

Sander Elias Høyland

Kan en app gjøre beitedrift enklere?

Masteroppgave i Lektorutdanning i realfag for trinn 8–13

Veileder: Svein-Olaf Hvasshovd

Juni 2023



Sander Elias Høyland

Kan en app gjøre beitedrift enklere?

Masteroppgave i Lektorutdanning i realfag for trinn 8–13
Veileder: Svein-Olaf Hvasshovd
Juni 2023

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



Kunnskap for en bedre verden

Soli Deo gloria

Abstract

There are 13 000 sheep farmers in Norway, and they have a total of almost 2 million sheep that graze out in the open during every summer. To ensure proper animal welfare and reduce the number of sheep that perish, the government requires physical supervision of the sheep. This is to be done at least weekly, and more frequently when the farmer suspects an increased level of risk. The government provides operating allowances and subsidies for sheep that are killed by predators, but in return they may require documentation of the conducted supervision. At the moment, this documentation is typically provided in unstructured text, and therefore the variation in documentation provided is substantial. This thesis examines whether an application can be created to document the conducted supervision. The application must be simple to use while satisfying both the expectations of the farmers and the requirements of the government. This thesis describes the planning, design, development and testing of such an application prototype.

The prototype developed throughout the project was evaluated by testing it on a former sheep farmer. The test demonstrated that the prototype worked as intended in many ways, but that it also had some substantial shortcomings. These shortcomings are primarily the result of the timeframe of the project. All the problems discovered as a result of the testing have clear solutions that can be implemented through further development of the application. The thesis thereby demonstrates that it is possible to create an application that satisfies both the farmers' expectations and the government's requirements.

Sammendrag

I Norge er det over 13 000 bønder som driver med sauedrift, og de har til sammen nesten 2 millioner sauer som slippes ut på utmarksbeite hvert år. For å sikre god dyrevelferd og redusere tap av sau, stilles det krav om tilsyn til sauene. Dette skal gjennomføres fysisk, minst ukentlig, og hyppigere ved mistanke om økt fare. For å få driftstilskudd og erstatning for tap av dyr, kan myndighetene stille krav til dokumentasjon av tilsynet. I dag dokumenteres dette hovedsakelig i fritekst, og dokumentasjonen er derfor lite strukturert. Denne oppgaven undersøker om det er mulig å lage en applikasjon for å dokumentere slike tilsynsturer. Applikasjonen skal være enkel å bruke, men samtidig tilfredsstillende både bøndenes forventninger og myndighetenes krav til dokumentasjon. Denne oppgaven beskriver planleggingen, designet, utviklingen, og testingen av en prototype til en slik applikasjon.

Prototypen som ble utviklet i løpet av prosjektet ble evaluert ved testing på en tidligere sauebonde. Testen viste at prototypen på mange måter fungerte som planlagt, men at den også har noen tydelige svakheter. Disse skyldes i stor grad tidsbegrensningen til prosjektet. Alle problemene som ble kartlagt gjennom brukertesten har klare løsninger som kan implementeres ved videre utvikling av applikasjonen. Oppgaven viser dermed at det er mulig å lage en applikasjon som møter både bøndenes og myndighetenes behov.

Forord

Denne masteroppgaven ble skrevet våren 2023, og representerer slutten på lektorutdanningen i realfag ved NTNU. Jeg vil takke veileder Svein-Olaf Hvasshovd for en spennende oppgave og oppfølging gjennom hele prosjektperioden. Jeg vil også takke Trond Magnus Haugen for at han tok seg tid til å teste applikasjonen og komme med verdifulle tilbakemeldinger. I tillegg vil jeg takke familien, som har støttet meg gjennom hele prosjektet.

Ikke minst vil jeg takke kollektivet, huskirka, venner og resten av 5. klasse på LUR, som har bidratt med motivasjon, oppmuntring og til tider nyttige distraksjoner fra masterskrivingen.

Til slutt vil jeg takke Gud for meningen og livet. For det er i ham vi lever, beveger oss og er til.

Sander Elias Høyland
Trondheim, juni 2023

Innhold

Abstract	i
Sammendrag	iii
Forord	v
Innhold	viii
Figurer	ix
Forkortelser	x
1 Motivasjon	1
2 Problembeskrivelse	3
3 Mål og problemstilling	5
4 Bakgrunn	7
5 Eksisterende løsninger	9
5.1 Løsninger som benytter fysisk utstyr	9
5.1.1 Telespor	9
5.1.2 Findmy	10
5.1.3 Spordata	10
5.2 Digitale løsninger	10
5.2.1 Norgeskart outdoors	10
5.2.2 Beitesnap	11
5.3 Oppsummering	12
6 Konsept	13
7 Kravspesifikasjon	15
7.1 Funksjonelle krav	15
7.2 Ikke-funksjonelle krav	17
8 Designprosess	19
8.1 Brukerorienterte designprinsipper	19
8.2 Prosjektets designprosess	22
8.2.1 Første iterasjon: Figma-design	22
8.2.2 Andre iterasjon: Implementert prototype	23

8.2.3	Tredje iterasjon: Videre arbeid	24
8.3	Oppsummering	24
9	Utviklingsprosess	25
9.1	Utviklingsmetodikk	25
9.2	Konkrete smidige metoder	26
9.3	Versjonskontroll	27
10	Programvarearkitektur	29
10.1	Oversikt over arkitekturen	29
10.2	React Native og Expo	29
10.3	tRPC	30
10.4	Next.js	30
10.5	Prisma	30
10.6	PlanetScale	31
10.7	Clerk	31
10.8	Oppsummering	31
11	Prototyper	33
11.1	Designfilosofi	33
11.1.1	Konseptuell modell	33
11.1.2	Tilbydelighet (affordance)	34
11.1.3	Antydning (signifier)	34
11.1.4	Overføring (mapping)	34
11.1.5	Tilbakemeldinger	35
11.1.6	Begrensninger	35
11.2	Prototype 1 - Designprototype	36
11.2.1	Profil	36
11.2.2	Nedlastede kart	37
11.2.3	Fullførte tilsynsturer	38
11.2.4	Ny tilsynstur	39
11.2.5	Observasjon	40
11.3	Prototype 2: Expo-prototype	41
11.3.1	Autentisering	41
11.3.2	Ny tilsynstur	42
11.3.3	Pågående tilsynstur	43
11.3.4	Observasjoner	44
11.3.5	Fullførte tilsynsturer	47
11.3.6	Nedlastede kart	49
11.3.7	Profil	49
11.4	Oppsummering	51
12	Brukertest	53
12.1	Bakgrunn	53
12.2	Testsubjekt	53
12.3	Omfang	54
12.4	Testmetodikk	54
12.5	Gjennomføring	56
12.6	Oppsummering	59

13 Evaluering av brukertesting	61
13.1 Styrker	61
13.2 Svakheter	61
13.3 Resultater	62
13.3.1 Positive punkter	62
13.3.2 Punkter som må forbedres	63
13.4 Oppsummering	64
14 Evaluering av løsningen	67
14.1 Funksjonelle krav	67
14.1.1 Kommentarer til funksjonelle krav	69
14.2 Ikke-funksjonelle krav	69
14.2.1 Kommentar til ikke-funksjonelle krav	69
15 Evaluering av prosjektet	71
15.1 Evaluering opp mot suksesskriterier	71
15.2 Evaluering opp mot problemstilling	72
16 Konklusjon	75
17 Videre arbeid	77
17.1 Oppfyllelse av eksisterende krav	77
17.1.1 Forbedringer av eksisterende funksjonalitet	77
17.1.2 Implementering av manglende funksjonalitet	78
17.2 Ny funksjonalitet	78
References	79
Vedlegg:	83
A Rapportskjemaer	84
A.1 Rapportskjema for beitesesongen 2021 Vestland	84
A.2 Rapport generert av Beitesnap	88
B Brukertest	93
B.1 Oppgaver brukt i brukertest av prototype 2	93

Figurer

8.1.1	Designprosessen i fire steg	21
11.2.1	Fane 4: Profil	36
11.2.2	Fane 3: Kart	37
11.2.3	Fane 2: Fullførte tilsynsturer	38
11.2.4	Fane 1: Ny og pågående tilsynstur	39
11.2.5	Registrer observasjon	40
11.3.1	Autentiseringsskjermer	42
11.3.2	Ny tilsynstur	43
11.3.3	Pågående tilsynstur	44
11.3.4	Ny observasjon	45
11.3.5	Observasjon av sauer	45
11.3.6	Observasjon av død sau	48
11.3.7	Observasjon ullfunn	48
11.3.8	Observasjon av rovdyr	49
11.3.9	Observasjoner på kartet	50
11.3.10	Fullførte tilsynsturer	50
11.3.11	Nedlastede kart	51
11.3.12	Profilskjermer	52

Forkortelser

Oversikt over forkortelser i alfabetisk rekkefølge:

- **API** Application programming interface
- **GPS** Global Positioning System
- **IoT** Internet of things
- **LTE** Long-term evolution
- **LTE** Long-term evolution for machines
- **MVP** Minimum viable product
- **ORM** Object-relational mapping
- **OTA** Over the air
- **PWA** Progressiv web-applikasjon
- **RPC** Remote procedure call
- **SQL** Structured Query Language

Kapittel 1

Motivasjon

Sauereproduksjon er en betydelig næring i Norge. Hvert år slippes nesten 2 millioner sauer ut på utmarksbeite [1], og det er over 13 000 norske bønder som driver med sau [2]. De fleste sauer sendes i Norge på utmarksbeite i sommerhalvåret, da dette blant annet bidrar til god dyrevelferd og gir god utnyttelse av naturressursene [3].

