaktig hvor mye data et kartutsnitt på et visst areal vil kreve da det avhenger av området. Tabell 7.2 viser en oversikt over flisantallet og datamengden som kreves for å vise et område på 10km^2 for hvert zoomnivå. Området er et utsnitt av Bymarka i Trondheim som strekker seg fra 63.42971, 10.20203 i nordvest og til 63.41071, 10.27607 i sørøst. Som man ser av tabellen øker mengden data raskt for de høyeste zoomnivåene. De lave zoomnivåene forblir på én flis fordi det kreves høyt zoomnivå før det definerte området trenger flere fliser for å vises. For å bevare muligheten til å få både oversiktlig og detaljert bilde, og samtidig begrense datamengden (som direkte påvirker nedlastningstiden), er kartklienten i applikasjonen begrenset til zoomnivåene 12 til 17.

Zoomnivå	Antall fliser	Datamengde
11	1	45,5 kB
12	1	53,9 kB
13	4	191,2 kB
14	12	443,4 kB
15	40	1,0 MB
16	126	3,1 MB
17	476	7,5 MB
18	1760	34,5 MB
1-18	2418	47,2 MB
12-17	659	12,4 MB

Tabell 7.2: Eksempel på egenskaper ved kartdata for et område på 10km²

7.4 Taleregistrering

7.4.1 Konfigurasjon av talegjenkjenning på enheten

Mobilapplikasjonen bruker enhetens innebygde talegjenkjenning. Det betyr at brukeren står fritt til å velge mellom de leverandørene som enheten tilbyr. For en Samsung-telefon kan man eksempelvis velge mellom "Samsung voice input" og "Google Voice Typing". Da vi kun har hatt mulighet til å teste applikasjonen på Android-enheter, er det disse to vi har utforsket. I henhold til kavspesifikasjonen (seksjon 3.2.2) må talegjenkjenningen kunne brukes uten nettverksforbindelse, og av de to nevnte alternativene er det kun Google Voice Typing som støtter dette.

Fremgangsmåten for å oppnå fungerende talegjenkjenning uten nettverk varierer fra telefon til telefon. Noen telefoner støtter dette med fabrikkinnstillinger, andre må konfigureres. I grove trekk går man inn i telefonens innstillinger og sørger for at Google Voice Typing er aktivert og at engelsk (US) er i listen over språk, før man til slutt laster ned tilsvarende språkpakke gjennom innstillingene for Google Voice Typing. På noen telefoner må Samsung voice input deaktiveres, og på noen telefoner blir språkpakker automatisk nedlastet når språket legges til. Samsung-enheters talegjenkjenning støtter med andre ord ikke norsk språk uten nettverksforbindelse, fordi Google Voice Typing i skrivende stund ikke tilbyr nedlasting av norsk språkpakke. Derfor må taleinput være på engelsk. Dersom enheten ikke er konfigurert på riktig måte, vil knappen for å starte dialogbasert registrering skjules, og taleinnstillingene erstattes av en forklaring til hva som må gjøres for å få det til å fungere.

7.4.2 Forbedring av treffprosent

Vi oppdaget allerede under forprosjektet (se vedlegg A) at offline talegjenkjenning verken var god eller stabil. Siden dette er negativt i seg selv, og fordi det hadde blitt et stort irritasjonsmoment fordi man utfører mange sauregistreringer i løpet av en oppsynstur, ønsket vi å forbedre denne. Da det var mye som skulle utvikles og prosjektet var begrenset i tid, ønsket vi først å se om det var mulig å oppnå en god nok forbedring uten å ta i bruk maskinlæring. Løsningen vi valgte var å lage et justeringsfilter som baserer seg på å rette feiltolkninger, som vist i tabell 7.3. Alle feiltolkningene i tabellen er erfaringsbaserte, dvs. at de er basert på de feiltolkningene vi har erfart med Google sin offline talegjenkjenning. Eksempler på feiltolkninger er "hive" og "i've", som justeres til "5". Med denne løsningen oppnår vi høyere treffprosent, som beskrives nærmere i neste avsnitt.

		Ord som justeres
0	Zero	see row - sierra - siri - null - co - euro - see you though - hero - see road - sarah
1	One	won - want - who won - on - what - when - won't
2	Two	too - to - to you - do
3	Three	treat - tree - the
4	Four	for - or - ford - full
5	Five	hive - i've - pipe
6	Six	sex - sick - thanks - seats
7	Seven	satin
8	Eight	ate - aight - hate - hey
9	Nine	mine - line - night
10	Ten	time - tom - turn - and - pam - done - them
11	Eleven	i love them
12	Twelve	tyler - corral - to health

Tabell 7.3: Justeringsfilter for tall

Dette avsnittet beskriver hvordan vår implementasjon forbedrer treffprosenten til den innebygde talegjenkjenningen, og gjengir i korte trekk testresultatene fra forprosjektet i vedlegg A. Det ble utført tre tester av Google sin talegjenkjenning; (1) med internett, (2) uten internett og (3) uten internett med justeringsfilter. Det ble brukt to justeringsfilter, ett for tall og ett for farger. Det førstnevnte filteret var en tidligere utgave av filteret vist i tabell 7.3, og inneholdt dermed færre feiltolkninger. Justeringsfilteret for farger vises ikke her, men kan ses i vedlegg A, fordi vanlig sauregistrering kun benytter seg av talegjenkjenning av tall. Testen ble gjennomført av 13 deltakere, og innebar å si enkeltord etterhvert som de ble vist på skjermen, og la applikasjonen tolke dem. Resultatene viste at taleregistrering med nettverksforbindelse ga en treffprosent på 95%, ingen forbindelse ga 68% og ingen forbindelse med justering ga 95%. Med andre ord klarte vi med justeringsfilter å forbedre offline talegjenkjenning med 17 prosentpoeng slik at denne ble like god (for vårt bruksområde) som online gjenkjenning uten justering. Det ble

under denne testen oppdaget flere feiltolkninger som i ettertid har blitt lagt til i filteret.

7.4.3 Skjuling av knapper under taleregistrering

Saueregistrering skiller seg fra de andre registreringstypene fordi den kan gjennomføres ved bruk av tale, som medfører at skjermen har en mikrofon-knapp for å starte taleregistreringsdialogen. Denne knappen er kun synlig om taleregistrering er tilgjengelig, altså riktig konfigurert som beskrevet i seksjon 7.4.1. Knappen skjules imens dialogen er aktiv. Det viste seg å kreve noe ekstra innsats å få implementert denne funksjonaliteten.

Årsaken til at bruk av tale ble noe krevende å implementere er todelt. Det skyldes delvis hvordan talegjennkjenningsbiblioteket vi brukte fungerer, og delvis hvordan widget-treet i Flutter fungerer. Biblioteket har callback-funksjonene "onStatus" og "onError", som henholdsvis brukes både til å lytte på statusendring (isAvailable, isListening) og til å lytte etter feil. For å vite når dialogen setter seg på pause (fordi lyttetiden er over og brukeren ikke har sagt noe, eller at brukeren har pauset dialog ved å trykke på tilbake-knapp), benytter implementasjonen seg av "onError" til å gjøre mikrofon-knappen synlig igjen. Denne callbacken defineres i bibliotekets initialize-metode, som kun skal kalles én gang i løpet av applikasjonens levetid. Når callbacken er definert, kan den med andre ord ikke endres. Siden koden for å oppdatere en widget er widget-spesifikk (setState-metoden til en widget rerenderer den), støtte vi på et problem der alt fungerte på den første saueregistrerings-skjermen, men ikke på den neste. Dette er fordi setState-metoden som ble kjørt i onError-callbacken fortsatt tilhørte den første saueregistreringswidgeten, som ikke lengre eksisterte i widget-treet. Vi ble dermed nødt til å initialisere biblioteket på et høyere nivå i widget-treet, selv om tale kun brukes for saueregistrering. Biblioteket ble dermed initialisert mange nivåer oppover i treet, i en widget som aldri poppes ut av widget-stacken under applikasjonens levetid.

Neste problem var at setState-metoden i onError-callbacken nå tilhørte en annen widget på høyere nivå i widget-treet, og at Flutter-rammeverket ikke så noen grunn til å render saueregistreringswidgeten på nytt da widgeten på et høyere nivå i treet ble rendret på nytt (fordi tilstanden til saueregistreringswidgeten var uendret). Vi måtte dermed finne en måte å be saueregistreringswidgetene om å render seg på nytt uten å ha tilgang til deres spesifikke setState-metode. Løsningen på dette ble å legge saueregistreringswidgeten i en "ValueListenableBuilder", som konfigureres til å lytte på en bool kalt "ongoingDialog". Saueregistreringswidgeten blir dermed rendret på nytt hver gang ongoingDialog endrer verdi, og mikrofon-knappen fungerer dermed som vi ønsker.

7.5 Registreringer

7.5.1 GPS-intervall

Siden enhetens GPS-posisjon normalt kun hentes hvert 15. sekund, vil posisjonen i tillegg beregnes hver gang en registrering startes. Dette gjøres hovedsaklig for å korrekt kunne beregne avstanden mellom enheten og det som registreres, og vurdere om denne er over eller under 50 meter for saueregistreringer. Oppdatert enhetsposisjon gir i tillegg mer korrekt plasserte siktlinjer mellom enheten og samtlige registreringstyper.

7.5.2 Abstrakt klasse

Alle registreringsklasser implementerer den samme abstrakte klassen da de har mange fellestrekk. Poenget med denne er å sørge for at alle klassene har funksjoner for (1) å hente GPS-posisjon og oppdatere grensesnittet når den er hentet, (2) fullføre registrering etter egen validering, (3) håndtere bruk av enhetens fysiske tilbake-knapp, og (4) håndtere at skjermen poppes av stacken (lukkes). De to førstnevnte funksjonene er abstrakte og må implementeres av hver registreringsklasse, mens de to siste ikke er abstrakte og brukes direkte. På denne måten følger hver registreringsklasse samme mal, og kodeduplisering reduseres.

7.6 Oppsynstur uten nettverksforbindelse

Oppsynsturer kan gjennomføres uten nettverksforbindelse. Ved første start av mobilapplikasjonen må man logge seg inn, og er dermed nødt til å ha nettverkstilgang. Etter innlogging vil data fra de gårdene oppsynspersonen er registrert som oppsynspersonell hos lastes ned. Etter nedlasting vil all informasjon om gården(e) som er nødvendig for å gjennomføre offline oppsynstur skrives til fil på JSON-format. Innholdet er en liste der hver oppføring er et objekt som inneholder alle feltene i de respektive "Farms"-dokumentene, med unntak av informasjon om oppsynspersonell (dvs. "name", "address", "farmNumber", "eartags", "ties" og "maps", se kodeliste 4.2). Denne filen holder seg oppdatert ved at den blir overskrevet ved hver innlogging.

Filen representerer kun den sist innloggede brukeren. Dvs. at dersom man logger seg inn med en annen brukerkonto på samme enhet, vil filen til den første brukeren overskrives av den sist innloggede brukeren. Offline oppsynstur med den første brukeren vil deretter ikke la seg gjøre før brukeren logger seg inn på nytt. Dette er et bevisst designvalg da vi mener det er rimelig å anta at en mobil enhet er privat. Det kan likevel tenkes at noen gårder ønsker å ta i bruk mobilapplikasjonen på et felles nettbrett. Løsningen på dette er helt enkelt å lagre én fil per bruker, og gi mulighet til å velge hvilken bruker man skal gå som i offline-modus. Dette har ikke blitt implementert fordi det er et spesielt tilfelle med lav prioritet.

Når mobilapplikasjonen startes uten nettverksforbindelse hopper den over innlogging, går rett til skjermen for start av oppsynstur (se figur 5.24) og sjekker om filen eksisterer. Oppsynstur kan ikke starte dersom filen ikke eksisterer, og oppsynspersonen blir i dette tilfellet møtt av en skjerm som forklarer situasjonen, som vist tidligere i figur 5.24a. Dersom filen eksisterer, vil resten av applikasjonsflyten være normal, men med to unntak. Det vil ikke være mulig å laste ned kart, og oppsynsturer vil arkiveres på enheten istedenfor å lastes opp. Hvordan det sistnevnte foregår beskrives nærmere i seksjon 7.7.

7.7 Håndtering av data fra oppsynsturer

Klassen `TripDataManager` brukes for å håndtere data fra en oppsynstur. Denne klassen har medlemsvariabler som holder data fra en oppsynstur. De fleste medlemmene er enkle typer, med unntak av registrasjoner som er en liste av `Map`-objekter hvor hver oppføring er en registrering, uavhengig av om det er av typen `sau`, `skade`, `kadaver`, `rovdyr` eller `notat`. I tillegg har klassen en "post"-metode som laster opp data den holder til Cloud Firestore, og en "archive"-metode som skriver dataen til fil, samt konstruktører for å starte en ny oppsynstur eller laste inn data fra en JSON-fil. Det blir opprettet ett `TripDataManager`-objekt i `MainPage` for hver nye oppsynstur.

Når en oppsynstur avsluttes vil enheten teste nettverkstilkoblingen. Om det er nettverksforbindelse vil data fra oppsynsturen lastes opp umiddelbart. Derimot om enheten ikke har nettverksforbindelse vil data fra oppsynsturen skrives til fil på JSON-format, i en egen katalog i applikasjonens dokumentmappe. Applikasjonen vil forsøke å synkronisere data både etter innlogging og etter fullføring av oppsynsturer, med andre ord hver gang brukeren entrer siden hvor man starter en oppsynstur. Om applikasjonen ikke blir synkronisert her, grunnet manglende nettverksforbindelse, vil det bli aktivert en tidtaker som forsøksvis synkroniserer hvert femtende sekund. Tidtakeren er utelukkende aktivert når brukeren er på siden for å starte oppsynstur og applikasjonen ikke er synkronisert, i alle andre tilfeller er den deaktivert. Ved et synkroniseringsforsøk vil `trySynchronize()` kjøres, denne metoden kaller først `isSynchronized()` som sjekker om det finnes noen arkiverte oppsynstur-filer i katalogen for lokalt lagrede oppsynsturer. Om dette er tilfellet vil `isConnectedToInternet()` sende en HTTP-foresprøsel til `google.com` for så bekrefte at enheten ikke bare er tilkoblet nettverk, men at den også har internettforbindelse. Om enheten har internett vil `synchronize()` kalles. Denne metoden vil for hver fil i katalogen lese inn filen som JSON, kalle `TripDataManager.fromJson()`-konstruktøren med innlest json som parameter og lage et nytt `TripDataManager`-objekt. Dette objektet vil kalle `post()` som laster opp data til Cloud Firestore før filen til slutt slettes fra enheten. Som beskrevet i seksjon 5.3.4 vil statusbaren på skjermen for å starte oppsynstur signalisere om applikasjonen er i synkronisert eller ikke-synkronisert tilstand.

7.8 Årsrapport

Årsrapporten genereres som en PDF med filnavn på formatet "ÅrsrapportXXXX", der XXXX er årstallet rapporten er fra. Eksempel på årsrapport kan ses i vedlegg D. Rapportens første side inneholder overskriften "Årsrapport XXXX" og en tabell med hovedinformasjon om gården og dets oppsynspersonell. Nærmere bestemt inneholder tabellen navnet, adressen og gårdsnummeret til gården, fullt navn, e-postadresse og telefon-nummer til gårdseier, fullt navn på alle oppsynspersoner som har gått oppsyn det året, i tillegg til antall oppsynsturer som har blitt gått. Rapportens resterende sider inneholder en tabell der hver rad representerer én av oppsynsturene som ble gått. Hver rad inneholder dato og tidsperiode, i tillegg til antall sauer, voksne, lam, skadde, døde og rovdyr som ble registrert på oppsynsturen. For hver 25. oppsynstur vil tabellen fortsette på en ny side med nye kolonne-overskrifter øverst, slik at leseren slipper å måtte huske hva tallene i de ulike kolonnene betyr.