For å sikre at sauene blir tatt vare på mens de er på utmarksbeite, og for å begrense tap av sau til rovdyr, sykdom og lignende, krever myndighetene at bøndene har jevnlig tilsynsturer for å kontrollere at det står bra til med sauene. Tilsyn skal utføres minst ukentlig, og hyppigere ved mistanke om økt fare [4, 5]. Mattilsynet kan kreve dokumentasjon på at bestemmelsene i Forskrift om velferd for småfe etterlevs [4], og den aktuelle erstatningsmyndigheten kan kreve dokumentert loggbok for tilsyn for å tilkjenne erstatning for dyr som har blitt drept eller skadet av rovdyr [5].

Det er i dag vanlig praksis at bønder registrerer informasjon i forbindelse med tilsynsturer, og dette journalføres i fritekst. Det er ingen strukturelle krav til dokumentasjonen som føres, og det er derfor stor variasjon i hvilken dokumentasjon som registreres og hvor detaljert denne er [6, Appendix C.1].

På bakgrunn av dagens situasjon har det blitt uttrykt et ønske om et digitalt verktøy som kan bistå i prosessen for å dokumentere tilsynsturer. Dette verktøyet bør være enkelt å bruke, registrere strukturert informasjon og tilfredsstillende både bøndenes og myndighetenes behov for dokumentasjon. Denne masteroppgaven omhandler utviklingen av et produkt som forsøker å møte disse behovene.

Kapittel 2

Problembeskrivelse

Problembeskrivelsen for masteroppgaven, formulert av professor Svein-Olaf Hvasshovd, er som følgende:

Sau slippes på vårparten ut på beite. Hele sommeren går sauene fritt rundt i terrenget. I denne perioden krever myndighetene at det drives jevnlig oppsyn med sauene. Bonden må da ut og gå i terrenget hvor sauene befinner seg for å sjekke at alt står bra til. I tilfelle han detekterer at noen sau er syke eller døde må han iverksette tiltak for ivaretagelse av sauenes velferd.

Fram til i dag har bøndene ført oversikt over hvilke oppsynsturer de har vært på, hvor de har gått og hva de har sett, på papir. Det er et ønske at rapporteringen av dette flyttes over til et databasert verktøy som bøndene kan ta med ut i felt på hver oppsynstur. Innholdet i rapporten og kartet over hvor bonden har gått, lastes så ned når han kommer hjem og en standardisert rapport produseres ved sesongens slutt og sendes myndighetene.

Basert på denne beskrivelsen ble det utviklet et prosjektmål og en problemstilling bestående av fire forskningsspørsmål. Disse beskrives i neste kapittel.

Kapittel 3

Mål og problemstilling

Basert på problembeskrivelsen til Hvasshovd, ble følgende prosjektmål utformet:

Utvikle et produkt som lar bønder effektivt registrere detaljert lokasjonsbasert informasjon om sauer på utmarksbeite som tilfredsstiller myndighetenes dokumentasjonskrav på en brukervennlig måte.

Dette målet ble deretter brutt ned i følgende suksesskriterier:

S1: Produktet skal være brukervennlig, intuitivt og enkelt å bli kjent med for brukerne.

S2: Løsningen skal møte bøndenes behov for å registrere informasjon i forbindelse med tilsynsturer.

S3: Informasjonen registrert i produktet skal i tilstrekkelig grad dekke myndighetenes krav til dokumentasjon.

S4: Løsningen skal lages med moderne teknologi og være i stand til å videreutvikles og vedlikeholdes etter at prosjektet er fullført.

Problemstillingen for prosjektet ble utformet med bakgrunn i disse suksesskriteriene, og lyder som følgende:

I hvilken grad er det mulig å utvikle en mobilapplikasjon som enkelt og effektivt lar bønder registrere detaljert lokasjonsbasert informasjon om sauer på utmarksbeite, som oppfyller myndighetenes dokumentasjonskrav og kan videreutvikles til en løsning som kan tas i bruk i praksis?

Kapittel 4

Bakgrunn

Det er flere grunner til at både bønder og tilsynsmyndigheter ønsker et mer sofistikert verktøy til å føre tilsynslogger enn det som brukes per dags dato. For det første er det nyttig for bønder å se hvor sauene har vært i løpet av beitesesongen. Dette kan bistå i kartlegging av hvilke beiteområder som er best egnet for utmarksbeite, hvor sauene er spesielt utsatt for rovdyr eller sykdom, og gjøre det enklere å forutsi hvor sauene beveger seg fra dag til dag [6, Appendix C.3]. For det andre kan det være nyttig for bønder å ha detaljerte tilsynslogger for å kunne dokumentere tilsyn ovenfor myndighetene på en tilfredsstillende måte. Dette er spesielt aktuelt i forbindelse med ulike tilskuddsordninger.

Det er en rekke offentlige aktører som er involvert i produksjon av sau, både som tilsynsorganer, lovgivende organer og forvaltere av ulike tilskuddsordninger. Det stilles strenge krav til hvordan sauehold skal foregå, og sauebønder må forholde seg til flere lover og forskrifter som legger føringer for driften av saueproduksjon. Det finnes også flere tilskuddsordninger for sauebønder som bidrar til en bærekraftig næring.

Bønder har en lovfestet plikt til å utføre jevnlig tilsyn med sauene på utmarksbeite, hvor hyppigheten skal være minst ukentlig [4, 5] og oftere ved behov. Hvis dette kravet ikke tilfredsstilles, kan bøndene risikere at de ikke får innvilget søknader om tilskudd, og de kan også risikere bøter eller fengselsstraff, avhengig av blant annet graden av uaktsomhet og konsekvensene av denne [4, 7, 8]. Hvis erstatningsmyndigheten for erstatning når husdyr blir drept eller skadet av rovdyr mener at det er betydelig usikkerhet rundt dyrehold, døds- eller skadeårsak, kan de stille ekstra krav for å gi erstatning i fremtiden. Ett av disse kravene kan være dokumentert loggbok for tilsyn til sauebesetningen [5]. Det er også fylker som har egne miljø- eller produksjonstilskudd, blant annet Trøndelag fylke. Ett av disse tilskuddene er til drift av beitelag, og her stiller Statsforvalteren i Trøndelag krav om at «Det skal foreligge en plan for tilsyn og for sanking, og laget skal dokumentere tilsynet» [9]. Det er altså flere offentlige instanser som deler ut tilskudd i forbindelse med saueproduksjon hvor det som bonde kan være nyttig å ha detaljert dokumentasjon på utført tilsyn.

I dag er det uklart hvor detaljert det dokumenterte tilsynet skal være. I rapportskjemaet fra statsforvalteren i Vestland står det [Vedlegg A.1]:

Tilsynslogg kan nyttast for å sikre dokumentasjon på utført tilsyn med dyra i utmarka gjennom beitesesongen. Kvar tur skal noterast fortløpande i tabellen. Aktuelle opplysningar kan t.d. vera: daude og skadde dyr, tapsårsaker, observasjon av rovvilt, sauer som manglar lam, generell uro i beitet, beiteforhold med meir.

Statsforvalteren i Møre og Romsdal skrive på sine sider «Send gjerne med tilsynslogg» [10] i informasjonen rundt søknad om rovvilterstatning, som indikerer at en tilsynslogg er ønskelig, men ikke nødvendig. Samtidig skriver statsforvalteren også at «Manglende dokumentasjon over buskapen er grunnlag for avkorting av erstatninga» [10]. I Forskrift om erstatning når husdyr blir drept eller skadet av rovvilt [5] står det at «SNO har anledning til å be om utfyllende dokumentasjon og til å vurdere et funn til ikke påvist rovviltsskade der fremlagt dokumentasjon ikke tilfredsstillir faglige krav.» En tilsynslogg som blant annet viser hvor dyrene har vært kan være et nyttig verktøy for å kunne fastslå at en sau har blitt drept av et rovdyr.

Mangelen på standardiserte dokumentasjonskrav forårsaker uforutsigbarehet både blant bønder og tilsynsmyndigheter. Bøndene vet ikke nødvendigvis hva som kvalifiserer som tilstrekkelig detaljnivå for dokumentasjonen de registrerer. Tilsynsmyndighetene vet ikke hvordan dokumentasjonen de mottar vil se ut, hvilke detaljer den vil inneholde eller om den er tilstrekkelig detaljert til å kunne behandle søknader om tilskudd ved for eksempel tap av sauer til rovdyr. At dokumentasjonen som mottas er ustrukturert og uforutsigbar gjør også at det er utfordrende å lage god statistikk på området.