7.9 Kodepraksis

For å øke kodekvaliteten utviklet vi widgets med navngitte og påkrevde parametre, akkurat som originale Flutter-widgets. Dette øker lesbarheten og gjør koden enklere å vedlikeholde og videreutvikle. For å redusere kodeduplisering utviklet vi en del mindre widgets som brukes på flere skjermer. Eksempler på disse er nedtrekksmeny for slipsvalg, inputfelter for øremerke, spinning circle, passordfelt, hjørneknapper og brytere. Videre er stilrelaterte konstanter samlet i en fil kalt "styles", som inneholder ulike tekststiler, padding, insets, feltdekorasjoner, ikonstørrelser, tabellbredder osv. Dette gjør at applikasjonenes utseende i stor grad kan endres i denne filen.

Kapittel 8

Testing

I dette kapitlet blir testing av systemet og dets brukergrensesnitt beskrevet. Først gis det en beskrivelse av programvaretestingen og konfigurasjonen av denne. Deretter skisseres brukertestens prosedyre, før kapitlet avslutningsvis presenterer resultatene fra denne.

8.1 Automatisert testing

Testing i Flutter er inndelt i tre typer; (1) unittester, (2) widgettester og (3) integrasjonstester. Unittester tester logikken til enkeltfunksjoner, metoder og klasser. Widgettester tester at utseendet til, og interaksjonen med, en widget er som forventet. Integrasjonstester tester større deler av systemet, og skal verifisere at de delene (også eksterne) som testes samarbeider som forventet. [13]

Prosjektet har brukt automatisert testing i form av to workflows i GitHub Actions. Den første workflowen utfører statisk kodeanalyse, sjekker kodeformatering og kjører unit- og widgettester, mens den andre kjører integrasjonstester. Workflowene kjøres ved hver pull request mot main-branchen for å sørge for at all kode er kvalitetssikret før den merges inn i main.

Workflow 1

Den første workflowen kjører på en Ubuntu-maskin og utfører Flutter-kommandoene format, analyze og test. "Flutter format"¹ sørger for at all kode har samme stil, og workflowen feiler dersom det ikke er tilfelle. Vi har konfigurert vårt lokale kodeeditor til å formattere kodefilene hver gang endringer lagres, ved å bruke en Dart-utvidelse², slik at formatterings-sjekken i workflowen i praksis fungerer som en ekstra sikring. "Flutter analyze"¹ utfører statisk kodeanalyse, og gjør at workflowen feiler dersom den eksempelvis oppdager ubrukte imports, tomme if/else-blokker eller logikk i createState-metoden. "Flutter test"¹ kjører testene i

¹<https://docs.flutter.dev/reference/flutter-cli>

²<https://dartcode.org/>

alle filer som ligger innunder "test"-mappen i prosjektets rot-nivå, med filnavn på formatet "**_test.dart".

Workflow 2

Den andre workflowen kjører på en macOS-maskin og utfører diverse delkommandoer før den til slutt kjører integrasjonstester med hovedkommandoen i kodeliste 8.1.

Kodeliste 8.1: Kommando for å kjøre integrasjonstester i app og web

```
cd app && firebase emulators:exec 'flutter test integration_test' &&
cd ../web && firebase emulators:exec 'flutter test integration_test'
```

Integrasjonstesting av mobilapplikasjonen gjøres mot en kjørende Android-emulator, som typisk konfigureres gjennom Android Studio³. For å teste Firebase-relatert funksjonalitet, altså widgets som utfører eksterne nettverkskall, brukes Firebase sin "Local Emulator suite"⁴, som eksempelvis installeres gjennom npm⁵. Vi bruker én Authentication-emulator og én Cloud Firestore-emulator. Hvilke Firebase-emulatorer som skal brukes, og hvilke porter de skal kjøre på, konfigureres i filen "firebase.json" som ligger på rot-nivå i både mobil- og webprosjektet. For å få applikasjonene til å kommunisere med emulatorene istedenfor Firebase, spesifiserer testfilene at Authentication-instansen og Firestore-instansen skal bruke emulatorene på de valgte porter.

Integrasjonstestene er ganske rett frem å kjøre lokalt på egen datamaskin, men krevde noe mer tid å sette opp i GitHub Actions. Her benytter vi oss av en action kalt "Android Emulator Runner"⁶ for å kjøre Android-emulator. Denne fungerer ikke på Ubuntu-maskiner, og bør kjøres på macOS for å utnytte maskinvareakselerasjon. Dette er en av to årsaker til at integrasjonstesting gjøres i egen workflow. Den andre årsaken er at minuttene en macOS-maskin bruker er ti ganger dyrere enn på en Ubuntu-maskin⁷. Det siste steget som må gjøres før integrasjonstestkommandoen (i kodeliste 8.1) kan kjøres er å installere npm-pakken "firebase-tools"⁸, som gjør at man kan kjøre Firebase-emulatorer.

Integrasjonstesting mot nettleser viste seg å være trøblete fordi dette ikke støttes av Flutter enda. Vi forsøkte derfor å følge Flutter sin guide⁹ med en alternativ fremgangsmåte som krever installasjon av ChromeDriver¹⁰. Denne fremgangsmåten fungerte helt til vi initialiserte en Firebase-applikasjon. Da slutter testen

³<https://developer.android.com/studio/run/managing-avds>

⁴<https://firebase.google.com/docs/emulator-suite>

⁵<https://docs.npmjs.com/about-npm>

⁶<https://github.com/marketplace/actions/android-emulator-runner>

⁷<https://docs.github.com/en/billing/managing-billing-for-github-actions/about-billing-for-github-actions>

⁸<https://www.npmjs.com/package/firebase-tools>

⁹<https://docs.flutter.dev/cookbook/testing/integration/introduction>

¹⁰<https://chromedriver.chromium.org/home>

å feile når den skal (til og med "assert(false);" kjører grønt), noe som hindret oss i å integrasjonsteste alle deler av webapplikasjonen som involverer Firebase-funksjonalitet. Vi fikk deretter en idé og valgte å kjøre integrasjonstesting av webapplikasjonen på Android-emulator, for de web-widjetene som passer på en mobilskjerm. Vi besluttet videre å kun kjøre vanlig widget- og unittester på resterende web-widgets siden det er bedre enn ingen test.

Testbegrensninger

Etterhvert som vi utviklet flere integrasjonstester, støtte vi på problemer. Det første, og mest betydelige, var at workflowen feilet på helt tilfeldige steder i pipeline av maskin-spesifikke årsaker, som vi med andre ord ikke hadde mulighet til å gjøre noe med. Et eksempel er diverse timeouts fordi emulatoren ikke startet, eller plutselig ikke svarte halvveis gjennom testene. Det andre problemet var at det ble svært ressurskrevende. Kjøring av integrasjonstestene på våre egne maskiner tok rundt seks minutter, mens det for pipeline tok 16-18 minutter. Den største årsaken til denne tidsbruken er at runneren vi blir tildelt er langt fra optimalisert for emulator-relatert testing, som krever en del maskinressurser. En mindre årsak er at firebase-tools installeres på runneren hver gang den kjøres, noe som tok rundt fire minutter. Med helt tilfeldig ikke-testrelatert feiling og over et kvarters ventetid per pipeline, kunne det fort ta en time før pipeline fikk en suksessfull gjennomkjøring. Dette førte til utsettelse i utviklingen, var et irritasjonsmoment og ledet i tillegg til svært høyt forbruk av gratisminutter (spesielt siden macOS-minutter er dyrere). På bakgrunn av dette valgte vi å droppe automatisert integrasjonstesting, og kjørte heller disse manuelt på egne datamaskiner. Statisk kodeanalyse, formateringssjekk og widget- og unittester forble automatisert.

Vi syntes ikke det var noe poeng i å teste dersom dette ikke ble gjort nøye. Derfor testet vi ikke bare happy day scenario, men også alle spesielle tilfeller. I dialog med veileder bestemte vi oss 10. mars for å nedprioritere testing av programvaren, fordi dette var svært tidkrevende og ville føre til at kravspesifikasjonen ikke hadde blitt oppfylt, fordi vi ikke hadde rukket å ferdigstille applikasjonene. Det som likevel har blitt grundig testet er innlogging og registrering av brukerkonto i begge applikasjoner, samt nedlasting av kart, start av oppsynstur og registrering av sau for mobilapplikasjonen, i tillegg til innfylling av feltene i menyen "Min gård" for webapplikasjonen.

8.2 Brukertesting

Før brukertesting startet, utførte vi den planlagte brukertesten på oss selv.

8.2.1 Prosedyre

Vi vil i denne seksjonen beskrive hovedpunkter ved brukertesten. Detaljert prosedyre ligger i vedlegg E. Begrunnelse av prosedyren og en forklaring til hvorfor de utvalgte delene av systemet som testes ble valgt gis i seksjon 9.2.

Testintroduksjon

Hver testdeltaker ble først gitt generell informasjon om hva som skulle foregå, at det er systemet og ikke deltakeren som skulle testes, og deltakerne ble oppfordret til å tenke høyt. De ble gjort oppmerksomme på at de kom til å bli vist et bilde av en skadet sau og et sauekadaver for å høre om de var komfortable med dette. Deretter ble det gitt en kort forklaring til domenet, mer spesifikt hvorfor det blir gått oppsynstur, hva som er interessant å registrere og at sauer merkes med øremerker og slips. Deltakerne ble deretter vist et bilde av en sau med øremerke og slips, og gitt en forklaring av disse. Da testen skulle starte fikk deltakerne scenariet ”Du er en sauebonde som skal ta i bruk en ny løsning for oppfølging av sau på beite”. Videre er brukertesten delt inn i fire deler med ulike oppgaver.

Del 1

Første del foregikk på webapplikasjonen, med hensikt å undersøke om metoden for definering av beiteområde var intuitiv. Deltakeren ble gitt en laptop som allerede var innlogget på en brukerkonto der alt annet enn beiteområder var ferdig definert. Deretter fikk deltakerne første oppgave, som var ”Legg inn Moholt Studentby som beiteområde”. Her ble det undersøkt om det var enkelt å navigere seg til riktig side, definere beiteområde, gi det et navn og lagre det.

Del 2

Andre del foregikk på mobilapplikasjonen. Deltakerne ble forklart at det var dårlig nettverksforbindelse i området, og at det derfor var nødvendig å laste ned kartet før vi skulle dra til området der oppsynsturen skulle gjennomføres. Deretter ble deltakerne gitt en mobiltelefon og fikk oppgaven ”Last ned kartet”. Her var hensikten å undersøke hvorvidt det var forståelig at nedlast-ikonet på startskjermen er til å trykke på, eller om deltakerne istedenfor trykker på knappen for å starte oppsynstur.

Del 3

På den tredje delen skulle deltakerne gjennomføre en simulert oppsynstur. De fikk beskjed om å slå følge med oss, og fikk oppgaven ”Utfør registreringer når du blir vist bilder” (bildene kan ses i vedlegg E). Denne delen var nøye planlagt på forhånd, og planlagt rute vises i figur 8.1. I figuren er ruten som skal gås markert gul, startpunktet er den grønne sirkelen og gåretningen vises av den grønne pekeren. Det skal gjøres én registrering på posisjonen til hver lilla sirkel, og lilla stiplede linje

fungerer som siktlinje. Hver lilla sirkel har et tall i seg, som definerer rekkefølgen registreringene skal gjøres i. I stigende rekkefølge er innholdet på bildene som vises til hver registrering som følger:

1. Brun sau og to hvite lam (avstandregistrering)
2. Sau og lam med rødt øremerke (nærregistrering)
3. To ulver
4. Skadd sau med gult øremerke, og øremerketegning med ID
5. Sauekadaver med gult øremerke, og øremerketegning med ID
6. Ødelagt gjerde
7. Søye og to lam, rødt slips og grønne øremerker (nærregistrering, dialog)
8. Svarthodet sau, hvit sau og fem hvite lam (avstandsregistrering, dialog)

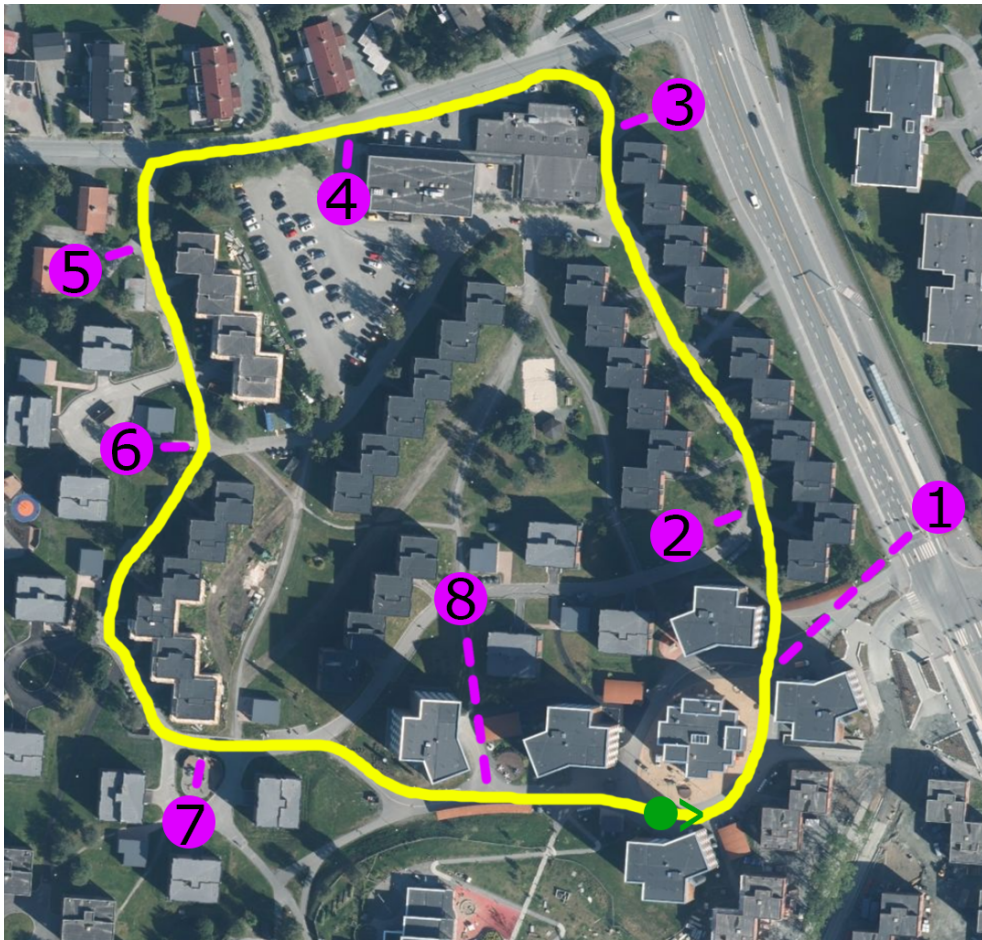
Registreringene, spesielt saueregistreringene, har økende vanskelighetsgrad. Først nærregistrering, altså antall sauer og lam og antall av hver ullfarge, deretter nærregistrering som i tillegg involverer telling av de ulike øremerke- og slipsfargene. I de to siste registreringene skal taleregistreringsdialogen testes. Deltakerne får ikke beskjed om at taleregistreringen er dialogbasert, men det nevnes at det finnes en taleassistent. Deretter bes deltakerne om å se så lite som mulig på skjermen etter å ha påbegynt dialogen. Vi ønsket her å undersøke om grensesnittet var forståelig, mer konkret om deltakerne forstod at dialogen startes ved å trykke på mikrofonknappen, om lydsignalet for pauset dialog forstås, og om deltakerne gjenopptar dialogen dersom tiden de trengte til å telle var lengre enn applikasjonens lyttetid. Vi valgte for den siste registreringen å vise et bilde av mange sauer og lam, både for å se om den Android-spesifikke lyttetiden er så liten at dialogen går i pausemodus imens deltakeren teller, men hovedsaklig for å undersøke hvordan dialogen fungerer når brukeren har flere tellinger å holde styr på uten å se på skjermen.

Del 4

På den fjerde og siste delen av brukertesten skulle deltakerne, i webapplikasjonen, se gjennom oppsynsturen de nettopp gikk. Første oppgave var ”Se på oppsynsturen du gikk”, med formål å se på navigering til riktig hovedfane, og om det er underforstått at den nyligste oppsynsturen ligger øverst i listen over oppsynsturer. Andre oppgave var ”Se på bildene du tok av sauekadaveret”, der hensikten var å se om grensesnittet gjør det enkelt å finne frem til enkeltregistreringer. Den siste oppgaven var ”Se på hva du registrerte på lekeplassen”, og vi ønsket her å undersøke om deltakerne forstod at markørene i kartet er klikkbare.

Avslutning

Hver deltaker ble gitt flax-lodd som takk for deltakelsen. Videre var det lagt inn gode pauser mellom hver deltaker slik at notater kunne renskrives og utfylles for å være enklere å forstå senere. Begge roller samarbeidet om en konkluderende oppsummering, for å ta i bruk begge synspunkter.



Figur 8.1: Planlagt rute for brukertest

Testroller

Vi gikk selv inn i rollene som testleder og observatør, som forklares her.

Testleder hadde hovedansvaret for å løse deltakerne gjennom hele testen. Det innebar å introdusere testen, gi forklaring til domenet og vise bilder ved hver registrering. Testlederen stilte i tillegg spørsmål der det var naturlig, for å få deltakerne til å si hva de tenkte.

Observatøren hadde to oppgaver. Den første var å følge med på testdeltakerens samhandling med applikasjonene og notere alt som var av interesse i et skjema. Observatøren måtte hele tiden kunne se skjermen deltakerne brukte. Den andre oppgaven var å sørge for at alt det tekniske var i orden ved å klargjøre enhetene som skulle brukes (brukerkontoer og innlogging), både under og mellom hver gjennomføring.

8.2.2 Resultater

Vi vil i denne seksjonen presentere resultatene fra totalt ni gjennomføringer av brukertesten. Notater fra hver enkelt gjennomføring ligger i vedlegg F. Totalt sett gikk gjennomføringene bra, alle testedeltakerne fikk til å løse oppgavene de fikk. Det var likevel noen vanskeligheter som gikk igjen, noen mindre misforståelser og enkelte deltakere som brukte lengre tid enn andre på samme oppgave.

Del 1

På testens første del fikk deltakerne i oppgave å definere et beiteområde som strekker seg over Moholt Studentby. Alle fikk til å navigere seg fra startsiden til siden for definering av beiteområde. Her var det kun et fåtall som tok seg tid til å lese den overordnede beskrivelsen, som riktignok ikke er spesielt viktig å få med seg for å løse oppgaven. Samtlige deltakere trykket på pluss-knappen for å legge til beiteområde. Noen få valgte å skrive inn navn på beiteområdet først, men de fleste gjorde det til slutt. Videre kan deltakerne deles inn i to kategorier; (1) de som leste kartdefinerings-instruksjonene før de gikk i gang med å definere område, og (2) de som leste instruksjonen etterhvert. Kun fire av de ni deltakerne havnet i den første kategorien, og disse fire hadde ingen problemer med å markere et område i kartet. De resterende fem deltakerne havnet i den siste kategorien, og det er usikkert hvorvidt instruksjonen ble oppdaget etterhvert eller om de bare droppet å lese den. Disse fem hadde små problemer med å få definert et område før instruksjonen ble lest, og forsøkte blant annet å klikke diverse steder og å holde og dra i kartet. Kort tid senere leste alle fem instruksjonene, og fire av disse klarte deretter å markere et område. Den siste deltakeren forsøkte (etter å ha lest) klikk, hold og dra, og forstod ikke at dette er noe annet enn klikk og hold. Her måtte det bli gitt noen små hint for at oppgaven skulle løses. Her lurte vi litt på om det er instruksjonen som ikke er tydelig nok, eller om det var tilfeldigheter som gjorde

at instruksjonen ikke nådde gjennom til én av ni deltakere.

Testens første del bekreftet mistankene vi hadde vedrørende kartdefinerings brukervennlighet. Seks av de ni deltakerene var ikke fornøyd med enten første hjørneplassing eller hele det markerte området fordi det ble for lite. Samtlige seks deltakere forsøkte å rette opp i dette, de fleste ved å prøve å dra i hjørner eller markeringslinjer, de resterende lette etter angre-funksjonalitet. Alle seks endte opp med å avbryte defineringen og starte på nytt, og to av disse ble heller ikke fornøyd denne gangen og begynte på nytt en tredje gang.

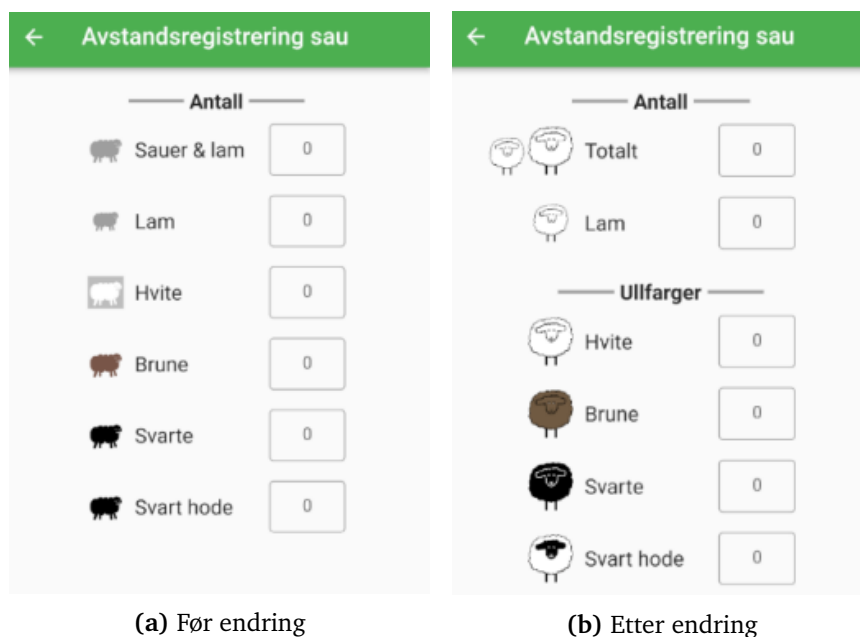
Del 2

Under andre del av testen var oppgaven å klargjøre mobilapplikasjonen ved å laste ned kartet, og det skulle undersøkes hvorvidt deltakerne forstod at nedlast-ikonet ved siden av kartområdet er klikkbart. Seks av de ni deltakerne trykket på nedlast-ikonet, de resterende tre trykket på knappen for å starte oppsynstur. Samtlige deltakere løste oppgaven uten problemer, siden kartet uansett lastes ned, men det er likevel interessant å se at fremgangsmåten var ulik.

Del 3

Tredje del av testen innebar å gå en simulert oppsynstur bestående av åtte registreringer. Samtlige deltakere navigerte seg rundt i kartet, zoomet inn, roterte, forstod ikonet for egen posisjon og linjen som viser ruten som er gått. Videre fikk alle til å gjøre registreringene på egenhånd, inkludert taleregistrering, og forstod hvordan dialogen gjenopptas, hvordan registreringer avbrytes og hvordan oppsynsturen avsluttes. Det er likevel noen mindre situasjoner som gjentok seg underveis.

På de fire første gjennomføringene var det tre deltakere som stusset på betydningen av "Sauer & lam"-feltet på vanlig sauregistrering. To av disse lurte på om det betydde totalt antall og kom til riktig konklusjon. Den siste av de tre hadde problemer med å forstå hva som var galt i registreringen, og feilmeldingen som påpekte problemet var ikke hjelp nok. Deltakeren fortalte etter gjennomføringen at den ble forvirret av at ullfarge-feltene vises rett under "Sauer & lam"-feltet (deltakeren tenkte at tre hvite indikerer at det er tre sauer og lam, og tenkte ikke videre på at sauer og lam skal skilles i tellingen). En annen tilbakemelding vi fikk var at én deltaker trodde knappen i hovedskjermens nedre høyre hjørne med teksten "+?" ville lede til en slags hjelp-meny, og foreslo å fjerne spørsmålsteget. Med bakgrunn i det som ble avdekket etter fire gjennomføringer, valgte vi gjøre to visuelle modifikasjoner for å undersøke om disse bedret intuitiviteten av brukergrensesnittet i de påfølgende gjennomføringene. Den første var å endre "Sauer & lam"-teksten til "Totalt", legge til et mindre saue-ikon ved siden av for å vise både voksen sau og lam, og å endre til figurer vi selv har laget som bedre symboliserer de ulike ullfargene. Den andre modifikasjonen var å endre knappeinnholdet fra teksten "+?" til å være et pluss-ikon. Endringene vises i figur 8.2 og 8.3. Det kan



Figur 8.2: Utklipp av saueregistrering før og etter endring

tyde på at endringene i saueregistreringen var suksessfulle da alle deltakerne i de fem resterende gjennomføringene raskt fylte inn feltene uten å stusse på betydningene. Det er vanskelig å vurdere suksessen til den andre endringen, men forståelsen av knappen ble hvertfall ikke noe lavere da deltakerene fortsatte å bruke knappen til riktig tid.

Syv av ni deltakere startet minst én saueregistrering ved å holde inne på kartet uten å gå gjennom plussmenyen først. Likevel viste det seg at ikke alle husket eller forstod at dette alltid åpner saueregistrering. Det er verdt å merke seg at det ikke er kommunisert at holding på kartet åpner saueregistrering direkte, og at dette kan forveksles med holding på kartet etter å ha valgt registreringstype fra plussmeny. Seks av deltakerne som hadde åpnet saueregistrering direkte gjennom kartet minst én gang tidligere på turen, holdt senere direkte på kartet og avbrøt registreringen fordi de skulle registrere noe annet enn sau, og én deltaker måtte avbryte to ganger på rad.

Minst fire av de ni deltakerne forsøkte å zoome lengre inn i kartet uten at dette var mulig, fordi de allerede hadde nådd det høyeste zoomnivået som var nedlastet. Dette er en indikasjon på at de ønsket minst ett zoomnivå høyere, men vi er noe usikre på om situasjonen hadde vært den samme dersom oppsynsturen hadde foregått i reelle omgivelser. Dette er fordi området den simulerte oppsynsturen ble gått i er tettbebygd, noe som betyr at det er ekstra mange referansepunkter, som sannsynligvis får deltakerne til å være mer presise på registreringsposisjoner enn de hadde vært i reell terreng. Det er likevel en interessant observasjon som må



(a) Før endring (b) Etter endring

Figur 8.3: Utklipp av plussmeny-knapp før og etter endring

testes i faktiske beiteområder for å få et sikkert svar.

Åtte av de ni deltakerne opplevde å få feilmelding på saueregistrering, altså at noen av tallene ikke stemte overens med totalt antall sauer og lam. Seks av disse forstod feilmeldingen og rettet opp raskt, imens de to resterende brukte litt tid på å forstå den. Det er verdt å nevne at den første av de to brukte den gamle registreringsskjermen (figur 8.2a) og slet med å forstå betydningen av "Sauer & lam". Den andre av de to forsøkte gjentatte ganger å fullføre registreringen til tross for feilmeldingen, før den til slutt gikk inn for å prøve å forstå hvorfor det ikke gikk. I tillegg var det to deltakere som fikk valideringsfeil på øremerkets individnummer, den ene fordi feltet ikke ble fylt ut, den andre fordi kun tre av fem siffer ble skrevet inn ("123" istedenfor "00123"). Begge deltakerne forstod hva som var galt gjennom rødt tekstfelt med tilhørende feilmelding, men deltakeren som fylte inn tre siffer forstod først ikke hvorfor registreringen ikke ble fullført, fordi feltet og feilmeldingen ikke var synlige. Her ble det avdekket et spesielt tilfelle; dersom man har tastaturet åpent på skade- eller kadaverskjermen og har scrollet øremerkefeltene ut av skjermen så får ikke brukeren med seg at disse feltene er ugyldige.

De to siste registreringene testet taleregistreringsdialogen. Overordnet ser vi at denne fungerte bra fordi alle deltakerne fant ut hvordan de starter taleregistrering, og fordi alle kom seg gjennom uten hjelp og uten å få vite at det er en dialog. Alle innrettet seg etter dialogens lydssignaler etter kun få spørsmål, og alle fikk til å gjenoppta dialogen dersom den stanset. Videre var det et par hendelser vi bet oss merke i. Fire av de ni deltakerne svarte med flere ord første gang de skulle taleregistrere ("three sheep" eller "x sheep y lambs"). Da taleassistenten sa at den ikke forstod hva som ble sagt og stilte spørsmålet på nytt, skjønnte de at de kun skulle svare med ett tall. Vi anser ikke dette som et problem siden alle fire forstod hvordan taleregistreringen fungerer etter kun ett feilforsøk. Det er mulig at et ikon med to mennesker som snakker tydeligere hadde symbolisert dialogbasert registrering. Et fåtall av deltakerne ble stående og vente en liten stund etter endt dialog, sannsynligvis fordi det ikke ble gitt noe signal om at dialogen var gjennomført.

Del 4

På testens siste del fikk deltakerne i oppgave å se på oppsynsturene de gikk i webapplikasjonen. Alle fikk til å navigere seg til riktig hovedfane og antok at den øverste oppsynsturen var den nyeste, noen dobbeltsjekkete dette ved å se på tidsperioden. Alle løste også den andre oppgaven, men med noe ulik fremgangsmåte. Seks av de ni deltakerne åpnet popup-vinduene med både markør i kart og knapp på høyre del av siden, mens de tre resterende kun benyttet seg av kartmarkører. Videre ser det ut til at alle forstod at den høyre delen av siden er scrollbar.

Oppsummering

Del én, kartdefinerings i webapplikasjonen, er nok den delen med høyest forbedringspotensiale, siden flere av deltakerne hadde ulike startvansker. For selve defineringen av beiteområde ser vi en klar korrelasjon mellom å lese instruksjoner og å få til definering på første forsøk, og at instruksjonen med fordel kan tydeliggjøres gjennom større skriftstørrelse, annen tekstfarge eller lignende. Videre mener vi det er svært sannsynlig at alle deltakerne fint hadde fått til denne oppgaven dersom de ble gitt den på nytt, fordi de har lært.