Alle disse faktorene indikerer at det kan være hensiktsmessig å ha et verktøy som gjør det enklere å dokumentere presis og detaljert informasjon under tilsynsturer. Hvis et slikt verktøy utvikles i samarbeid med de aktuelle tilsynsmyndighetene, kan det lages på en måte hvor det sikrer at bruk av verktøyet fører til at bøndene oppfyller kravene til dokumentasjon. Det kan også gjøre det enklere for bønder å registrere nyttig informasjon, blant annet ved å spore posisjonen til bonden under tilsynsturer og legge til rette for at observasjoner av sauer og andre interesseobjekter kan posisjoneres på et kart.

Kapittel 5

Eksisterende løsninger

Det finnes i dag flere tekniske løsninger som er laget for å bistå bønder og bondelag som har sauer på utmarksbeite om sommeren. Det eksisterer også generelle løsninger som er utviklet med et mer generelt formål, men som også kan brukes til oppfølging av sauer på utmarksbeite. Løsningene kan deles inn i to hovedkategorier:

- Løsninger som ved hjelp av fysisk utstyr gir bønder informasjon om sauene fra avstand over internett.
- Løsninger som ved hjelp av programvare bistår bønder i registrering og dokumentering av informasjon om sauene.

5.1 Løsninger som benytter fysisk utstyr

Det har de siste årene dukket opp en rekke aktører som tilbyr enheter som kan bistå med oppfølging og lokalisering av sauer over avstand. Disse løsningene består typisk av GPS-enheter som festes til en andel av sauene og lar bøndene spore hvor sauene befinner seg. I 2020 hadde mellom 5 og 10 prosent av sauene i Norge installert sporingsutstyr [11]. Enhetene som monteres på sauene bruker ulike tekniske løsninger for kommunikasjon og sporing, og har også forskjellige filosofier når det kommer til batteriløsninger [12].

5.1.1 Telespor

Telespor utvikler produktet Radiobjella [13], og er på sin fjerde generasjon av produktet. Radiobjella benytter satelittsporing for å observere posisjonen til sauene, LTE-M og Narrowband IoT for kommunikasjon med bonden og en bevegelsessensor for å overvåke om den aktuelle sauene er i bevegelse. Radiobjella kan dele posisjonen sin så ofte som hvert 5. minutt, og intervallet kan endres mens bjella er i bruk. Bjella inneholder utskiftbare batterier som har en beregnet levetid på én sesong, og den er vanntett. Den inneholder også funksjonalitet for å varsle bonden om sauene har sluttet å bevege seg eller hvis posisjonen til en sau er utilgjengelig gjennom flere rapportsykluser. Telespor er aktøren som har holdt på med GPS-sporing av sau lengst, og er også den aktøren i Norge som sporer flest sauer, med rundt 100 000 bjeller i bruk [12].

5.1.2 Findmy

Findmy produserer produktet E-bjella [14, 15], og denne kommer i tre ulike modeller. De ulike modellene differensieres ved at de enten tilbyr GPS-sporing, sporing og kommunikasjon gjennom mobilnettet eller begge deler. Modellen med bare GPS-sporing støtter enveis kommunikasjon fra bjella til bonden, men har ikke støtte for kommunikasjon fra bonden til bjella. Modellen med bare mobilnett har støtte for toveiskommunikasjon mellom bjella og bonden, men har mindre nøyaktig sporing da den ikke har støtte for GPS. Modellen som tilbyr både GPS-sporing og mobilnettfunksjonalitet veksler mellom mobilnettet og GPS for å få den mest nøyaktige plasseringen til en hver tid. Alle modellene har støtte for tilpasset sendeplan, hvor bonden kan velge hvor ofte bjella skal kommunisere med bonden, og hvert batteri varer lenge nok til at bjella kan sende 2000 meldinger. Alle bjellene har også funksjonalitet som gjør at den kan gi beskjed om sauene er inaktiv, stresset eller har beveget seg innenfor/utenfor et geo-gjerdet område. Findmy har rundt 40 000 bjeller i bruk [12].

5.1.3 Spordata

Spordata produserer produktet Gjetargut [16]. Gjetargut tilbyr GPS-sporing av sauene på samme måte som de andre tilbyderne, men kommuniserer over LoRa (Long range radio). Denne teknologien fungerer ved at det settes opp egne basestasjoner som kommuniserer med Gjetargut-enhetene over LoRa-spekteret, og deretter sender informasjonen videre til bonden ved hjelp av mobildata. Ved å ta i bruk denne teknologien kan bjellene fungere også i områder hvor det er dårlig mobildekning. Gjetargut-enheten sender et fast antall posisjoner i døgnet for å sikre god batterilevetid, og selskapet har rundt 5000 enheter i bruk [12].

5.2 Digitale løsninger

Løsningene beskrevet i forrige delkapittel gjør jobben som sauebonde enklere, da det blir lettere å lokalisere sauene og en kan potensielt motta varsler om uønsket adferd registreres i flokken. Allikevel er det ikke en tilstrekkelig løsning, da det er offentlige krav til fysisk tilsyn av sau på utmarksbeite [4]. Hjelp til å lokalisere sauene kan være gunstig i forbindelse med tilsynsturer, men bøndene vil fortsatt være nødt til å utføre tilsynet fysisk [4, 5]. Dette delkapittelet beskriver digitale løsninger som i større eller mindre grad bistår bønder i registrering av relevant informasjon i løpet av tilsynsturer.

5.2.1 Norgeskart outdoors

Norgeskart Outdoors [17] er en app som er tilgjengelig på Android og iOS. Appen bruker karttjenesten Norgeskart fra Kartverket som kartløsning. Appen kan brukes til å registrere turer hvor appen sporer lokasjonen til brukeren, og brukeren har også mulighet til å legge inn punkter med tittel og beskrivelse i kartet. Turer registrert med appen inneholder blant annet informasjon om starttidspunkt, sluttidspunkt og total lengde for turen. Punkter registrert i appen kan kategoriseres, men brukeren kan bare velge mellom en liste med

forhåndsdefinerte kategorier. Disse inkluderer blant annet «Hjem», «Bil» og «Geocache», og kategoriene er i liten grad egnet for bønder. Ved å registrere punkter på kartet har bonden mulighet til å legge inn informasjon gjennom feltene «Tittel» og «Beskrivelse», men appen støtter ikke noe mer strukturert informasjon enn dette. Appen har en premium-versjon som gir brukeren tilgang til flere kart, mulighet til å laste ned kart til telefonen, skylagring av data, osv. Premium-versjonen koster 59 kroner i året.

Sammenlignet med dagens løsning, hvor bønder fyller ut feltene «dato», «rute/område» og «observasjoner» for hver tilsynstur, gir denne løsningen mulighet for å registrere mer detaljert informasjon. Ved å ta i bruk appen Norgeskart Outdoors, kan bøndene registrere en posisjon for hver observasjon, få en oversikt over varighet og distanse for tilsynsturen, og få en mer nøyaktig oversikt over ruten de har fulgt. Appen gir brukere mulighet til å eksportere både registrerte punkter og turer som .gpx-filer, som er en type XML-skjema som kan brukes til å lagre GPS-data [18]. Denne filtypen kan relativt enkelt lastes opp i kartløsninger som Google Maps eller mtbmap.no. Dermed kan den lagrede informasjonen enkelt fremstilles visuelt i tillegg til tekstlig. Applikasjonen inneholder derimot ingen annen funksjonalitet for å generere rapporter, og er dermed i hovedsak et verktøy som tillater brukeren å registrere informasjon. Fordelen ved å bruke denne applikasjonen som verktøy under tilsynsturer er at den sporer posisjonen til brukeren og tillater posisjonsfestede observasjoner. Ulempene er at det er vanskelig å få gjort noe fornuftig med det registrerte datamaterialet, og informasjonen vil trolig måtte overføres manuelt til et annet system for å kunne brukes som dokumentasjon.

5.2.2 Beitesnap

Beitesnap er en app laget til Android og iOS, og er skapt som et verktøy som kan hjelpe bønder med registrering av tilsyn. Den beskrives på nettsiden til applikasjonen [19] som en app laget av gårdsbrukere, for gårdsbrukere, og har to hovedfunksjoner. For det første skal den gjøre det lettere for bønder å registrere informasjon under tilsynsturer, og støtter blant annet sporlogg av der bonden har beveget seg under turen. I tillegg lar appen turgåere sende inn observasjoner av sauer på beite, så bøndene kan få løpende oppdateringer om hvordan det går med sauene av alle som ferdes i naturen. Brukere av appen har deretter mulighet til å generere en rapport på slutten av sesongen med en oversikt over tilsynsturer og observasjoner gjort av både bonden selv og andre turgåere. Vedlagt ligger et eksempel på en slik rapport [Vedlegg A.2]. I følge BeiteSnap fikk gardbrukeren som sendte inn denne loggen ingen avkortning på erstatningssøknaden sin [20]. Appen virker i følge utvikleren både med og uten internett, og laster opp registrert informasjon til skyen når den blir koblet på internett. Appen er gratis å bruke for turgåere, men koster 1200 kroner pluss MVA per sesong for bønder.