For del to og fire er det ikke så mye å si, alle deltakerne løste oppgavene uten problemer. Del tre gikk over all forventning, selv om det kom frem noen mindre situasjoner. To av situasjonene ble løst gjennom de visuelle modifikasjonene vist i figur 8.2 og 8.3. Et annet poeng som kom frem var at selv om de fleste deltakerne holdt direkte inne på kartet, var det ikke alle som lærte seg at dette leder direkte til sauregistrering. Videre ble det avdekket at feilmeldingen for øremerke-input ikke alltid er synlig på skjermen dersom tastaturet er åpent. Til slutt vil vi legge til at alle forstod applikasjonsflyten og at feilmeldinger og taleregistreringsdialog fungerte bra, selv om én av tilbakemeldingene fra deltakerene var at det tok litt lang tid å gå gjennom hele dialogen for nærregistrering.

Kapittel 9

Diskusjon

Dette kapitlet inneholder en samling diskusjoner rundt ulike aspekter ved prosjektet. Kapitlet bygger videre på innhold presentert tidligere i rapporten. Teknologimessige valg drøftes, ytterlige sider ved brukertesting gres ut, muligheter for taleregistrering beskrives, sauebønder som marked vurderes, valg rundt utforming begrunnes, fremdriften reflekteres over, mangler og kjente feil berettes, og forslag til forbedringer og videre arbeid fremmes før oppgaven avslutningsvis kort kritiseres.

9.1 Valg av teknologier

Denne seksjonen beskriver de teknologi-alternativene som ble vurdert, og begrunner valgene som ble tatt. En nærmere beskrivelse av de valgte teknologiene gis i seksjon 7.1.

9.1.1 Rammeverk for frontend

Rammeverk/SDK (Software Development Kit) for frontend må støtte talegjennkjennning og fremvisning av kart, samt generering og eksportering av PDF.

Android SDK/iOS SDK er en rekke verktøy utviklet av Google/Apple for å lage native Android-/iOS-applikasjoner¹ ². Fordelen med å utvikle en native applikasjon er at man jobber tett på operativsystemet og dermed har muligheten for god ytelse. Ulempe er at man er begrenset til én plattform. Vi valgte tidlig å gå bort fra disse SDK-ene fordi ytelse ikke ble sett på som en flaskehals, fordi vi ikke ville begrense oss til én plattform, og fordi vi verken hadde tid til å utvikle to versjoner av mobilapplikasjonen. Vi hadde uansett ikke mulighet til å utvikle mot iOS fordi vi ikke hadde tilgang på Mac-maskiner, som er påkrevd.

¹https://en.wikipedia.org/wiki/Android_SDK, besøkt 11.01.2022

²https://en.wikipedia.org/wiki/iOS_SDK, besøkt 11.01.2022

React Native er et rammeverk utviklet av Facebook som muliggjør bruk av web-rammeverket React for å lage mobilapplikasjoner³. Med React Native kan man lage native applikasjoner til Android og iOS med en felles kodebase. Man vil likevel ikke oppnå samme ytelse som i en ”ekte” native app da rammeverket i seg selv gir overhead.

Flutter er en SDK utviklet av Google for utvikling av applikasjoner for Android, iOS og web med en felles kodebase [13]. Flutter skal være noe enklere enn React Native ved bruk av kontinuerlig geoposisjoner og kamera⁴. Da det ikke er store forskjeller mellom mulighetene som tilbys av React Native og Flutter velger vi å gå videre med Flutter, fordi vi er kjent med React og ønsker å lære noe nytt.

9.1.2 Rammeverk for backend

Backend-rammeverk må støtte server og database. Teknologi-alternativene som ble vurdert inkluderer MERN-stack og Firebase.

MERN -stacken er en anerkjent kombinasjon av de fire teknologiene MongoDB (dokument-database), Express JS (web-rammeverk for Node), React JS (frontend-rammeverk) og Node JS (web server)⁵. Noe av det som gjør denne stacken solid er skalerbarheten til MongoDB, hastigheten til Express JS og ”tilstanden” (statefulness) til events i React, samt deres gode dokumentasjon og store fellesskap.

Firebase ble valgt fremfor MERN-stack da vi mente at det var mest hensiktsmessig i vårt prosjekt som er begrenset i tid. Ved å bruke Firebase ville vi få mer tid til å utvikle funksjonalitet, fordi vi da ville unngå å utvikle en backend fra bunn, og drifte og vedlikeholde egen server.

9.1.3 Kart-bibliotek

Både Leaflet⁶ og Mapbox⁷ er tilgjengelig for Flutter. Begge er klienter for fremvisning av kart som også tilbyr funksjonalitet som nedlastning av kart og tegning på kart. Siden Leaflet er open source og gratis, og mapbox koster etter en viss mengde bruk, valgte vi å bruke Dart-utgaven av Leaflet.

9.1.4 Talegjenkjennings-bibliotek

Som nevnt i kravspesifikasjonen (kapittel 3) må talregistrering fungere uten nettverksforbindelse. I forprosjekt-rapporten (vedlegg A) ble Google sin talegjenkjenning testet grundig, og det ble besluttet at den er god nok for mobilapplikasjonens

³<https://reactnative.dev/>, besøkt 11.01.2022

⁴<https://nix-united.com/blog/flutter-vs-react-native/>

⁵<https://www.mongodb.com/mern-stack>, aksessert 11.01.2022

⁶https://pub.dev/packages/flutter_map, 11.02.2022

⁷<https://github.com/flutter-mapbox-gl/maps>, 11.01.2022

bruksområde. For at man skal kunne bruke denne, ble det en forutsetning at biblioteket vi valgte måtte støtte native talegjenkjenning. Det naturlige valget ble dermed Speech to text⁸, fordi det bruker enhetens native talegjenkjenning, som for både Android og iOS fungerer uten nettverksforbindelse.

9.1.5 Database

Siden Firebase ble valgt som backend, stod valget av database-teknologi mellom Cloud Firestore og Realtime Database. Disse har noen likheter, som at begge er NoSQL-databaser, begge støtter sanntidsoppdateringer og man betaler kun for det man bruker⁹. Videre er de ulike på flere områder, men den avgjørende faktoren i dette prosjektet handler om strukturering av data. Realtime Database lagrer data som ett stort JSON-tre, som gjør det rett frem å lagre enkel data, men vanskelig å organisere hierarkisk data. Cloud Firestore er dokument-orientert, som betyr at det er enklere å strukturere hierarkisk ved bruk av samlinger, dokumenter og undersamlinger. Vi valgte Cloud Firestore fordi denne ville gjøre det enklere å strukturere og bruke den dataen våre løsninger kom til å generere.

Cloud Firestore egnet seg likevel ikke til å lagre bilder, fordi dokument-størrelsen er begrenset til 1MB. Derfor bruker vi skylagringstjenesten Cloud Storage for dette.

9.2 Brukertest

9.2.1 Deltakerveiledning

For å unngå å teste deltakernes domenekunnskaper og heller fokusere på systemets brukervennlighet, fortalte vi deltakerne alt de trengte å vite om domenet. Vi svarte også på eventuelle domene-relaterte spørsmål deltakerne måtte ha underveis. Deltakerne fikk ikke hjelp til oppgaveløsning med mindre dette ble nødvendig for å komme seg videre. Vi mente at dette var den beste fremgangsmåten fordi det ville gi svar på hvorvidt systemet var brukervennlig eller ikke, for personer uten forkunnskaper om grensesnittet.

9.2.2 Avgrensning

Vi forstod raskt at vi ikke hadde mulighet til å teste systemet i sin helhet, fordi det består av mange skjermer og ulike edge-cases. Det ville ha vært svært tidkrevende å teste alt, også for testdeltakerne. Vi grovestimerte at en helhetlig testgjennomføring av systemet ville ha tatt over én time, og vi synes det var urimelig å be testdeltakere om å bruke så mye tid på en første test av systemet. Vi ønsket å få gjennomført åtte til ti tester, og fryktet at det skulle bli krevende å få nok deltakere

⁸https://pub.dev/packages/speech_to_text

⁹<https://firebase.google.com/docs/database/rtdb-vs-firestore>, besøkt 19.05.22

til å sette av én time av dagen sin, spesielt fordi vi ikke hadde noen stor premie (det ble brukt flaxlodd).

Vi så oss dermed nødt til å velge hvilke deler av systemet som skulle testes. Utvelgingen ble gjort med bakgrunn i tre tanker. Den første tanken var at det som var mest hensiktsmessig å teste var selve oppsynsturen, fordi det var denne som gav opphav til prosjektet, og fordi prosjektets måloppnåelse i stor grad vil vurderes på bakgrunn av denne. Derfor var den største delen av testen dedikert til en simulert oppsynstur med fokus på å teste alle registreringstyper, både med og uten tale. Den andre tanken var at vi ikke så noe behov i å teste sider vi mente var selvforklarende og mindre sentrale. Dette inkluderte innlogging og opprettelse av brukerkonto, samt de sidene i webapplikasjonen som består av svært enkle skjema. Vi ønsket derimot å teste oppsynstur-fanen i webapplikasjonen, fordi det mest sentrale use-caset for sauebonden er å se gjennomførte oppsynsturer. Den tredje og siste tanken var at vi regnet med at defineringsen av beiteområde i webapplikasjonen var noe knotete og lite brukervennlig, og at vi derfor ville ha stor interesse av å teste den for å finne ut av hvordan den kan forbedres.

9.2.3 Teknisk gjennomførelse

Testdeltakerne ble gitt ferdig innloggede enheter, der de delene av systemet som ikke skulle testes var ferdig satt opp. På denne måten unngikk vi lagring av potensielle personvern-relaterte data fordi deltakerne ikke ville bli knyttet til egne brukerkontoer. For defineringsen av beiteområde i webapplikasjonen ble det brukt en brukerkonto der alt utenom beiteområder var ferdig definert. Før neste testgjennomføring ble kontoen klargjort ved å slette området deltakeren definerte. For de tre resterende delene av testen ble det brukt en annen brukerkonto der vi i tillegg hadde definert beiteområde, slik at alle deltakerne stilte så likt som mulig. Da unngikk vi også følgefeil i tilfelle deltakerne ikke skulle få til å markere hele området vi skulle gå oppsynstur i. I tillegg ble alle resultatene samlet på én brukerkonto, som tilerrettela for enkel oversikt i etterkant.

9.2.4 Læring

Et viktig poeng vedrørende resultatet av brukertestene er at systemet læres. Det vil si at i begynnelsen av hver testgjennomføring vil resultatene sannsynligvis være dårligere enn resultatene på slutten av gjennomføringen. Altså vil begynnelsen av brukertesten være det tidspunktet brukerne samhandler dårligst med systemet. Verdien av brukertestene ligger derfor i stor grad i å se hvor raskt brukerne lærer seg systemet. Jo fortere brukerne lærer seg det, jo mer intuitivt utformet er det. Et system som derimot tar lang tid å lære seg kan føre til aggresjon og fortvilelse og at brukerne frastår fra å bruke systemet.

Vi ser fra resultatene beskrevet i seksjon 8.2.2, at brukerne i hovedsak lærte seg

systemet raskt. Samtidig var det to punkter vi la spesielt merke til når det kommer til læring. Første punkt gjelder definering av beiteområde. Her trengte brukerne tid til å lære seg applikasjonen, i tillegg måtte sidens instruksjoner benyttes for å lære fremgangsmåten fordi funksjonaliteten ikke var selvforklarende. Dette er nok fordi fremgangsmåten er atypisk, slik at brukerne ikke gjenkjenner den fra andre systemer. Da brukerne hadde gjennomført én iterasjon av å definere beiteområde viste det seg at fremgangsmåten ble lært fordi det videre gikk svært raskt å gjøre det på nytt.

Andre punkt som er verdt å merke seg med tanke på læring av systemet gjelder stedfesting av registreringer ved å holde inne på kartet. Dette gjøres enten direkte på kartet for å registrere sau eller etter å ha valgt en registrerstype via plussmenyen. Her viste det seg at brukerne ikke lærte flyten etter de åtte registreringene. Tanken var at siden en typisk oppsynstur har flest registreringer av sau vil det være gunstig å nå dette hurtig. Årsaken til at dette ikke ble lært så raskt kan være at det er en skjult funksjonalitet at man kan gå direkte til saueregistrering fra kartet. Samtidig inneholdt testrunden (vist i figur 8.1) uvanlig stort mangfold av type registreringer slik at dette kan ha vært vanskelig å forstå med så få forsøk. Det kan også argumenteres for at siden flere holdt inne på kartet og deretter avbrøt saueregistrering, at dette er den intuitive registreringsflyten, altså å holde inne på kartet og deretter velge type registrering. Argumentet mot dette er at det i en typisk oppsynstur vil være svært mange saueregistreringer sammenlignet med de andre registreringstypene. Dette betyr at saueregistrering direkte fra kartet fører til færre klikk. Økt bevissthet om denne funksjonaliteten i form av en beskjed ved førstegangsbruk, eller når brukeren har registrert sau via plussmenyen et gitt antall ganger, kan hjelpe læringen.

9.2.5 Feilmeldinger

Det ble oppdaget to problemer med feilmeldinger under brukertestene. Feilmeldingene er ment å hjelpe brukeren og blir derfor vurdert etter deres evne til å gjøre nettopp dette. Som nevnt i seksjon 8.2.2 hadde én bruker problemer med forstå feilmelding under registrering av sau, men deltakeren forstod heller ikke betydningen av "Sauer & lam", som senere ble endret til "Totalt". Vi forstår resultatene slik at det var betydningen av "Sauer & lam" som var problematisk da resterende feilmeldinger ledet deltakeren på rett vei. Det andre problemet vi oppdaget var at feilmeldingen på en av registreringsskjermene ikke var synlig fordi siden kunne scrolles og feilen befant seg på toppen av siden. Da deltakeren trykket på knappen for å fullføre registrering fikk den ingen respons og trengte tid til å forstå at skjermen måtte scrolles oppover for å finne feilen. Dette er en direkte glipp i brukervennlighet. Det finnes flere alternativer til hvordan dette kan løses, to av dem er som følger: enten kan det være en fast posisjon for feilmeldinger øverst i skjermbildet slik som i registrering av sau (se figur 5.30) eller så kan det automatisk scrolles opp til feltet hvor feilen er markert når man trykker på "fullfør

registrering”. Utover de to nevnte problemene hjalp feilmeldingene brukerne til å rette opp feil.

9.2.6 Totalvurdering av resultater

Med bakgrunn i at alle deltakerne fikk til å løse oppgavene de ble gitt, uten noen form for forklaring av grensesnittene, er vi svært fornøyde med gjennomføringene. Vi ser stor verdi i det som kom frem av brukertestene, fordi vi avdekket hvordan applikasjonsbrukere tenker, noe som er umulig å finne ut på egenhånd når man har utformet applikasjonene selv. Forslag til mulige løsninger på det som kom frem av testene diskuteres i seksjon 9.8.3.