I følge et intervju med Bondevennen [21] ble appen utviklet fra 2016 til 2017, og ble brukt av flere bondelag i 2017. Mellom 2017-sesongen og 2018-sesongen ble appen utbedret etter tilbakemeldinger om mangler og behov fra brukerne.

I mars 2019 annonserte BeiteSnap at appen ville være gratis tilgjengelig for både turgåere og bønder i 2019-sesongen [20]. Selskapet bak appen oppgir i annonseringsinnlegget at grunnen til at appen blir gjort gratis tilgjengelig er på grunn av få brukere. Målet med å gjøre appen gratis tilgjengelig for sesongen er å øke brukertallet. Dette innlegget er den siste oppdateringen på nettsidene til Beitespor. Appen er per dags dato (06.03.2022) utelukkende tilgjengelig på Apple's App Store [22]. Appen har ikke mottatt noen programvareoppdateringer siden 2019. I hvilken grad applikasjonen bistår bønder i utførelsen av tilsynsturer er dermed ikke spesielt relevant, da løsningen tilsynelatende er avviklet, og i hvert fall ikke under aktiv utvikling. En løsning som er avviklet er åpenbart ikke til spesielt stor nytte.

5.3 Oppsummering

Det er flere tilbydere av tekniske løsninger som ser ut til å være til nytte for bønder, og flere av disse blir også brukt aktivt. Det er derimot tydelig at det per dags dato ikke eksisterer et godt verktøy for å registrere informasjon tilknyttet tilsynsturer. Løsninger som Norgeskart Outdoors kan tilsynelatende brukes til dette formålet, men appens generelle publikum (turgåere) og mangelen på aktuell funksjonalitet for bønder med sauer på utmarksbeite gjør løsningen kronglete å bruke i praksis. Det er heller ingen tjenester i drift som automatisk genererer en tilsynsrapport som kan brukes som dokumentasjon ovenfor tilsynsmyndigheter. Basert på denne situasjonen er det tydelig at det er rom for en ny løsning som kan fylle dette behovet, og denne oppgavens mål er dermed meget relevant.

Kapittel 6

Konsept

Med utgangspunkt i prosjektmålet og det kartlagte behovet for informasjon fra både bøndernes og myndighetenes side, ble følgende konsept utviklet:

En applikasjon tilgjengelig på Android og iOS, laget for å gjøre det enklere for bønder å registrere informasjon under tilsynsturer. I appen har brukerne mulighet til å gjennomføre tilsynsturer, se tidligere gjennomførte tilsynsturer og laste ned kart til bruk når enheten ikke har internett tilgjengelig.

Før applikasjonen kan tas i bruk, må brukeren registrere en brukerkonto. Alle tilsynsturene som gjennomføres i applikasjonen knyttes til brukeren som er logget inn, og dette tillater at en bruker har tilgang til all registrert informasjon på tvers av enheter. Dette gjør det også mulig for flere personer å bruke den samme brukeren på ulike enheter, noe som er en stor fordel hvis det er flere personer som samarbeider om å føre tilsyn for en flokk med sauer.

I løpet av en tilsynstur skal brukeren av applikasjonen se et kartutsnitt fra Norgeskart. På kartet skal brukerens nåværende posisjon være synlig, samt ruten som brukeren har beveget seg siden tilsynsturen ble startet. I løpet av tilsynsturen har brukeren mulighet til å registrere ulike typer observasjoner, hvor hver observasjon har ulike valgmuligheter, i tillegg til et beskrivelsesfelt hvor brukeren kan registrere annen relevant informasjon om observasjonen. Hver observasjon skal også gi brukeren mulighet til å legge til bilder. Observasjonene registreres med en posisjon på kartet, og posisjonen brukeren har når observasjonen registreres lagres også som en del av observasjonen. Det bør også være mulig å redigere disse etter at de er registrert. Observasjonene vises også på kartet i løpet av tilsynsturen. Etter at tilsynsturen er avsluttet kan informasjonen som ble lagret i løpet av turen, inkludert brukerens rute og de registrerte observasjonene, vises ved å velge turen fra en liste med gjennomførte tilsynsturer.

De ulike observasjonene brukeren kan velge å registrere består som et minimum av følgende kategorier:

- Saueflokker, hvor brukeren kan registrere søyer og lam i ulike farger, i tillegg til eventuelle skader.
- Døde sauer, hvor brukeren kan registrere om den døde sauen er en søye

eller et lam, hvilken farge den har og hva som er antatt dødsårsak.

- Ullfunn, hvor fargen på ullfunnet registreres.
- Rovdyr, hvor observerte rovdyr kan registreres.

Applikasjonen skal kunne brukes både med og uten internett, og brukeren skal kunne velge om hver observasjon skal lagres bare på nett, bare på enheten eller begge steder. Den skal også gi mulighet til å generere en rapport basert på de gjennomførte tilsynsturene, som kan brukes som dokumentasjon til de relevante tilsynsmyndighetene.

Kapittel 7

Kravspesifikasjon

7.1 Funksjonelle krav

For å kunne vurdere kvaliteten på løsningen som ble utviklet i løpet av prosjektet, ble det utarbeidet et sett med funksjonelle krav basert på konseptet beskrevet i forrige delkapittel. Disse kravene anses å være nødvendige for at applikasjonen skal oppfylle de fire suksesskriteriene beskrevet i kapittel 3. Hvert krav har prioriteten *høy*, *middels* eller *lav*, og under følger en oversikt over hva disse prioritetene representerer.

Krav med prioriteringen *høy* anses som nødvendige for at applikasjonen skal kunne brukes til formålet beskrevet i kapittel 3. Konsekvensen av at krav i denne kategorien ikke oppnås er at applikasjonen ikke fungerer i det hele tatt, eller at applikasjonen ikke dekker minimumsbehovene for at den skal ha verdi for brukerne.

Krav med prioriteringen *middels* anses som viktige for at brukerne skal oppleve at applikasjonen dekker behovene deres. Konsekvensen av at krav i denne kategorien ikke oppnås kan være at enkelte brukere ikke kan bruke applikasjonen eller at enkelte brukere må supplere informasjonen som registreres i applikasjonen med informasjon registrert på andre måter.

Krav med prioriteringen *lav* anses som nyttige, men ikke nødvendige. At disse kravene tilfredstilles vil forbedre brukeropplevelsen til applikasjonen, men krav i denne kategorien som ikke oppnås vil ikke begrense hvem som kan bruke applikasjonen eller hva den kan brukes til.

Id	Beskrivelse	Prioritet
F1	Brukeren skal kunne starte og fullføre tilsynsturer	Høy
F1.1	Tilsynsturer skal kunne startes i appen	Høy
F1.2	Tilsynsturer skal kunne fullføres i appen	Høy
F1.3	Tilsynsturer skal kunne avbrytes i appen	Middels
F1.4	Brukeren skal advares med en bekreftelsesdialog når vedkommende prøver å avbryte/avslutte turen	Lav
F2	Brukeren skal se et kart over Norge i løpet av tilsynsturen	Høy
F2.1	Kartet som vises skal være Norgeskart fra Kartverket	Middels
F2.2	Kartet skal vise den nåværende posisjonen til brukeren	Middels
F2.3	Kartet skal vise ruten brukeren har beveget seg under turen	Lav
F3	Brukeren skal kunne registrere observasjoner med tilstrekkelig informasjon i løpet av tilsynsturene	Høy
F3.1	Det skal være mulig å registrere saueflokker bestående av et vilkårlig antall søyer og lam	Høy
F3.1.1	Brukeren skal kunne velge fargen på hver enkelt sau og om hver enkelt sau er skadet eller ikke	Høy
F3.1.2	Det skal være mulig å registrere slipsfarge og slipsnummer på hver enkelt sau	Middels
F3.2	Det skal være mulig å registrere døde søyer og lam i appen	Høy
F3.2.1	Observasjoner av døde sauer skal ha støtte for å registrere slipsnummer, slipsfarge og fargen til sauen	Middels
F3.2.2	Observasjoner av døde sauer skal gi mulighet for å legge inn en dødsårsak	Middels
F3.3	Det skal være mulig å registrere ullfunn i appen	Middels
F3.4	Brukeren skal kunne registrere observasjoner av rovdyr i appen	Middels
F3.5	Alle observasjonene skal lagre brukerens posisjon på tidspunktet da observasjonen ble registrert	Middels
F3.6	Alle observasjonene skal ha støtte for å legge inn en beskrivelse	Middels
F3.7	Alle observasjonene skal ha støtte for å legge inn bilder	Lav
F4	Brukeren skal kunne se og redigere registrerte observasjoner i løpet av tilsynsturen	Middels
F4.1	Det skal være mulig å se lagret informasjon og endre denne ved å trykke på observasjonene på kartet i løpet av tilsynsturen	Middels
F5	Appen skal kontinuerlig spore posisjonen til brukeren i løpet av hver tilsynstur	Høy