9.3 Taleregistrering

For taleregistrering vurderte vi i all hovedsak de to følgende typene:

- **Dialogbasert taleregistrering** der enheten stiller spørsmål og brukeren svarer med tall
- **Kommandobasert taleregistrering** der brukeren sier en forhåndsdefinert kombinasjon av ord og tall

Et eksempel på dialogbasert registrering er: ”Hvor mange sauer og lam?” ”4”, ”Hvor mange lam?” ”2”, ”Hvor mange hvite?” ”4”, ”Hvor mange gule øremerker?” ”4”, ”Hvor mange røde slips?” ”2”. En av fordelene med denne taleregistreringstypen er at den er svært enkel å forstå, og at den ikke krever noen form for forkunnskaper. En annen fordel er brukervennligheten den gir ved at brukeren registrerer én telling om gangen, fordi eventuelle feiltolkninger oppdages med én gang, og fordi systemet kan gi tilbakemelding ved å lese svaret tilbake eller ved å fortelle at svaret ikke ble forstått. Ulempen med dialogbasert registrering er at den kan være tidkrevende fordi den alltid stiller alle spørsmålene basert på hva som er definert av sauebonden i webapplikasjonen, noe som sjelden er relevant fordi det er uvanlig å se alle ull-, øremerke- og slipsfarger samtidig. I tillegg går det med ekstra tid under lytting til, og tolkning av, svaret, samt stilling av hvert spørsmål.

Ulempen med dialogbasert taleregistrering gav opphav til alternativet med kommandobasert taleregistrering. Det samme eksempelet kan for kommandobasert registrering være: ”4 total, 2 lambs, 4 white, 4 yellow eartags, 2 red ties”. Fordelen med denne typen er effektivitet fordi brukerne kan si utelukkende det de faktisk ser, og slipper å bruke tid på det de ikke ser. Den største ulempen med kommandobasert taleregistrering er at den overlater brukeren på egenhånd ved at den forutsetter forkunnskaper om hvordan taleregistreringen fungerer, og at brukeren husker alle kommandoordene. En annen utfordring er hvordan feiltolkninger skal håndteres på en brukervennlig måte. Det kan være utfordrende å gi presise tilbakemeldinger, for eksempel dersom både antallet og kommandoordet

bak tolkes feil.

I en ideell verden ville vi ha påstått at kommandobasert taleregistrering er det beste alternativet, men det var tre tungtveiende motargumenter; (1) bruk av tale i fagområdet er helt nytt, (2) vi ønsket ikke å stille krav til brukeren, og (3) vi ville sørge for å ha utviklet nok av systemet til å gjennomføre testing av taleregistrering integrert i dette. Derfor valgte vi å gå for dialogbasert registrering. Denne er programmeringsmessig relativt rett frem å implementere, og vi kunne dermed få mulighet til å brukerteste dette skikkelig, og på den måten finne ut om bruk av tale er hensiktsmessig for fagområdet. Dersom svaret ble ja, ville dette bane vei for en videreutvikling av taleregistreringen, kanskje spesielt da å undersøke kommandobasert registrering nærmere. Resultatene ble positive, og våre tanker om videreutvikling beskrives i seksjon 9.8.

9.4 Sauebønder som marked

En diskusjon eller vurdering av sauebønder som marked er relevant til oppgaven da dette indikerer om en kommersiell videutvikling av systemet kan være lønnsom, eller om systemet må bli et statlig initiativ. Som vist i seksjon 2.2 er kostnadene for innkjøp av sporingssystemer på mellom ett til to tusen kroner per enhet og deretter rundt hundre til tre hundre kroner i drift hvert år per enhet. Inntektene en sau gir i løpet av ett år kommer fra ull, subsidier fra staten, og livdyrsalg (salg av levende dyr) eller slakt (se seksjon 2.2.6). Tabell 9.1 viser et eksempel på inntektene ei søye kan generere i løpet av ett år, medregnet to lam og slakt av disse. Merk at dette bare er et eksempel og at det er en rekke faktorer som påvirker inntektene. Ei søye får oftest ett kull med tvillinglam hvert år¹⁰. En gjennomsnittlig bestand er på 52 vinterfødte søyer¹⁰, dette antallet brukes videre i talleksempler. Vi ser av tabellen at inntekten som genereres av ei søye kan være 6 060 kroner, om vi ganger dette med 52 får vi 315 120 kroner. Sauebønder har også faste utgifter som strøm til fjøs, vinterfôr og vedlikehold av fjøs og maskiner, i tillegg til uventede utgifter som tap av sau til terreng eller rovdyr. Innkjøp av sporingssystemer uten økonomisk støtte vil føre til en oppstartsutgift på mellom 50 tusen til 100 tusen kroner om alle 52 søyer skal spores. Siden dette er en betydelig sum er det et alternativ å spore en andel sauer, da ned til en fjerdedel av bestanden, men dette gir dårligere oversikt. Videre vil det koste 5 200 til 12 400 kroner i årlig drift av systemene gitt at alle 52 søyer spores og ingen enheter går tapt. Argumentet for bruk av sporingssystemer er at det potensielt kan redusere arbeidet for bonden, øke dyrevelferden i form av mer tilsyn og minke frafallet av dyr på beite. Markedet for produkter til sauebønder er altså trangt med mange konkurrenter (seksjon 2.2) og lite kjøpekraft hos bøndene. Det er til nå gjort ett forsøk på en kommersiell mobilapplikasjon for oppsyn kalt Beitesnap (se seksjon 2.3.1), men den ble nedlagt i 2020 av økonomiske årsaker [7, 9].

¹⁰<https://snl.no/sau>, 20.05.2022

Produkt	Inntekt (NOK)
Ull	300
Ull fra 2 lam	200
Slakt av 2 lam	1 900
Subsidier for 3 sau	2 955
Subsidier for 3 sau på utmarksbeite	705
Sum	6 060

Tabell 9.1: Eksempel på årlig inntekt generert av en søye

9.5 Valg rundt utforming av systemet

9.5.1 Verdien av valg

Som beskrevet i seksjon 2.3 er et flertall av tidligere løsninger svært like på enkelte punkter og brukertesting av disse ga blandede resultater. Vi valgte å gå vår egen vei på flere av disse punktene. Det er hovedsaklig to grunner til at vi gjorde dette. Den første grunnen er at vi hadde troen på løsningene vi så for oss. Den andre grunnen er at det for feltet som helhet ligger høyere verdi i å prøve noe nytt, selv om det viser seg å gi dårlige resultater. Eksempelvis valgte det store flertallet en løsning hvor en registrering blir plassert i midten av skjermen med en form for fast plassert sikte for å indikere dette, mens vi valgte å bruke langt press på kartet for å plassere en observasjon. Tidligere er dette kun gjennomført av Styve i 2021 [10], da med negative resultater. Dette betydde at resultatene vi ville få fra brukertestene, uavhengig om de er positive eller negative, ville ha høyere verdi fordi de vil tilføre mer til feltet enn å gjenta noe som er gjort mange ganger tidligere. Gode resultater vil indikere at det er noe som likevel har potensiale, mens dårlige resultater peker mot en bekreftelse av Styves resultater. I tillegg kan kart trekkes frem som et eksempel. Vi er de første som bruker metoden for definering av kartutsnitt. Med metoden fikk vi relativt svake resultater på brukertestene. Med denne kunnskapen, og kunnskapen fra tidligere prosjekter, vet vi at programvaren kan forbedres ved å implementere kartdefineringsmetoden de fleste har brukt med forutsigbare resultater.

9.5.2 Strømlinjeutforming og brukerprivilegier

Systemet vi har utviklet består av en mobilapplikasjon og en webapplikasjon. I utgangspunktet er det mulig å lage hele systemet i én mobilapplikasjon slik det har blitt gjort en del ganger tidligere (se seksjon 2.3). Måten vi har fordelt systemets funksjonalitet på i webapplikasjon og mobilapplikasjon, gir derimot noen fordeler.

Det er konseptuelt to roller man kan ha som bruker av systemet ("oppsynsperson" og "sauebonde") selv om det teknisk sett ikke er noen forskjell mellom dem. Hvilken rolle man har avhenger av hvilken plattform man bruker. Ved å bruke

nettsiden er man sauebonde og har muligheten til å administrere egen gård og eget oppsynspersonell. Om man bruker mobilapplikasjonen er man oppsynspersonell for alle gårdene man har blitt lagt til i. Dersom man ikke eier egen gård trenger man aldri å logge inn på webapplikasjonen. Mobilapplikasjonen er med andre ord isolert til å omfatte kun gjennomføring av oppsynsturer.

Det er lagt opp til at bruken av mobilapplikasjonen skal være så strømlinjeformet som mulig, såkalt ”plug and play”. Systemet minimerer mengden informasjon om oppsett av applikasjonen sauebonden trenger å kommunisere til sitt oppsynspersonell. Det eneste oppsynspersonell må gjøre er å opprette en brukerkonto og gi beskjed til sauebonden når dette er gjort, slik at sauebonden kan legge til kontoen som oppsynspersonell. Om oppsynspersonell skulle ha definert eget kart for beiteområdet kunne dette ha ført til feil kartutsnitt. Vi mener at sauebonden kjenner beiteområdet og flokken sin best selv og at det derfor er best at bonden definerer dette selv.

9.6 Refleksjon over fremdrift

Under forprosjektet (se vedlegg A) satt vi en overordnet tidsplan med milepæler som vi løst skulle navigere etter. I denne seksjonen ser vi på hvordan prosjektet faktisk utartet seg sammenlignet med planen, og reflekterer litt rundt dette. Planen var som følger:

- **Sette opp arbeidsmiljø** 10. januar
- **Design** 11.-14. januar
- **Smidig utvikling** 15. januar-1.mai
- **MVP** 1. april
- **Ferdigstilling av rapport** 2.-19. mai
- **Bufferperiode** 20.-31. mai
- **Leveranse** 1. juni

Vi har i grove trekk klart å følge planen. De planlagte periodene for oppsett av arbeidsmiljø og design ble fulgt. Videre ble det ikke utviklet ny funksjonalitet etter 29. april, så utviklingsperioden var hovedsaklig innenfor planen selv om det ble gjort små visuelle endringer i brukergrensesnittet og fiksing av noen bugs i løpet av mai. Vi hadde som milepæl å ha en MVP¹¹ klar 1. april. For denne skulle det være mulig å fullføre en saueregistrering, fra å angi posisjon i kart til å registrere både med og uten tale. MVP-en var klar allerede 17. februar, godt innenfor planen. Planlagt periode for ferdigstilling av rapport ble lengre enn planlagt, og vi benyttet oss av hele den påfølgende bufferperioden. Vi leverte prosjektet 6. juni, noen dager etter milepælen 1. juni.

Selv om utviklingsperioden i stor grad var innenfor planen, var det her spesielt

¹¹Minimum Viable Product

to ting som stjal mye tid. Den første var review på pull requests, som beskrevet i seksjon 6.1.6. Det andre som stjal mye tid var testing av programvaren, som dermed ble nedprioritert fra 10. mars til fordel for utvikling av funksjonalitet. Videre brukte vi i utviklingsperioden mindre tid enn planlagt på å skrive rapport parallelt med programmering. Dette henger sammen med at vi ville utvikle et komplett og brukervennlig system som skulle oppfylle hele kravspesifikasjonen, og gjerne litt til. Rapportskriving ble med andre ord prioritert lavere enn utvikling på daværende tidspunkt. I tillegg ble det gjennomført brukertester innenfor den planlagte perioden for ferdigstilling av rapport, som tok mye tid og gjorde det urealistisk å bli ferdig med rapporten innenfor denne perioden.

9.7 Mangler og kjente feil

Mangler i denne seksjonen regnes ikke som mangler mot kravspesifikasjonen, men mangler ut ifra hva som kan forventes av et komplett system på et generelt plan. Prosjektet er en del av pågående arbeid som utføres og utvikles med årene, tidligere arbeid beskrives i seksjon 2.3. Vi ønsket å lage et system som har gjennomgående komplett funksjonalitet og kan settes i produksjon. Med dette menes at kjernefunksjonaliteten er på plass, altså at det er mulig å gå oppsynsturer, registrere hva man ser, se oversikt over gjennomførte turer og generere en årsrapport basert på disse. På bekostning av dette ble funksjonalitet som ble sett på som åpenbar nedprioritert til fordel for mer nyskapende funksjonalitet. Blant funksjonalitet som selvfølgelig bør være en del av systemet før det kan settes i produksjon er muligheten til å slette nedlastede kart, endre passord og avslutte brukerkonto. Det skal dog nevnes at i et "happy day scenario" der det ikke er behov for slike handlinger kommer systemet fullendt ut, når det kommer til kjernefunksjonaliteten som er beskrevet tidligere i avsnittet.

9.7.1 Liste over kjente feil

Videre følger en liste over kjente feil eller "bugs" i programvaren:

- Webapplikasjonen tvinger ikke sauebonden til å lagre navn på gården. Det gir feil i mobilapplikasjonen.
- Gårdsnavn er ikke unike. Dersom én bruker er oppsynsperson for to gårder med likt navn, vil dette gi feil i mobilapplikasjonen fordi nedtrekksmenyen ikke kan ha like alternativer.
- På Android 10 (hvertfall på Galaxy S9+) er det en feil der knappene for å fullføre sauregistrering og starte/fortsette taleregistreringsdialog tidvis ikke dukker opp igjen når telefonen ikke registrerte taleinput.
- Instruksjon i webapplikasjonen om at sørøstlig hjørne må være sørøst for nordvestlig hjørne vises dersom man holder inne på kartet etter at hele området er definert.
- I mobilapplikasjonen blir posisjonen til enheten registrert som null om man holder inne på kartet rett etter at det har åpnet seg.

- Om man i mobilapplikasjonen endrer gård og/eller kart under nedlastning av kart vil det stå at det kartet som sist er endret til er kartet som er lastet ned.

9.8 Utbedringer og videre arbeid

For videre utvikling av kodebasen med mål om å lage et komplett produkt anser vi det som høyst prioritert å utvikle funksjonaliteten og rette opp i feilene beskrevet i seksjon 9.7. Med mål om videre forskning og ytterligere effektivisering av oppsynsturer bør taleregistrering utbedres først, som diskuteres i seksjon 9.8.1, etterfulgt av kombinasjon av systemer som diskuteres i seksjon 9.8.2. Til slutt følger en liste av ytterligere funksjonaliteter som vil utbedre programvaren.

9.8.1 Taleregistrering

Videre arbeid vil for dialogbasert taleregistrering være å undersøke hvordan denne fungerer på andre operativsystemer enn Android (som er det vi har testet på), spesielt med tanke på lyttetid og statuslyder. Det vil også være naturlig å videreutvikle talljusteringsfilteret som brukes, både for å oppdage og legge til flere feiltolkninger, men spesielt ved å legge til flere tall siden filteret kun går fra null til tolv. I tillegg kan dialogen videreutvikles ved å legge til talekommandoen "next" for å gå til neste spørsmål, per nå er man nødt til å svare for å komme til neste spørsmål. Det er mulig å effektivisere dialogen ved ta i bruk en annen talekommando, eksempelvis "skip", for å hoppe over de resterende spørsmålene i en underkategori (ull-, øremerke- eller slipsfarge). Dette kan brukes dersom det for inneværende registrering ikke er flere farger i kategorien tilstede enn det som alt er registrert. For å unngå å se på skjermen vil det også være interessant å se på mulighetene for å starte og gjenoppta dialogen ved bruk av en tredje talekommando, som skal fungere på samme måte som "Hey Google" og "Hey Siri" for taleassistentene til henholdsvis Google og Apple.

En annen måte å forbedre dialogen på er å gi den et snev av intelligens, slik at den kan forstå at det ikke er noe poeng i å stille flere spørsmål. Et eksempel på dette er når brukeren har svart at det er to sauer og lam totalt og ett lam. Så fort summen av ullfarger er to kan dialogen hoppe over resterende ullfarge-spørsmål, fordi det ikke kan være flere ullfarger enn det er sauer og lam. Det samme gjelder for øremerker og slips, men for slips kan dialogen hoppe over resterende slips-spørsmål dersom summen av slipsfarger blir lik antall søyer, fordi lam ikke har slips. Dette vil gjøre dialogbasert registrering mer effektivt.

Som nevnt i seksjon 9.3, er det naturlig å se videre på kommandobasert taleregistrering da den dialogbaserte var en suksess. Kommandobasert taleregistrering må kunne tolke farger, og det er dermed relevant å ta i bruk justeringsfilteret for farger som vi utviklet i forprosjektet (se vedlegg A). Her er det også mulig

å kombinere registreringstypene, eksempelvis ved å ha dialogbasert registrering for antall sauer og lam og deretter åpne for kommandobasert registrering for de resterende kategoriene (ull-, øremerke- og slipsfarger). Det kan tenkes at dette hadde blitt for mye for oppsynspersonen å holde kontroll på og huske si på en gang. I tillegg hadde ikke oppsynspersonen nødvendigvis visst om applikasjonen valgte nær- eller avstandsregistrering uten å se på skjermen, og dermed ikke om øremerke- og slipsfarger skal være en del av registreringen. Vi ser derfor for oss at det er en god idé at enheten stiller spørsmål per underkategori, og at man svarer kommandobasert. Et eksempel på dette er at enheten sier ”Registrer ullfarger” og brukeren svarer ”two white one black”, med tilsvarende måte for øremerke- og slipsfarger. Ellers vil det være relevant å se på måter å implementere tale for de andre registreringstypene (skade, kadaver og rovdyr). Det kan godt tenkes at det finnes flere gode alternativer å bruke tale på enn de som diskuteres i denne seksjonen.

9.8.2 Kombinasjon av systemer

Vi ser et stort potensiale i å kombinere systemene som er gjennomgått i kapittel 2, spesielt sporingssystemer og systemer for oppsynsturer siden disse er så tett knyttet. Ved å integrere sporingssystemene inn i en applikasjon for oppsynsturer kan turen i høy grad effektiviseres. Oppsynspersonell vil da kunne tilpasse ruten som gås etter hvor dyrene faktisk befinner seg, som eksempelvis kan vises med heatmaps i kartet. Vi ser også en mulighet for at grupperingene av sau kan være forhåndsmarkert i kartet og ferdig innfylt som observasjoner oppsynspersonell kun trenger å bekrefte. En potensiell ulempe med dette er at det er enklere for oppsynspersonell å slurve eller være unøyaktig, men samtidig er dette også mulig med dagens systemer da kvaliteten av registreringer er den oppsynspersonell legger i den. Videre er det også mulig å integrere melding av rovdyrobservasjoner slik at disse direkte videreføres til myndighetene, og å utvide for flere typer beitedyr som i Beitesnap (seksjon 2.3.1) og flere av posisjonssystemene.

9.8.3 Løsningsforslag på det som ble avdekket av brukertester

Her er en liste over forslag til løsninger på mindre problemer som ble avdekket av brukertestene (seksjon 8.2.2).

- Gi verbal beskjed når taleregistreringsdialog er gjennomført
- Mulighet til å avbryte nedlasting av kart
- Forbedre brukervennligheten for kartdefinering. Helst noe som er selvforklarende og gir mulighet til å redigere og angre

9.8.4 Ytterligere utbedringer

Videre følger en liste av ytterligere utbedringer tatt i korthet. Listen kan suppleres med punkter fra seksjon 2.4 som ikke er implementert, samt de feilene og den funksjonalitet som er å forvente av et system som nevnt i seksjon 9.7.

- Dokumentasjon av koden med doc-kommentarer¹²
- Forhindre brute-force-angrep
- Firebase App Check, for å kun tillate Firebase-forespørsler fra egne applikasjoner¹³.
- Verifisere at telefonnummer og e-postadresse tilhører personen som oppretter brukerkonto
- Kartmarkør på oppsynsturenes start- og sluttposisjon
- Dynamiske kartmarkførfarger. For eksempel gule markører på saueregistreringer med færre lam enn slips(ene) tilsier
- Slette eller redigere en registrering under en oppsynstur
- Kompass på kartskjerm med mulighet for automatisk nord-orientering
- Registrere om skadd sau er sau eller lam
- Sjekke at oppsynsperson fortsatt er registrert som oppsynsperson hos gården før en (arkivert) oppsynstur lastes opp
- Innstilling for å bestemme hvilken avstand som gir avstandsregistrering og/eller la brukeren bytte mellom disse på registreringskjermen
- Varsle sauebonde om at oppsynsturer er fullført når de gjennomføres av noen andre
- Aktiv databruk i visning av oppsynsturer, som Halvorsen i 2019 [4]
- Taleregistrering av skadd sau, sauekadaver og rovdyr
- Håndtere avvist GPS-tillatelse på mobil enhet
- Forbedre nøyaktigheten til GPS-posisjonering
- Redigere definerte beiteområder
- Arealbegrensning på definering av beiteområde
- Mulighet til å markere faste punkter i definert kartutsnitt, som saltsteiner o.l
- Søke opp steder i kart ved definering av beiteområde
- Redusere antall lesinger fra Cloud Firestore tilknyttet årsrapport-siden
- Filtreringsmuligheter ved visning av oppsynstur (datoer, beiteområder osv.)
- Knytte øremerker som ikke tilhører egen gård til et gårdsnavn
- Autofokus på tekstfelt ved definering av beiteområde og oppsynspersonell
- Sørge for at gårdsnavn er definert før oppsynspersonell defineres
- Ikke gi mulighet til å velge øremerke- eller slipsfarge som allerede er definert
- Tydeligere markering av feil på defineringssider
- Tilpasse webapplikasjon til bruk i mobil nettleser
- Vise den valgte gårdens adresse i mobilapplikasjonen
- Forbli innlogget etter at applikasjonene avsluttes
- Verifisere at gårdsnummeret som skrives inn i webapplikasjonen faktisk tilhører gården

¹²<https://dart.dev/guides/language/effective-dart/documentation>, besøkt 18.05.2022

¹³<https://firebase.google.com/docs/app-check>, besøkt 30.05.22

9.9 Kritikk av oppgaven

Siden oppgaven gjennomføres flere ganger med den samme grunnfunksjonaliteten (eksempelvis nedlastning av kart, markering av registreringer i kart, logging av posisjon osv.) bør kjerneprogramvaren utvikles slik at den kan brukes om igjen og bygges videre på i hver iterasjon av masteroppgavene. På den måten kan det fokuseres på de mer nyvinnende delene av programvaren som for eksempel registrering uten å se på skjermen ved bruk av sveiping eller tale. Samtidig vil dette være problematisk fordi den teknologien som brukes vil kunne begrense hvilke valg som kan gjøres i fremtidige prosjekter, og at det på et tidspunkt sannsynligvis må gjøres et større arbeid for å oppdatere kjerneprogramvaren. I tillegg kan det begrense potensiell videreutvikling av kjerneprogramvaren i form av brukergrensesnitt og programflyt.

Kapittel 10

Konklusjon

10.1 Drøfting av resultater

Med utgangspunkt i oppgavebeskrivelsen og formålene beskrevet i kapittel 1, og kravspesifikasjonen i kapittel 3, mener vi at oppgaven har blitt løst på en tilstrekkelig måte.

10.1.1 Effektmål

Systemet når effektmålene i seksjon 1.4.1. Basert på resultatene fra brukertest kan systemet vurderes som bedre egnet enn penn og papir. På oppsynstur gir bruk av datateknologi presise registreringer og mer data for mindre innsats. For sauebonden automatiserer teknologien systematisering av gjennomførte oppsynsturer og produksjon av årsrapport. Brukertestene viste at talegjenkjenningsteknologi er hensiktsmessig å bruke da deltakerne fullførte registreringer uten å se på skjermen.

10.1.2 Læringsmål

Prosjektmedlemmene har gjennom prosjektperioden tilegnet seg kunnskaper om hold og tilsyn av sau. Videre har de fått erfaring med å sette seg inn i et problem fra et ukjent domene og samarbeide seg frem til en løsning på dette. I tillegg har de fått erfaring med å vurdere et selvutviklet system ved å utforme en prosedyre for brukertest og gjennomføre denne. Under utvikling av systemet økte de egne programmeringskunnskaper gjennom utvikling mot skytjeneste og bruk av moderne frontend-rammeverk. Medlemmene har med andre ord oppnådd læringsmålene i seksjon 1.4.2.

10.1.3 Resultatmål

Kravene til leveranse i seksjon 3.2.3 er oppnådd ved at (1) systemets kodebase ligger på GitHub (<https://github.com/rebner-mork/master-thesis>), (2) kapittel

5 er å regne som bruksinstruksjoner og (3) denne rapporten beskriver, vurderer og diskuterer prosjektet i sin helhet.

10.2 Konklusjon

Vi ønsket med dette prosjektet å effektivisere det manuelle oppsynet av sau på beite ved å digitalisere arbeidsoppgavene. Et system av en mobilapplikasjon og en webapplikasjon ble utviklet. Mobilapplikasjonen effektiviserer oppsynet ved å automatisk tilføre posisjonsrelatert data til oppsynsturen slik at oppsynspersonen kun behøver å registrere hva han eller hun ser. I tillegg benytter den talegjenkjenningsteknologi, som gjør det mulig å registrere sau uten å se på skjermen. Når oppsynsturen er gjennomført, sendes data til webapplikasjonen. Webapplikasjonen gir systematisk oversikt over gjennomførte oppsynsturer, og automatiserer generering av en årsrapport som skal sendes til myndighetene. Det ble gjennomført brukertesting av hele systemet for å undersøke om det var brukervennlig. Basert på resultatene mener vi at vi har utviklet et brukervennlig system som viser at digitale verktøy effektiviserer, og kan øke kvaliteten på, både oppsyn og etterarbeid, som resulterer i at sauebonden får mer tid til andre oppgaver.

Bibliografi

- [1] S. Dysthe og A. Kjerstad, «Effektivisering av manuell oppfølging av sau på beite,» *NTNU Open*, s. 1–198, 2018.
- [2] T. Alm og A. O. Gård, «Manual Follo-up of Sheep on Free Range Pasture,» *NTNU Open*, s. 1–162, 2018.
- [3] F. M. Schmidt-Hanssen, «A Mobile Application and Web System to Monitor Free-Range Grazing Sheep,» *NTNU Open*, s. 1–104, 2018.
- [4] J. A. Halvorsen, «Sheep Shepherding - Grand Overview,» *NTNU Open*, s. 1–108, 2019.
- [5] L. Svendsen, «Gjenfinning av Sau på Utmarksbeite,» *NTNU Open*, s. 1–74, 2019.
- [6] P. Flytøren, «Manual monitoring of grazing sheep,» *NTNU Open*, s. 1–82, 2020.
- [7] K. D. Abtahi og T. V. Gjeldseth-Borgen, «Applikasjon for effektivisering av tilsyn med sau på utmarksbeite,» *NTNU Open*, s. 1–152, 2021.
- [8] J. E. Carlsen, «Digitalizing the Sheep Supervision Documentation Process,» *NTNU Open*, s. 1–205, 2021.
- [9] M. C. Hyll og T. M. Torgersen, «Mobilapplikasjon for forenkling av manuelle tilsyn av sau på beite,» *NTNU Open*, s. 1–98, 2021.
- [10] S. O. Styve, «Evaluating the Utility of Using a Mobile Application for Data Registration of Grazing Sheep on Open Pastures,» *NTNU Open*, s. 1–74, 2021.
- [11] Firebase. (2022), adresse: <https://firebase.google.com/> (sjekket 18.05.2022).
- [12] C. Baham og R. Hirschheim, «Issues, challenges, and a proposed theoretical core of agile software development research,» *The Information System Journal*, s. 1–27, 2021.
- [13] Flutter. (2022), adresse: <https://flutter.dev/> (sjekket 18.05.2022).

Vedlegg A

Forprosjekt

Vedlegget inneholder linken til en forprosjektrapport som forfatterne utførte i forkant av dette masterprosjektet. Formålet med forprosjektet var enkelt forklart å undersøke om taleregistrerings-teknologi har kommet langt nok til å være hensiktsmessig å bruke i domener som saueoppsyn.

https://studntnu-my.sharepoint.com/:b:/g/personal/anderhre_ntnu_no/EdAlgQ7AZMJHnQkfEPEi2D8BtT3pLSwLeXzUAdRax0fw0Q?e=ujGLEm

Vedlegg B

Papirprototype webapplikasjon

Vedlegget inneholder papirprototypene av webapplikasjonens brukergrensesnitt, som ble tegnet i prosjektets startfase og som ga grunnlaget for utviklingen.

Den første figuren viser webapplikasjonens startside der man skal kunne logge seg inn eller opprette brukerkonto.

Den andre figuren viser den første av tre faner som vil dukke opp etter innlogging. I denne fanen er det tenkt at man skal få en grov oversikt over alle turene som har blitt gått, og at man ved å klikke på en av dem skal kunne se detaljert informasjon om oppsynsturen. Den detaljerte informasjonen er tenkt å inkludere et kart som viser ruten som ble gått, markører ved hver registrering og stiplede linjer som viser hvor registreringene ble gjort fra. Videre skal den vise metadata som tidspunkt for turen og hvem som gjennomførte den, samt tabeller som gir oversikt over antall registreringer per øremerkefarge, slipsfarge og ullfarge.

Tredje figur viser fanen "Årsrapporter", der det skal være mulig å få lastet ned en rapport over alle oppsynsturer som ble gått for et valgt år.

Fjerde figur viser fanen "Min side", der man skal kunne lagre informasjon om gården og definere øremerker, slips og oppsynspersonell.

Oppsynsturerweb



informasjon om
hva oppsynsturer
er og hvordan
denne nettsiden
kan gjøre det
enklere

bruker navn

Logg inn

Glemt passord

Opprett konto

Oppsynsturer

Årsrapporter

Min Side

2019
2020
2021

April

Mars

07

14

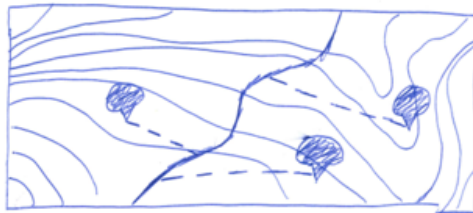
21

28

Mai

Juni

2022



Dato: 07. 06. 2019

Tid: 13:00 - 17:00

Gått av: Steinar Savenmann

Øremerker	Antall
Rød	7
Blå	19
Gul	152

Sau	Lam	Total
-----	-----	-------

Oppsynsturer

Årsrapporter

Min Side

Litt info om årsrapport

Velg år ▾

Last ned årsrapport

Oppsynsturer

Årsrapporter

Min Side

Meg

Personopplysninger

Gunnar Haukeblakk
mm@mm.com • 900 00 000

Fjern

Svein Ullfull

mm@mm.idk • 123 45 078

Fjern

Min Gård

Gårdsdata

Øremerker

Slips

Oppsynspersonell

Legg til

Innstillinger

Personvern

Vedlegg C

Papirprototype mobilapplikasjon

Vedlegget inneholder papirprototypene av mobilapplikasjonens brukergrensesnitt, som ble tegnet i prosjektets startfase og som ga grunnlaget for utviklingen.

Første rad med figurer viser sidene for innlogging og starting av oppsynstur, der det for den sistnevnte er tenkt at man velger hvilken gård man skal gå oppsyn for, og i hvilket kartområde.