F6	Brukeren skal kunne se en liste over gjennomførte tilsynsturer i appen	Høy
F6.1	Når brukeren trykker på en av turene i lista, skal appen vise et kart som inneholder observasjonene og ruten fra turen	Lav
F6.1.1	Når brukeren trykker på en av observasjonene på kartet, skal appen vise den registrerte informasjonen fra observasjonen	Lav
F7	Appen skal gi brukeren mulighet til å generere en rapport som inneholder informasjon om de gjennomførte tilsynsturene	Høy
F7.1	Rapporten skal kunne brukes som dokumentasjon ovenfor relevante myndigheter	Høy
F7.2	Rapporten skal inneholde dato, starttidspunkt, sluttidspunkt, distanse og observasjoner fra hver tilsynstur	Høy
F7.3	Rapporten skal inneholde et kartutsnitt som viser ruten som ble fulgt under turen, samt observasjonene som ble gjort	Middels
F8	Appen skal lagre gjennomførte turer på nett	Middels
F8.1	Appen skal lagre alle nye observasjoner fortløpende på nett	Middels
F8.2	Appen skal med jevne mellomrom lagre posisjonsdata på nett	Middels
F9	Appen skal kunne brukes uten internett	Middels
F9.1	Appen skal ha støtte for å laste ned kartutsnitt	Middels
F9.2	Appen skal lagre gjennomførte turer på enheten	Middels
F9.3	Appen skal lagre alle nye observasjoner fortløpende på enheten	Middels
F9.4	Appen skal med jevne mellomrom lagre posisjonsdata på enheten	Middels
F10	Appen skal kreve innlogging for å kunne brukes	Middels
F10.1	Brukeren skal kunne opprette en konto i appen	Middels
F10.2	Brukeren skal kunne se data lagret på nett når innlogget	Middels
F10.3	Brukeren skal bare ha tilgang til egen data i appen	Middels
F10.4	Brukeren skal kunne logge inn med Google, Apple og Facebook	Lav

7.2 Ikke-funksjonelle krav

I tillegg til de funksjonelle kravene har det blitt utformet et sett med ikke-funksjonelle krav. Disse kravene har ikke fått noen prioritet, da alle kravene anses å være nødvendige for at applikasjonen skal kunne tas i bruk i praksis. De ikke-funksjonelle kravene er kategorisert basert på kategorier hentet fra ISO/IEC

25010:2011 [23].

Id	Kategori	Beskrivelse
IF1	Kompatibilitet	Appen skal være tilgjengelig både på Android og iOS
IF2	Brukbarhet	Appen skal kunne brukes uten å lese instruksjoner eller dokumentasjon på forhånd
IF3	Brukbarhet	Det skal ikke være mulig å forårsake serverfeil ved å bruke appen
IF4	Pålitelighet	Appen skal bevare alle observasjoner og det meste av posisjonsloggen om den krasjer
IF5	Ytelse	Appen skal skalere dynamisk basert på antall brukere og mengden lagret informasjon

Kapittel 8

Designprosess

Denne oppgaven har tatt utgangspunkt i ISO-standard 9241-210:2019 sitt rammeverk for menneskeorientert design for interaktive systemer [24]. Rammeverket beskrevet i standarden har blitt fulgt så langt det har latt seg gjøre med de gitte begrensningene til prosjektet. I dette prosjektet har designprosessen blitt brukt som utgangspunkt for hele utviklingsprosessen, da den tekniske implementasjonen også er med på å forme hvordan interaksjonen med produktet ser ut. Uttrykkene *designet* og *produktet* brukes derfor om hverandre.

8.1 Brukerorienterte designprinsipper

ISO-standard for brukerorientert design [24] beskriver seks prinsipper for brukerorienterte designprosesser som bør følges:

1. **Designet er basert på en eksplisitt forståelse av brukerne, oppgavene og brukskonteksten.** For å utvikle et produkt som løser problemene til en brukergruppe, er det viktig å forstå hvem de potensielle brukerne av produktet er, hvilke oppgaver eller problemer produktet kan løse, og hvordan brukerne kan bruke produktet som skal designes.
2. **Brukere er involvert gjennom hele design- og utviklingsprosessen.** Ved å involvere brukerne gjennom hele designprosessen kan misforståelser og feilaktige antakelser forhindres eller raskt oppklares. Dette reduserer sannsynligheten for at beslutninger tas på feil grunnlag og at brukernes krav til produktet som utvikles er kjente og tas høyde for. Det er viktig at de involverte brukerne representerer bredden i brukergruppen til produktet.
3. **Designet drives og videreutvikles basert på brukersentrert vurdering.** Å teste designet på reelle brukere er med på å sikre at produktet møter behovene til brukerne. Det gjør også at man får testet ut designet i den virkelige verden, og løsninger som virker bra i teorien kan forbedres hvis det viser seg at de ikke fungerer i praksis.
4. **Prosesen er iterativ.** Ved å ta i bruk en iterativ prosess, kan man kontinuerlig redusere graden av usikkerhet gjennom hele utviklingsprosessen. Dette er også med på å sørge for at brukernes behov møtes. Det kan oppstå uforutsette problemer eller situasjoner i bruken av

et produkt som ikke kunne vært planlagt på forhånd, og ved å ta i bruk en iterativ designprosess kan disse problemene løses i neste iterasjon av designet.

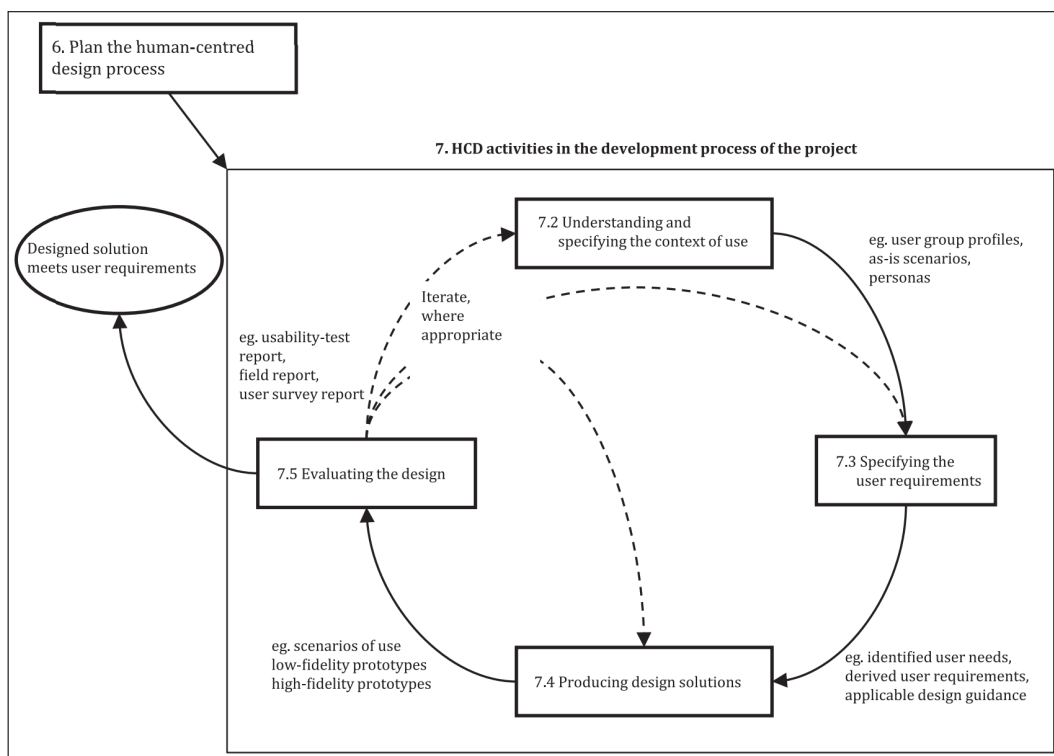
5. **Designet adresserer hele brukeropplevelsen.** Det er viktig å vurdere alle aspekter av hvordan produktet som designes vil påvirke brukerne av produktet. Dette inkluderer aspekter som ikke er direkte knyttet til produktets funksjonalitet, som hvordan produktet påvirker brukernes jobbtillfredshet, eller hvor trygt det er å bruke produktet.
6. **Designteamet består av medlemmer med tverrfaglige ferdigheter og perspektiver.** Ved å ha en rekke ferdigheter og perspektiver representert, er det større sjanse for at produktet som utvikles fullstendig adresserer behovet til brukerne. Et sammensatt team gir også økt innsikt i utfordringene og begrensningene som hvert i steg i produktutviklingen må forholde seg til. At en designer har innsikt i hva som er enkelt eller vanskelig å implementere teknisk gjør at vedkommende kan ta høyde for disse faktorene allerede før den tekniske implementeringen har begynt.