Andre rad med figurer viser hovedsidene i applikasjonen. Til venstre vises kart med enhetens posisjon, stien man har gått, burgermeny og fargede symboler med ikoner som symboliserer registreringstype og alvorlighetsgrad. Et touch på kartet er tenkt å føre brukeren videre til registrering av sauer på kartposisjonen. Knappen nede til venstre viser antall sauer og lam som er registrert til nå. Denne er tenkt å åpne en enkel rapport med hva som er registrert, se som vist i den midterste tegningen. Knappen nede til høyre kan brukes til å registrere sau, men også andre registreringstyper, se tegningen til høyre.

Tredje figurrad viser de ulike sidene for registrering, nærmere bestemt registrering av sau, skade, død, rovdyr og notat.

Fjerde rad med figurer viser sidene for opplasting av oppsynstur, både med og uten nettverk.

Femte og siste figurrad viser de tenkte innstillingene auto-dialog (dialog som starter automatisk ved registrering) og les tilbake (det som ble tolkes leses tilbake før neste spørsmål stilles).

Hand-drawn login screen with the following elements:

- Input field for "Brukernavn" (Username)
- Input field for "Passord" (Password)
- Buttons for "Logg inn" (Login) and "Ny bruker" (New user)

Hand-drawn profile screen with the following elements:

- Home icon and menu icon in the top left and right corners.
- Profile icon and "Navn" (Name) field.
- "Gård" (Farm) field with a dropdown arrow.
- "Kart" (Map) field with a dropdown arrow.
- "Start oppsynstur" (Start inspection tour) button.

Hand-drawn map selection screen with the following elements:

- Title: "Velg kart" (Select map)
- Three map thumbnails, each with a wavy line representing a map.
- "Legg til kart" (Add map) button at the bottom.



Hand-drawn registration statistics screen with the following elements:

- Title: "Registrert" (Registered)
- Statistics:
 - 🐑 Sauer: 36
 - 🐑 Lam : 15
- Horizontal dashed line separator.
- Statistics:
 - 🟡 Mine : 40
 - 🟠 Mads : 10
 - 🟢 Blå : 1
- Bottom right: "+?" icon.

Hand-drawn registration list screen with the following elements:

- Title: "Registrer" (Register)
- List of registration types with icons:
 - 🐑 Sauer
 - 🐑+ Skade/syk
 - ☠️ Død
 - 🐾 Rovdyr
 - 📄 Notat
- Bottom left: "X" icon.
- Bottom right: "+?" icon.

Registrering sau

Antall

Sauer

Lam

Hvite

Svarbe

Svart hode

Slips

Røde

Blå

Gule

Øremerker

Røde

How mange sauer?

 Registrer

Registrer skade

Øremerke


Slips

Moderat Alvorlig

○ ○

Årsak

Notat

 Registrer

Registrer død

Øremerke

Slips

Årsak

Bilde

Notat

 Registrer

Registrer rovdyr

Ulv	Jerv
Gaupe	Biørn

Antall

1 2 3


Unge

1 2 3

Notat

 Registrer

Registrer notat

 Registrer

 Registrer

Ferdig?

Ja Nei

Send Oppsynstur Fortsett oppsynstur

Oppsynstur sendt

Sendt

Ingen nettverk

Synkroniser når du er tilkoblet

Innstillinger

Auto-dialog Av På

Les tilbake Av På

Mine kart

Vedlegg D

Årsrapport

Vedlegget inneholder et eksempel på en årsrapport som er generert fra webapplikasjonen, som beskrevet i seksjon 7.8.

Årsrapport 2022

Gård	Farmen
Gårdsnummner	00496188
Adresse	Bondeveien 3
Gårdseier	Ola Nordmann ola.nordmann@gmail.com 95721944
Oppsynspersonell	Anders Rebner Nikolai Mork
Antall oppsynsturer	6

Dato	Tid	Totalt	Voksne	Lam	Skadde	Døde	Rovdyr
14/03/2022	10:21 - 12:54	63	40	23	0	0	0
21/03/2022	11:14 - 13:58	61	38	23	1	0	0
28/03/2022	10:16 - 11:42	65	40	25	0	1	1
04/04/2022	10:01 - 11:51	59	29	30	2	4	3
11/04/2022	13:36 - 16:15	65	36	29	0	0	2
04/05/2022	18:02 - 18:05	57	38	19	0	1	0

Vedlegg E

Brukertest: Prosedyre

Vedlegget inneholder testprosedyren for brukertesten som ble gjennomført, som beskrevet i seksjon 8.2 og diskutert i seksjon 9.2

Introduksjon

Hei og velkommen til brukertest! Vi skal teste et system som består av web-løsning og mobil-app. Først skal vi teste web-løsning, så gå en tur rundt i Moholt Studentby for å teste appen, før vi avslutter med web-løsningen. Du vil få oppgaver å løse underveis, og det er viktig å poengtere at det er systemet som skal testes, ikke deg. Det er fint om du kan tenke høyt underveis. Én av oppgavene innebærer å se et bilde av en skadd sau og en død sau, er du komfortabel med det? Testen vil ta rundt 35 minutter.

Forklaring til domenet

Når sauer er på beite går man turer for å se til sauene, og noterer det man ser. For sau inkluderer dette antall av hver ullfarge, øremerkefarge og slipsfarge. Sauer merkes med fargede øremerker som inneholder en landskode (MT-NO), et gårdsnummer (7-8 siffer) og et individnummer (5 siffer). Sauer som har lam har gjerne et farget "slips" som symboliserer antall lam. Det er i tillegg aktuelt å registrere skadd sau, sauekadaver, rovdyr og eventuelle andre ting som kan være av interesse.



Av Anders H. Rebner



Scenario

Du er en sauebonde som skal bruke en løsning for oppfølging av sau på beite.

Del 1: Web-løsning

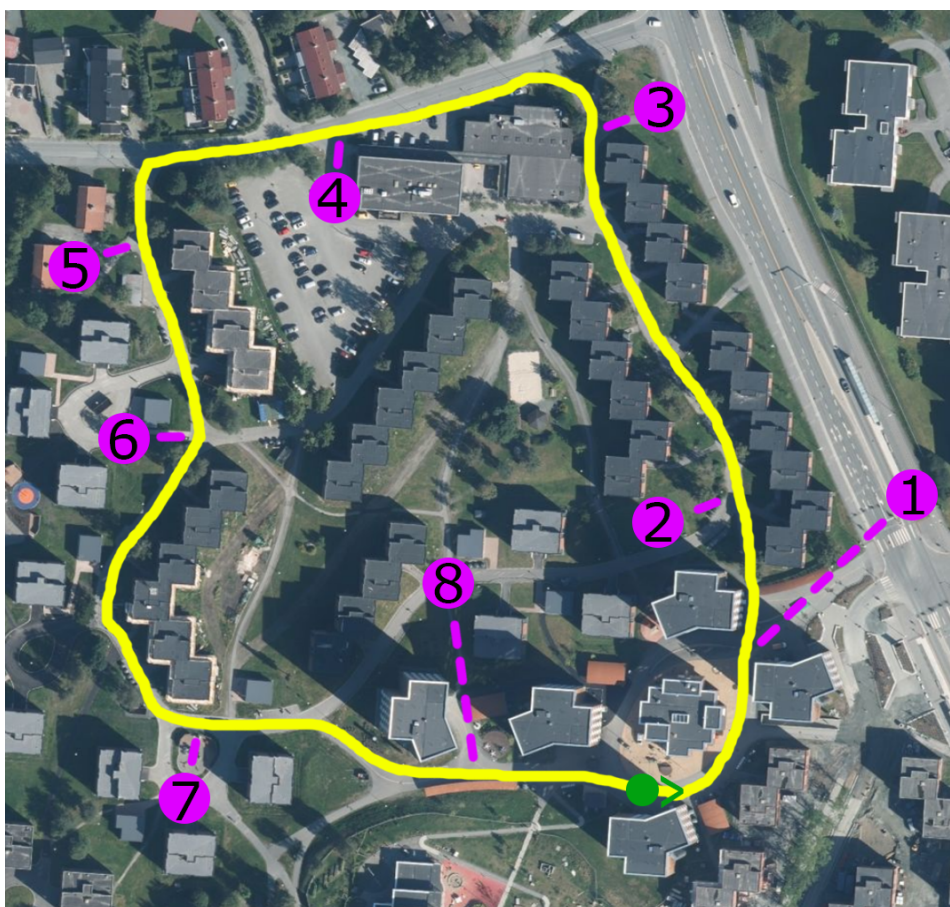
1. Legg inn Moholt Studentby som beiteområde

Del 2: Klargjør mobil-applikasjon

1. Last ned kartet

Del 3: Gjennomfør oppsynstur på mobil-applikasjon

1. Gå ruten du blir bedt om (deltaker får ikke se kartet under)
2. Utfør registreringer når du blir vist bilder



(Grønn prikk er start- og stoppunkt, gul linje er rute, lilla stiplede linjer er siktlinjer og lilla nummererte sirkler er posisjonene til registreringene på neste side med samme nummer.)

Del 4: Se gjennom oppsynstur på web-løsning

1. Se på oppsynsturen du gikk
2. Se på bildene du tok av sauekadaveret
3. Se på hva du registrerte på lekeplassen

Registreringer

De fleste bildene har CC-lisens (Creative Commons).

Bilde 4 har vi fått tillatelse til å bruke, og øremerke-bildene har vi laget selv.

1. Sauer og lam avstand, manuelt



<https://www.pigsels.com/no/public-domain-photo-zcdme>

2. Sauer og lam nær, manuelt



<https://snl.no/sau>, Sau, av Per Harald Olsen/NTNU

3. Rovdyr (ulv)



<https://www.flickr.com/photos/tambako/18039433124>,

Two brown wolves posing on the log, av Tambako The Jaguar

4. Skadd sau



<https://mre.no/har-du-tips-om-stygt-skadd-sau/19.11491>,
av Eldar Lingås

5. Kadaver



<https://www.geograph.org.uk/photo/6113287>,
Dead ewe beside the footpath near Llysty, av Jeremy Bolwell

6. Notat (ødelagt gjerde)



<https://www.geograph.org.uk/photo/485319>,
Broken gate, av Newbiggin Hall Scouts

7. Sauer/lam nærregistrering m/ dialog

Uten å se på skjermen etter påbegynt



Av Anders H. Rebner

8. Mange sauer avstandsregistrering m/ dialog

Uten å se på skjermen etter påbegynt dialog



<https://snl.no/sau>, Sauer og lam på beite i fjellet, av Per Harald Olsen/NTNU

Testroller

Rolle 1: Testleder

Testleder har hovedansvaret for å veilede deltakeren gjennom hele testen. Dette innebærer å introdusere testen, gi forklaring til domenet, samt å vise bildene som skal brukes til registrering.

Rolle 2: Observatør

Observatøren har to oppgaver. Den første er å følge med på testdeltakerens samhandling med løsningene og notere det som foregår i skjemaet på neste side. Observatøren må hele tiden kunne se skjermen. Den andre er å sørge for at alt det tekniske er klart, herunder å logge inn på de brukerkontoene som skal brukes, samt å slette de nylig definerte kartene etter hver gjennomføring.

Etter hver testgjennomføring vil begge diskutere gjennomføringen og fylle ut notatene.

Observatørskjema

Gjennomføring nr.	
Del 1	
Del 2	
Del 3	
Registrering 1	
Registrering 2	
Registrering 3	
Registrering 4	
Registrering 5	
Registrering 6	
Registrering 7	
Registrering 8	
Del 4	
Oppsummering	

Vedlegg F

Brukertest: Notater

Vedlegget inneholder notater fra hver gjennomføring av brukertesten beskrevet i seksjon 8.2 og prosedyren i vedlegg E.

Observatørskjema

Gjennomføring nr.	1
Del 1	Endret først på min gård. Fant beiteområder etterpå. Leser beskrivelsen. Prøver å lagre kun kartnavn. Leser ikke instruksjonene under. Fant dem til slutt. Restart pga. hjørne plassert feil. Forstod det og rettet opp. Fant ut at man kan zoome. Slettet forrige. Klarte fint å legge til riktig.
Del 2	Trykket på start oppsynstur istedenfor kartnedlast-ikon.
Del 3	
Registrering 1	Åpnet turoversikt og forstod innholdet. Går gjennom plussmeny. Registrerer 0 sauer og lam, 0 lam og 3 hvite. Får feilmelding. Betydning av "Sauer og lam"-feltet kom ikke godt nok frem, så det tok litt tid å rette opp.
Registrering 2	Feilmelding som følge av misforstått "sauer og lam". Ble forklart. Riktig antall øremerker.
Registrering 3	Alt ok.
Registrering 4	Holdt inne på kart uten å være i modus, havnet dermed på sauregistrering. Forstod selv at det var feil, avbrøt registreringen og valgte saueskade. Registrering ok.
Registrering 5	Alt ok, men zoomer ikke inn på kart. Tar to bilder.
Registrering 6	Alt ok.
Registrering 7	Sier "three sheep". Skjønner deretter raskt at det kun er tall som skal sies. Avsluttet tur ved uhell (bug oppdaget: trykk utenfor avslutt tur-popup avslutter). Resten av dialogen går fint (men bug der mikrofon forsvinner). Snakker før lyd, forstår at det er for tidlig.
Registrering 8	Alt ok. Dialog stopper, gjenopptas. Får feilmelding, forstod denne og rettet opp.
Del 4	Forstod at kartmarkører er klikkbare, brukte ikke tabeller på høyresiden.
Oppsummering	Kom med kritikk av sauer og lam, ble forvirret av at ullfarger vises samtidig. Foreslo at ullfarger dukker opp senere.