Standarden beskriver også en firstegs-designprosess for å utvikle et design eller et produkt. Prosessen illustreres i figur 8.1.1. Denne prosessen er iterativ, hvor man fortsetter på det første steget etter at man er ferdig med det siste. Den iterative naturen til prosessen gjør det mulig å starte på et vilkårlig steg i prosessen [24]:

1. **Forstå og spesifisere brukskonteksten.** For å kunne utvikle et produkt som møter brukernes behov, er det viktig å forstå hvordan produktet vil brukes. Dette kan innebære å studere eksisterende løsninger for å finne ut hvilke utfordringer disse løsningene adresserer, hvilke svaketer de har, og hvordan løsningene brukes av brukergruppen. Dette kan blant annet vise hvilke behov som ikke oppfylles av de eksisterende løsningene, som igjen kan være med å informere hvilke brukerkrav som bør stilles til det nye produktet som skal utvikles. Denne delen innebærer også å finne hvilke særtrekk brukergruppen har, hvilke hensyn som må tas med bakgrunn i konteksten produktet skal brukes i, og å kartlegge målene til brukerne.
2. **Spesifisere brukerkrav.** Når en har fått en oversikt over hvilke behov brukerne har, hvordan produktet en forsøker å lage kan møte disse utfordringene, og hvilken kontekst produktet skal brukes i, kan man begynne å spesifisere kravene brukerne har til produktet som skal designes. Dette innebærer krav som produktet må oppfylle for å løse problemene til brukerne, krav til brukervennlighet, ytelseskrav, og krav til hvilken kontekst produktet skal kunne brukes i. Kravene skal kunne testes, verifiseres av relevante aktører, ikke være selvmotstridende og oppdateres etter behov gjennom prosjektet.
3. **Produsere et design.** Etter at kravene er verifisert, kan produksjonen av et design begynne. For et produkt er dette typisk en prototype, hvor prototypen likner mer og mer på et ferdig produkt jo flere iterasjoner av designprosessen som har blitt gjennomført. Dette er fordi alle valgene som tas i utviklingen av produktet kan påvirke i hvilken grad produktet

oppfyller brukerkravene, og det er derfor viktig å teste prototyper som ligner mer og mer på sluttproduktet. Det er også et viktig prinsipp at det er bedre å lage et røft utkast tidlig i prosessen, teste det, og raskt få tilbakemeldinger på utkastet, enn å bruke mye tid på å gjøre produktet «ferdig nok» til å testes. Dette er for å sikre at tilbakemeldingene på prototypen kommer tidlig i prosessen, så tilpasninger og endringer kan gjøres før unødvendig mye ressurser har blitt investert i en idé som ikke fungerer i praksis.

4. **Evaluere designet.** Brukerorientert vurdering er en essensiell del av en brukerorientert designprosess. Det er ikke alltid praktisk gjennomførbart å teste designet på faktiske brukere i alle deler av designprosessen, og i disse tilfellene kan for eksempel simuleringer brukes som en erstatning. Målet med evalueringen er å hente inn ny informasjon om brukernes behov, teste designets styrker og svakheter, vurdere i hvilken grad brukerkravene oppfylles og eventuelt sammenlikne ulike design. Det er viktig at evaluering blir gjort både tidlig og sent i prosessen, at testingen er detaljert nok til at resultatene har verdi, at resultatene blir analysert og fører til endringer når det er nødvendig, og at løsningene på utfordringene som kommer fram i testingen brukes i utviklingen av neste iterasjon av designet.



Figur 8.1.1: Designprosessen i fire steg

8.2 Prosjektets designprosess

8.2.1 Første iterasjon: Figma-design

Dette prosjektet startet på steg 1 av designprosessen beskrevet ovenfor, ved å undersøke tidligere masteroppgaver med liknende mål og problemstillinger. Å studere hva tidligere studenter har gjort, i hvilken grad det fungerte og hvilke refleksjoner de har gjort seg rundt resultatene de fikk, ga god innsikt i brukergruppen og brukskonteksten til produktet. Parallelt med undersøkelsen av tidligere masteroppgaver, ble problemstillingen og forutsetningene diskutert med veileder Svein-Olaf Hvassovd, som har jobbet med utfordringene knyttet til tilsynsturer og dokumentasjon i flere år. Disse to kildene ga god innsikt i hvilke problemer applikasjonen bør forsøke å løse, hva brukerne trenger, hvilken funksjonalitet applikasjonen skal ha for at den er til nytte også for tilsynsmyndighetene, og hvilke utfordringer som eksisterer rundt å lage en slik løsning. Eksempler på særtrekk som skiller brukergruppen fra en typisk brukergruppe for en annen applikasjon er for eksempel behovet for å kunne registrere informasjon uten å kontinuerlig se på telefonskjermen. Dette er spesielt nyttig for brukergruppen da sauer ofte observeres ved bruk av kikkert. Da er det vanskelig å taste inn detaljert informasjon på en telefonskjerm om dette krever at du ser på skjermen mens informasjonen registreres.

Etter å ha fått en oversikt over tidligere forsøk på å løse liknende oppgaver og en grunnleggende forståelse av brukskonteksten, brukergruppen og deres behov, ble det i samarbeid med veileder utviklet et sett med overordnede krav til applikasjonen. Dette var krav som at applikasjonen skal bruke Norgeskart som kartløsning, ha mulighet for å laste ned kart til bruk uten internett, at det skal være mulig å registrere en rekke observasjoner i applikasjonen, og at applikasjonen skal spore brukerens posisjon. Disse kravene la føringer for hvordan sluttproduktet kunne se ut, og eliminerte mange potensielle måter å løse problemet på. Krav om posisjonssporing og mulighet for nedlasting av kart utelukket for eksempel muligheten for å lage en progressiv web-applikasjon (PWA) [25].

Når et grunnleggende sett med krav hadde blitt utformet, startet utviklingen av en første prototype i form av et design til applikasjonen. Designet ble utviklet ved hjelp av Figma [26] i form av en rekke bilder av ulike sider i applikasjonen, og dette fungerte som en skisse for hvordan applikasjonen kunne se ut. Figma gir også mulighet for å simulere bruken av designet, som lar utvikleren teste hvordan brukergrensesnittet fungerer i praksis, men uten å trenge å implementere produktet teknisk [27]. Denne funksjonaliteten ble ikke brukt i dette prosjektet, da det ikke var aktuelt å teste ut designet på brukergruppen i denne fasen av prosjektet. Det var utfordrende å produsere et design som både tilfredstilte kravene til funksjonalitet skissert i forrige steg av designprosessen og samtidig var intuitivt og brukervennlig. Det viste seg også senere i prosessen at noen av antakelsene som ble gjort i forbindelse med utviklingen av designet ikke var hensiktsmessige når det kom til den tekniske implementeringen av produktet.

Da førsteutkastet til designet av applikasjonen var ferdig, ble det gjort en rask evaluering av designet i samråd med veileder. Nøkkelpunktene fra denne evalueringen ble notert ned, og disse punktene var med å informere hvordan designet til applikasjonen endret seg fra prototype 1 til prototype 2. Et eksempel er posisjonen til brukeren på tidspunktet hvor en observasjon registreres. Designet tok ikke høyde for at denne posisjonen skulle lagres og vises, men veileder påpekte at det var hensiktsmessig å ha med denne informasjonen. Disse innspillene ble notert ned og tatt med i den videre utviklingen av applikasjonen.

8.2.2 Andre iterasjon: Implementert prototype

Proessen med å designe et førsteutkast til applikasjonen i Figma og den påfølgende evalueringen av designet sammen med veileder ga en grundig forståelse av behovene til brukergruppen - og utfordringene med å møte dette behovet. Utover dette ble ikke forståelsen av brukergruppen og brukskonteksten endret i spesielt stor grad i løpet av første designiterasjon. Andre designiterasjon handlet derfor i hovedsak om å bygge videre på designet fra første iterasjon. Det ble allikevel gjort noen små tilpasninger for å tilfredsstille tilbakemeldingene etter første evalueringsrunde. I tillegg var det noe funksjonalitet prototype 1 ikke tok høyde for, og tilpasninger som gjorde denne funksjonaliteten tilgjengelig ble derfor også gjort i prototype 2.

Før den tekniske implementeringen av applikasjonen ble igangsatt, ble det laget et mer detaljert sett med brukerkrav. Her ble noen av de mer generelle kravene fra designiterasjon 1 brutt ned i mer detaljerte krav, og det ble lagt til nye krav på bakgrunn av lærdommen fra iterasjon 1. For kravet om at applikasjonen skal kunne lagre observasjoner ble det spesifisert at alle observasjoner registrert i applikasjonen skal inneholde brukerens posisjon på registreringstidspunktet. Kravet om at Norgeskart skal brukes som kartløsning ble utdypet til å si at brukeren skal kunne se et kart over Norge i løpet av tilsynsturen, og at karttypen skal være Norgeskart. Det ble også lagt til at kartet kontinuerlig skal vise posisjonen til brukeren, og at ruten brukeren har beveget seg siden starten av tilsynsturen skal vises på kartet.

Den tekniske implementeringen bestod primært av å forsøke å oppnå flest mulig av brukerkravene som ble laget i forrige steg av designprosessen. Å utvikle en hel applikasjon tok mye tid, og det var lite hensiktsmessig å teste produktet før en betydelig del av kjernefunksjonaliteten fungerte. En stor andel av tiden til rådighet for prosjektet ble derfor brukt på dette steget. Det var til stor hjelp å ha både et design utviklet i Figma i forkant av dette steget og tydelige krav til hva applikasjonen skulle inneholde av funksjonalitet. Dette gjorde utviklingen målrettet. Siden målet i denne delen av designprosessen var å få ferdig den viktigste kjernefunksjonaliteten, var det mange brukerkrav som ble utsatt. Dette ble gjort enten fordi kravene ikke var kritiske for å kunne teste applikasjonen, eller fordi de i mindre grad var nødvendige å teste. Et eksempel på dette er nedlasting av kart, som er en viktig funksjonalitet, men ikke nødvendig for å teste de andre delene av applikasjonen. Et annet eksempel er støtte for posisjonsdata når applikasjonen kjører i bakgrunnen. Denne funksjonaliteten er

nødvendig for at applikasjonen skal kunne brukes i virkeligheten, men det er mindre viktig å teste brukeropplevelsen av posisjonssporing når applikasjonen kjører i bakgrunnen. På grunn av begrenset tid ble mange slike prioriteringer gjort i denne delen av prosjektet.

Da nok funksjonalitet var implementert til at applikasjonen kunne testes på brukergruppen, startet planleggingen av en brukertest. Testen ble utformet for å teste applikasjonens kjernefunksjonalitet. Dette ble gjort ved å lage en rekke oppgaver, eller brukerhistorier, som testbrukeren skulle forsøke å løse ved hjelp av applikasjonen. Disse oppgavene ble gitt en etter en til brukeren, og mens brukeren forsøkte å løse oppgavene ble samhandlingen med applikasjonen registrert. Dette ga verdifull innsikt i hvilke deler av applikasjonen som fungerte etter hensikten, hvilke svakheter applikasjonen hadde, på hvilken måte applikasjonen var intuitiv og hvilke elementer som burde endres og forbedres. Det var for eksempel utydelig for testbrukeren hvordan registreringen av en ny observasjon skulle igangsettes, men selve registreringen av informasjonen gikk fint.

8.2.3 Tredje iterasjon: Videre arbeid

Da brukskonteksten allerede var godt kjent, og i liten grad endret seg etter brukertesten av prototype 2, ble det første steget i designprosessen hoppet over i tredje iterasjon av designprosessen. Tidsrammen til prosjektet ga heller ikke rom for flere runder med utvikling av applikasjonen. De største problemene med applikasjonen ble allikevel beskrevet som et utgangspunkt for hva det hadde vært naturlig å forbedre ved videre utvikling av applikasjonen. Noen av disse problemene hadde blitt løst om flere av kravene i kravspesifikasjonen hadde blitt oppnådd, mens andre trolig ikke hadde blitt hverken løst eller oppdaget uten brukertesten. Mangelen på umiddelbar tilbakemelding når brukeren trykker på en knapp, eller mangelen på en åpenbar grunn til å gå inn på profilsiden, er eksempler på slike problemer. Ved videre utvikling av applikasjonen hadde det vært relevant å først forbedre de kartlagte problemene, og deretter jobbe videre med å implementere funksjonalitet som ble nedprioritert i prototype 2.

8.3 Oppsummering

Bruken av en iterativ designprosess og fokuset på brukernes behov har vært viktig for å ivareta en målrettet og fleksibel produktutvikling. Antakelsene som ble gjort i starten av prosjektet ble testet underveis, og prosessen la opp til at det var enkelt å gjøre endringer. Rammene til prosjektet begrenset antall iterasjoner det var praktisk mulig å gjennomføre, men fokuset på brukergruppens problemer og brukeropplevelse var et gjennomgående prinsipp gjennom hele utviklingsprosessen. Utviklingen av prototypene ble også gjort på en måte som gjør det enkelt å videreutvikle løsningen, og resultatet fra brukertestingen gir tydelige indikasjoner på hvordan applikasjonen bør forbedres. Rammeverket for menneskeorientert design har altså i stor grad bidratt til å oppnå suksesskriteriene og utforske problemstillingen til oppgaven.

Kapittel 9

Utviklingsprosess

Dette kapitlet beskriver valget av utviklingsprosess som ble brukt i utviklingen av applikasjonen tilknyttet dette prosjektet, og hvorfor denne ble valgt.

9.1 Utviklingsmetodikk

Innenfor programvareutvikling er det i hovedsak to fremtredende utviklingsstrategier, nemlig fossefallmodellen og den smidige modellen. Fossefallmodellen, introdusert av Winston W. Royce i 1970 [28], er en tradisjonell og lineær utviklingsmetodikk. Modellen består av et sett med faser, hvor hver fase fullføres før den neste fasen begynner. Smidig utvikling, formalisert i 2001 [29], er en fleksibel og iterativ tilnærming til programvareutvikling. I denne modellen er det fokus på samarbeid, rask prototyping og jevnlig dialog mellom utviklerne og kundene.

Det er fordeler og ulemper med begge disse tilnærmingene til programvareutvikling. Fossefallmetoden har en klarere struktur med distinkte faser, mens fleksibiliteten i den smidige metoden gjør at det er enklere å fortløpende gjøre endringer gjennom hele utviklingsprosessen. Dette er en fordel når det er knyttet usikkerhet til hvor lang tid utviklingen av produktet vil ta. Risikoen med fossefallmetoden er at man ikke rekker å komme gjennom alle fasene av utviklingen, og dermed står uten et produkt ved slutten av prosjektet. Ved å gå for en smidig utviklingsstrategi kan man tilpasse forventningene til produktet fortløpende, og på den måten sørge for at man er garantert at man har et produkt ved slutten av prosjektet, selv om produktet ikke nødvendigvis oppfyller alle behovene til brukergruppen.

En annen grunn til at fossefallmetoden fremsto som lite attraktiv i dette prosjektet var utviklerens kompetanse. Utvikleren hadde ved prosjektets begynnelse ikke tilstrekkelig kompetanse om de aktuelle teknologienes fordeler, ulemper og begrensninger til å si med sikkerhet hva som kunne og burde brukes til å imøtekomme kravene presisert i kapittel 7. Begrenset kompetanse gjør det vanskelig å finne de beste løsningene, da ikke alle de mulige løsningene er kjent. Et eksempel på dette er behovet for en nettside tilknyttet applikasjonen. Dette var i utgangspunktet en del av prosjektet, hvor nettsiden skulle vise gjennomførte oppsynsturer og generere rapporter. Kort tid etter at utviklingen

ble igangsatt, var det tydelig at all denne funksjonaliteten kunne legges inn direkte i mobilapplikasjonen, og behovet for en egen nettside ble vurdert som ikke lenger gjeldende. Ved å bruke en smidig tilnærming kan slike tilpasninger gjøres fortløpende, underveis i prosessen, og en slipper å være låst til en plan lagt før utviklingen starter. Ved å ta disse avgjørelsene i løpet av utviklingsprosessen, og endre dem ved behov, kan en i større grad sikre at valgene en tar er informerte og godt begrunnet.

Basert på de beskrevne fordelene og ulempene med fossefall- og smidig metodikk, samt usikkerheten og behovet for fleksibilitet som er tilknyttet prosjektet, ble det valgt å benytte en smidig utviklingsstrategi. Den smidige metoden ga mulighet for å håndtere endringer og tilpasse prosjektet etter hvert som utviklerens kompetanse økte og nye muligheter og løsninger ble oppdaget. En smidig metodikk ble dermed ansett som den best egnede metodikken til dette prosjektet.

9.2 Konkrete smidige metoder

To nøkkelkonsepter som brukes aktivt i dette prosjektet er hurtig prototyping og et fokus på å skape en MVP (minimum viable product). Hurtig prototyping går ut på å raskest mulig lage en prototype som fungerer, selv om den i liten grad oppfyller kravene til produktet. Deretter jobber man iterativt med denne prototypen for å tilføre ny funksjonalitet. Denne fremgangsmåten gjør at man oppdager potensielle problemer tidlig, så de kan løses før de skaper store utfordringer. En MVP er den mest grunnleggende versjonen av et produkt som fortsatt oppfyller essensen av kravene som stilles til produktet. Mens hurtig prototyping gjør at man raskt oppdager problemer og kan verifisere antakelser, tillater en MVP at man i løpet av kort tid kan begynne å teste produktet. Ved å raskt teste kjernefunksjonaliteten til applikasjonen, kan man verifisere at prototypens funksjonalitet møter brukerens behov, og gjøre nødvendige endringer før disse blir kompliserte og tidkrevende. Fokuset på å lage en MVP bidrar også til å sikre at det ved prosjektets slutt eksisterer et produkt som i det minste oppfyller noen av de grunnleggende kravene som stilles til produktet.

Det er vanlig at et prosjekt basert på smidig utviklingsmetodikk deler prosjektet opp i flere sprints på kortere tidsintervall, gjerne to til fire uker. Hver sprint inneholder en demonstrasjon for kunden, hvor utviklerne kan få tilbakemeldinger på produktet og valgene de har tatt siden forrige demonstrasjon. Kortere sprints sørger også for at alle utviklerne på teamet har oversikt over hvem som jobber på hva, og det er lettere å ivareta god kommunikasjon med jevnlig statusoppdateringer. I motsetning til typiske smidige prosjekter, består dette prosjektet bare av én utvikler, og det har heller ikke en enkelt kunde. Dette prosjektet har derfor brukt en noe løsere struktur, med jevnlig møter mellom utvikler og veileder, hvor foreløpig progresjon evalueres og veien videre diskuteres.