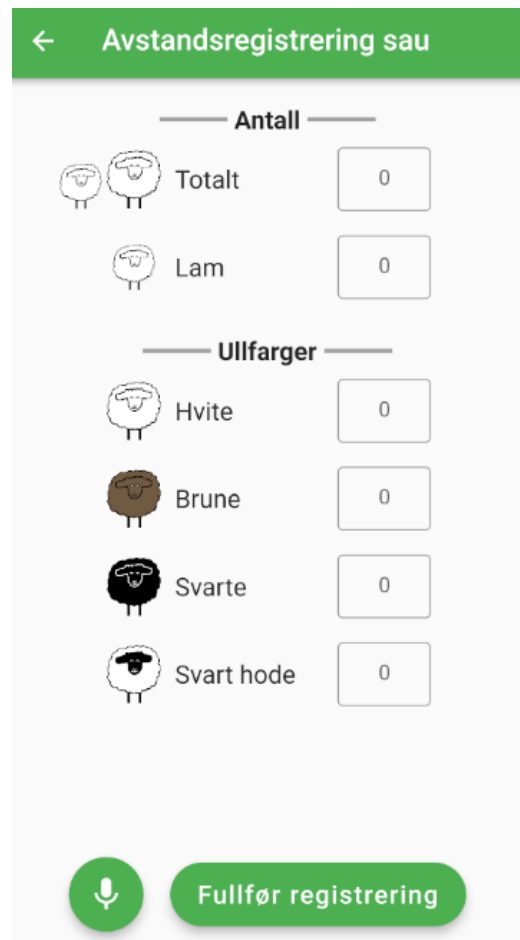
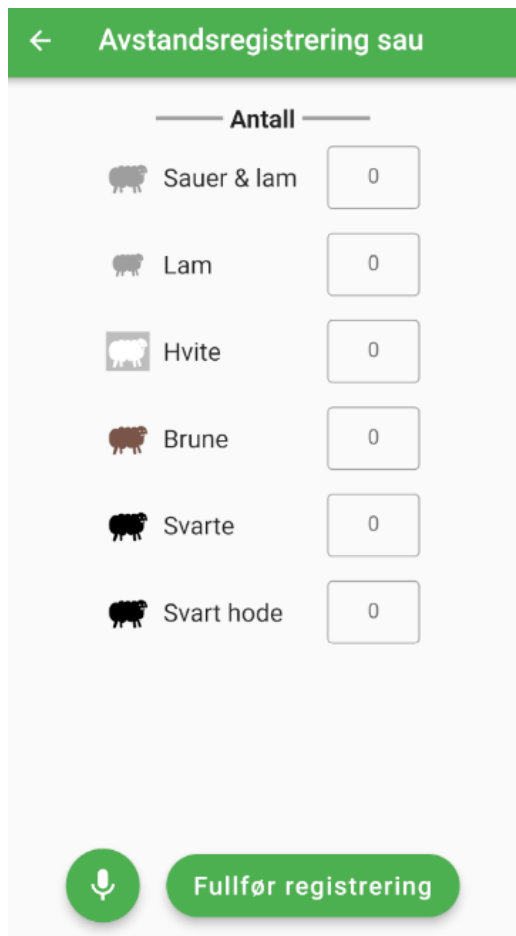
Gjennomføring nr.	2
Del 1	Alt ok. Navigerer riktig, zoomer inn, leser instruksjoner, markerer riktig område, skriver navn, lagrer.
Del 2	Alt ok. Bruker nedlast-ikon.
Del 3	
Registrering 1	Prøver å trykke på kart, skjer ingenting, holder inne på kart og kommer til registrering. Lurer på om "sauer og lam" er totalt eller ikke, kom til riktig konklusjon. Registrering ok.
Registrering 2	Alt ok. Holder inne på kart.
Registrering 3	Holder inne på kart, avbryter saueregistreringen. Finner deretter plussknapp, resten ok.
Registrering 4	Alt ok, men prøvde å zoome lengre inn, det gikk ikke.
Registrering 5	Alt ok.
Registrering 6	Alt ok.
Registrering 7	Dialog starter bra. Svarte zero på antall lam før lyttelyd ble gitt, svarte "none" neste gang, som ble feiltolket som "one". Får feilmelding etter dialog og rettet opp.
Registrering 8	Teller seg klar før dialogstart. Dialog stopper underveis, forstår hvordan den gjenopptas. Får feilmelding, forstår og fikser denne.
Del 4	Alt ok. Navigerer riktig, velger øverste tur, finner kadaverbilde gjennom kart.
Oppsummering	Var ikke redd for å trykke seg rundt på web eller app uten oppfordring. Gikk veldig fint. Lærte raskt underveis og fikk til alt. Kikket og scrollet på tabellene på høyresiden, åpnet popups gjennom kartmarkører. Oppdaget summerings-bug i appens turoversikt.

Gjennomføring nr.	3
Del 1	Endrer først på min gård. Navigerer til beiteområder og leser instruksjoner. Resten ok.
Del 2	Alt ok, brukte nedlast-ikon.
Del 3	
Registrering 1	Går gjennom plussmeny. Holder inne på kart. Stusser på "Sauer og lam" og "Lam". Finner ut av det. Resten ok.
Registrering 2	Alt ok. Går gjennom plussmeny. Prøver å zoome lengre, det går ikke.
Registrering 3	Alt ok. Går gjennom plussmeny. Ingen notat fordi ikke bemerkelsesverdig.
Registrering 4	Alt ok, skriver notat.
Registrering 5	Alt ok, skriver notat.
Registrering 6	Alt ok.
Registrering 7	Alt ok.
Registrering 8	Sier eeh under telling, dette ga lengre lyttetid så dialogen ikke stanset. Får feilmelding som forstås og rettes opp.
Del 4	Alt ok. Regner med at den siste oppsynsturen er øverst. Scroller på siden, zoomer. Åpner popup fra høyresiden. Skjønner at markører er klikkbare.
Oppsummering	Løser oppgavene fint.

Gjennomføring nr.	4
Del 1	Går ikke å skrive koordinater. Finner ut at man kan zoome. Leser instruks og forstår den. Vil angre markert område, men lurer på hvordan. Avbryter og begynner på nytt.
Del 2	Alt ok, brukte nedlast-ikon.
Del 3	
Registrering 1	Prøver å trykke på kart. Ser på turoversikt, finner deretter plussmeny. Forstår instruksjon, resten ok. Fylte inn felter nedenfra og oppover, fikk feilmelding som fikset seg etter siste utfylling.
Registrering 2	Alt ok. Fyller felter nedenfra og oppover. Prøver å zoome lengre inn, men det går ikke.
Registrering 3	Velger feil registreringsposisjon, avbryter og inn på nytt.
Registrering 4	Alt ok, skriver notat.
Registrering 5	Alt ok, skriver notat og tar tre bilder.
Registrering 6	Alt ok
Registrering 7	Alt ok
Registrering 8	Får feilmelding som rettes opp, resten ok.
Del 4	Navigerer i underfaner før h*n oppdager hovedfanene. Velger øverste oppsynstur og sjekker at tidspunkene stemmer. Bruker kartmarkører for å se på registreringer.
Oppsummering	Løser oppgavene fint. Klarer å definere beiteområde, men det blir for lite. Tenkte først at plussmeny-knappen ledet til hjelp, foreslo å kun bruke plusstegn. Liker siktlinjene. Poengterer at det tar tid å gå gjennom hele dialogen.

Her gjorde vi noen endringer på saueregistrering basert på gjennomføring 1-4:

- “Sauer & lam” -> “Totalt”, endret ikoner, ullfarger under egen overskrift)
- Knappen nede til høyre på hovedskjerm (“+?” → “+”)



Gjennomføring nr.	5
Del 1	Endrer først på min gård. Navigerer til beiteområder. Navn først, zoomer. Synes det er rart at det ikke zoomes mot musepekeren. Prøver å dra og å trykke. Finner så instruksjonen. Prøver å avbryte hjørneplassering, men får det ikke til. Avbryter med knappen og starter på nytt. Liker at markeringen er et hjørne. Resten ok.
Del 2	Alt ok. Klikket på start oppsynstur fremfor nedlast-ikon.
Del 3	
Registrering 1	Holder inne på kart for å registrere. Skriver inn antall av hver ullfarge før antall sau. Fikser feilmelding.
Registrering 2	Alt ok. Holder på kart for å registrere, fyller inn ullfarger først.
Registrering 3	Holder på kart, avbryter saueregistrering. Går gjennom pluss-menyen og ser instruksjon. Resten ok.
Registrering 4	Alt ok. Kommenterer at GPS-posisjon er registreringsposisjon.
Registrering 5	Alt ok.
Registrering 6	Alt ok.
Registrering 7	Ser seg rundt i appen, skrur på autostart dialog. Holder på kart for å starte registrering. Dialog settes på pause underveis, forstår hvordan den gjenopptas. Snakker før lyttelyd. Skjønner dette. Gjenopptar. Resten ok. Poengterte at dialogen ikke hadde trengt å spørre om den siste slipsfargen basert på det den hadde fått vite så langt (det er tre lam, to av dem uten slips og én med rødt, hvorfor spørre om grønt?).
Registrering 8	Får feilmelding som forstås og rettes opp. Stusset litt på forskjellen mellom fortsett og fullfør på avslutt tur-popup. Valgte riktig. Ønsket feedback på opplasting, ble ikke betrygget av synkroniseringsstatus.
Del 4	Zoomer i kart og scroller på venstreside. Bruker kartmarkører for å se popups.
Oppsummering	Klarte å løse alle oppgaver fint. Tok noe tid å forstå hvordan beiteområdet defineres på kartet. Så rom for forbedring i appens taleregistreringsdialog, og at man kan unngå å velge posisjon for skade og kadaver (fordi det er på egen posisjon).

Gjennomføring nr.	6
Del 1	Zoomer seg inn. Trykker mye på kart, skjer ingenting. Ser ikke instruksjon før det har gått litt tid. Instruksjon var ikke tydelig nok (prøvde klikk+hold+dra istedenfor bare klikk+hold), så det ble gitt hint. Ikke fornøyd med markert område så restarter to ganger.
Del 2	Trykker på tilbake-ikon og blir overraskende logget ut. Bruker nedlast-ikon.
Del 3	
Registrering 1	Åpner først turoversikt. Velger så saueregistrering fra plussmeny. Trykker på registreringsposisjon, skjer ingenting. Prøver å holde på registreringsposisjon. Resten ok.
Registrering 2	Prøver å zoome lengre inn på kart, går ikke. Saueregistrering gjennom å holde på kart. Resten ok. Lurer på om man kan endre en registrering.
Registrering 3	Holder på kart og får opp saueregistrering. Avbryter og velger rovdyrregistrering gjennom plussmeny. Resten ok.
Registrering 4	Ønsket ikke å se bilde. Valgte skaderegistrering gjennom plussmeny. Prøver å fullføre registrering uten å ha skrevet individnummer. Oppdager feilen når inputfeltet blir rødt og gir feilmelding.
Registrering 5	Alt ok. Ønsket ikke å se bilde.
Registrering 6	Alt ok.
Registrering 7	Usikker på hvordan taleregistrering skal startes. Prøver å si "three sheep" fra hovedsiden, skjer ingenting. Trykker seg litt rundt, skrur på autostart dialog og leser tooltip. Åpner saueregistrering og dialog startes automatisk. Sier feil antall, retter opp manuelt etter avsluttet dialog. Trykker på last opp-knappen, men velger å fortsette.
Registrering 8	Sier feil antall. Snakket før lyttelyd, forstår det etter 3-5 forsøk. Dialog ble dermed satt på pause flere ganger, forstår hvordan den gjenopptas. Får valideringsfeil etter å ha registrert 7 sauer, 7 hvite og 1 svarhodet. Velger feltet som skal rettes opp i og starter dialog. Dialog stiller et annet spørsmål. Går gjennom resten av dialogen og retter opp feilen med tale.
Del 4	Alt ok. Åpner én popup fra ikon på høyresiden, og en popup gjennom kartmarkør.
Oppsummering	Problemer med å få definert kart, fikk det ikke til uten hint. Lærte underveis. Feilmeldinger ble forstått.

Gjennomføring nr.	7
Del 1	Navigerer riktig, skriver først kartnavn. Leser instruksjon. Markering går fint. Prøver å dra markeringen, synes ikke den er stor nok. Avbryter og starter på nytt. Resten ok. Nevner at det hadde vært kult å kunne søke etter sted, og ønsker å kunne dra i markering.
Del 2	Alt ok, bruker nedlast-ikon.
Del 3	
Registrering 1	Ser først på turoversikt og innstillinger. Går gjennom plussmeny, leser instruksjon og holder på kart. Får valideringsmelding, prøver likevel å trykke på fullfør flere ganger. Forstår feilen til slutt og fikser.
Registrering 2	Alt ok, går gjennom plussmeny.
Registrering 3	Alt ok, går gjennom plussmeny. Liker rød linje.
Registrering 4	Alt ok.
Registrering 5	Holder inne på kart, avbryter saueregistrering. Går gjennom plussmeny. Resten ok.
Registrering 6	Holder inne på kart, avbryter saueregistrering. Går gjennom plussmeny. Resten ok.
Registrering 7	Holder inne på kart. Snakker før lyttelyd på tiende spørsmål, gjenopptar dialog når den stopper. Resten ok.
Registrering 8	Alt ok. Trykker først på fortsett oppynstur, forstod det og trykker på fullfør neste gang.
Del 4	Ser først på underfaner, oppdager deretter hovedfaner. Resten ok.
Oppsummering	Løser oppgavene fint. Får til å definere beiteområde, men det blir ikke stort nok. Vi er usikre på om h*n lærte at holding direkte på kart åpner saueregistrering.

Gjennomføring nr.	8
Del 1	Dobbelklikket for å zoome, ønsket +/- knapper. Trykker lagre rett etter legg til, finner deretter instruks. Prøver å dra i det markerte området, det går ikke. Ville egentlig endre firkanten. Lager nytt kart, oppdaget nå at zooming kan gjøres med musehjul. Markerer nok en gang mindre område enn ønsket, lurer på om det er mulig å endre.
Del 2	Alt ok. Trykket på start oppsynstur istedenfor nedlast-ikon.
Del 3	
Registrering 1	Alt ok. Ser på turoversikt, så plussmeny. Trykker først på kart, holder deretter inne. Fyller inn ullfarger før totalt.
Registrering 2	Alt ok, holder inne på kart.
Registrering 3	Holder inne på kart, avbryter saueregistrering. Prøver det samme enda en gang, avbryter saueregistrering. Går deretter gjennom plussmeny. Resten ok.
Registrering 4	Alt ok. Går gjennom plussmeny denne gangen. Skriver notat.
Registrering 5	Går gjennom plussmeny igjen. Tar ikke bilde, ser at det er mulig idet h*n trykker på fullfør. Lurer på om det er mulig å redigere registrering, konkluderer med at det ikke går. Registrerer en gang til. Resten ok.
Registrering 6	Holder inne på kart, avbryter saueregistrering. Går gjennom plussmeny, resten ok.
Registrering 7	Går gjennom plussmeny. Svarer på første spørsmål med x sheep y lambs. Snakker også litt før lyttelyd, men går bra. Forstår at ett tall er svaret på hvert spørsmål. Stusser litt over at det ikke kommer flere spørsmål, antar til slutt at dialog er ferdig. Får feilmelding som forstås og rettes opp.
Registrering 8	Går gjennom plussmeny. Dialog stopper, forstår ikke dette på annerledes lydsignal, men virker noe påvirket av oppfordring til å ikke se på skjermen. Forstår hvordan dialogen gjenopptas. Resten ok.
Del 4	Ser seg rundt og scroller. Trodde kanskje at ikonet for "ingen slips" slettet registrering. Åpner popup gjennom høyremeny. Prøver å finne neste registrering fra høyremeny, bruker deretter kartmarkør.
Oppsummering	Stort sett bra. Får til å definere beiteområde, men det blir for lite. Vi er usikre på om h*n lærte at holding direkte på kart åpner saueregistrering. Svarer x sheep y lambs på første dialogspørsmål.

Gjennomføring nr.	9
Del 1	Liker design på innlogging. Endrer først gårdsadresse. Navigerer til beiteområder og leser beskrivelse. Alt ok. Synes det er litt dumt at det bare er mulig å definere en firkant.
Del 2	Alt ok, trykker på nedlast-ikon.
Del 3	
Registrering 1	Alt ok. Går gjennom plussmeny, fyller ullfarger først.
Registrering 2	Alt ok, går gjennom plussmeny.
Registrering 3	Alt ok, skriver notat.
Registrering 4	Alt ok, skriver notat.
Registrering 5	Alt ok. Skrev tre av fem siffer i individnummeret. Så ikke denne feilen fordi tastaturet var åpent og det var scrollet nedover på siden. Oppdaget det etterhvert og fikset.
Registrering 6	Alt ok.
Registrering 7	Svarer x sheep y lambs på første spørsmål, forstår deretter at svaret på hvert spørsmål kun er ett tall. Resten ok.
Registrering 8	Alt ok.
Del 4	Åpner kadaver-popup gjennom høyre side. Leter etter den andre registreringen på høyre side, men husker ikke helt hva slags registrering det var på posisjonen. Fant etterhvert ut at kartmarkører er klikkbare.
Oppsummering	Oppgavene løses fint. Registrerer kun gjennom plussmeny, ikke gjennom kart.