9.3 Versjonskontroll

For å sørge for at utviklingen av applikasjonen ble gjort på en oversiktlig og pålitelig måte, ble versjonskontroll i form av GitHub tatt i bruk. GitHub er en plattform for git-basert versjonskontroll, hvor koden også lagres på nett og er tilgjengelig så lenge du er logget inn [30]. GitHub har også en rekke verktøy for automatisering av prosesser, samarbeid om kode, osv. Da dette prosjektet består av bare én utvikler, var motivasjonen for å bruke GitHub todelt.

Hovedmotivasjonen var at koden skulle være tilgjengelig på nett, og dermed sikkerhetskopiert, i tilfelle det skulle skje noe med maskinen koden ble skrevet på. Den andre faktoren var en oversikt over den historiske koden og hvilke endringer som ble gjort i løpet av prosjektet. Dette sikret at valg som ble tatt kunne gjøres om og endres tilbake hvis de ikke fungerte som antatt, og at tidligere versjoner av koden var tilgjengelige hvis det på ulike tidspunkt var deler av applikasjonen som ikke fungerte, men som hadde fungert tidligere. En bonus ved å bruke versjonskontroll også når prosjektet bare består av en utvikler, er at det legger opp til at prosjektet kan utvides eller tas over av andre på en mest mulig sømløs måte. Tilgang til versjonshistorikken kan være essensiell for å forstå hvorfor koden ser ut som den gjør på det nåværende tidspunktet, hvilke valg som har blitt tatt tidligere, og hvor og når eventuelle feil har oppstått.

Kapittel 10

Programvarearkitektur

Programvarearkitekturen til en applikasjon spiller en viktig rolle når det kommer til applikasjonens ytelse, skalerbarhet og hvor enkelt det er å holde applikasjonen ved like [31]. Applikasjonen som ble produsert i løpet av dette prosjektet bygger på teknologiene Expo, tRPC, Next.js, Prisma, Planetscale og Clerk. Disse teknologiene ble valgt for å gjøre utviklingen av applikasjonen effektiv, og for at potensiell skalering av antall brukere og mengden data skal fungere sømløst. I dette kapitlet forklares disse verktøyene og funksjonen deres i applikasjonen.

10.1 Oversikt over arkitekturen

Applikasjonens arkitektur er basert på en klient-server-modell, hvor klientsiden bygges med React Native gjennom Expo, og serversiden er laget med Next.js. Kommunikasjonen mellom klient og server gjøres med tRPC, som bidrer med et effektivt og typesikkert (type-safe) *remote procedure call (RPC)*-rammeverk. Prisma brukes for å kommunisere mellom server og database, Planetscale brukes som databaseløsning, og Clerk håndterer brukerautentisering og autorisering.

10.2 React Native og Expo

React Native er et åpen kildekode-rammeverk som kan brukes til å utvikle en applikasjons brukergrensesnitt [32]. Rammeverket er basert på React, og lar utviklere bygge kryssplattform-applikasjoner ved å bruke JavaScript og React. Én kodebase kan altså brukes til å lage en applikasjon som fungerer både på iOS og Android, og dette gjør utviklingen av applikasjonen mer effektiv. Disse faktorene, i tillegg til utviklerens eksisterende kjennskap til Javascript-biblioteket React, gjorde at det var naturlig å velge å bygge applikasjonen med denne teknologien.

Expo er en åpen kildekode-plattform som lar utviklere bygge og distribuere mobilapplikasjoner med React Native [33]. Expo tilbyr et sett med verktøy og tjenester som forenkler utviklingsprosessen og muliggjør rask prototyping og testing av applikasjoner. Utviklerappen Expo Go tillater forhåndsvisning av applikasjonen i sanntid på mobilen, som gjør det enkelt å teste og feilsøke applikasjonen under utvikling. Expo tilbyr også tjenester for å bygge applikasjonen før publisering til de aktuelle app-markedsplassene, og mulighet til

å oppdatere applikasjonen over-the-air (OTA) [34]. Denne formen for oppdateringer fungerer uten at brukerne trenger å laste ned en ny versjon av applikasjonen fra App Store eller Google Play, og gjør dermed også at oppdateringer ikke må godkjennes av Apple eller Google før de kan tas i bruk. Dette kan være veldig nyttig hvis en oppdatering inneholder en kritisk sikkerhetsoppdatering, eller en tidligere oppdatering hadde en feil som gjør at applikasjonen ikke fungerer som den skal, og som raskt må erstattes.

10.3 tRPC

tRPC er et typesikkert og fleksibelt remote procedure call (RPC)-bibliotek for TypeScript [35]. tRPC gjør kommunikasjon mellom klient og server sømløs ved å generere typer og API-klienter basert på serverkoden. Dette fungerer som et abstraksjonslag over protokoller som HTTP og WebSockets. Siden både serverkoden og kildekoden er skrevet i TypeScript, gjør tRPC det enkelt og effektivt å kalle serverfunksjoner fra klienten for å kommunisere med databasen og autentiseringstjenesten. Siden tRPC er typesikkert, reduseres muligheten for å gjøre feil, og koden blir sikrere.

10.4 Next.js

Next.js er et React-rammeverk med mange egenskaper som kan bistå i utviklingen og driften av nettsider [36]. Siden dette prosjektet utelukkende innebærer en mobilapplikasjon er det mange av disse egenskapene som ikke blir brukt. Motivasjonen for å bruke Next.js på tross av dette, er at Next.js tilbyr en API-server, som kan brukes til å kommunisere med databasen og autentiseringstjenesten som brukes i dette prosjektet. API-serveren prosesserer forespørsler fra klienten, og returnerer den forespurte dataen. En stor fordel med å bruke Next.js, er at det tillater bruk av serverlessfunksjoner [37]. Serverless-funksjoner håndterer automatisk serverinfrastrukturen og skaleringen som er nødvendig for at applikasjonen skal fungere uavhengig av antall brukere. Som et resultat kan API-serveren til Next.js brukes uten bekymring for skalering eller bekymring for hvilken maskinvare som trengs for å møte behovene til applikasjonen.

10.5 Prisma

Prisma er en åpent-kildekode ORM (Object-Relational Mapping)-bibliotek for Node.js og TypeScript som forenkler databasetilgang og -administrasjon [38]. Det fungerer som et abstraksjonslag mellom serveren og databasen. Biblioteket tilbyr en typesikker forespørselbygger for utviklere, som gjør at man kan samhandle med databasen uten å trenge å skrive SQL-spørringer manuelt. Responsen fra databasen er også typesikker, og dette gjør det enkelt og sikkert å kommunisere med databasen gjennom serveren. Prisma gir også tilgang til et grafisk brukergrensesnitt som kan brukes til å se og endre data i databasen.

10.6 PlanetScale

Prosjektet bruker databasetilbyderen PlanetScale som datahåndteringsløsning. Det er en serverless, skalerbar relasjonell databaseplattform basert på MySQL og Vitess databaseklyngesystem [39]. PlanetScale tar utgangspunkt i den etablerte teknologien MySQL, men har bygget videre på dette med egenutviklet teknologi for å få MySQL-systemet til å skalere bedre når store mengder data lagres. Dette er ikke spesielt relevant for dette prosjektet, da datamengden som vil bli lagret i denne databasen trolig aldri vil komme opp i en størrelsesorden hvor dette er nødvendig. Grunnen til at PlanetScale likevel ble valgt som databaseløsning er at de har en generøs gratisplan som dekker alle behovene til dette prosjektet, og fordi det enkelt kan brukes sammen med Prisma.

10.7 Clerk

Clerk er plattformen som brukes for å håndtere applikasjonens brukere. Det er en brukeradministrasjonsløsning som håndterer autentisering og autorisering av brukerne som bruker applikasjonen. Clerk tilbyr en API som gjør det enkelt å implementere en autentiseringsprosess som lar brukere logge inn med e-post og passord, sosiale innloggingsmetoder som Facebook, Google og Apple, samt støtte for multifaktorautentisering, tilbakestilling av passord, etc [40]. Ved å bruke Clerk til å håndtere registrering og autentisering av brukere øker sikkerheten til applikasjonen, da Clerk i større grad er i stand til å håndtere denne informasjonen på en trygg måte enn utvikleren av applikasjonen. Det bidrar også til å gjøre applikasjonen mer brukervennlig. Clerk gjør det nemlig enkelt å ta i bruk sosiale innloggingsmetoder, noe utvikleren av hensyn til prosjektes mål og omfang ikke hadde prioritert å implementere ellers.

10.8 Oppsummering

Applikasjonen som ble utviklet som en del av dette prosjektet bruker moderne teknologi som gjør utviklingsprosessen rask og effektiv, og håndterer skalering av brukere og datatrafikk sømløst. Å bruke verktøy som gjør utviklingen av produktet effektivt legger til rette for at mer tid kan brukes på å implementere og teste funksjonalitet. Dette fører videre til at kvaliteten på applikasjonen øker. Sømløs skalering av brukere og datatrafikk gjør også at applikasjonen ikke bare fungerer i teorien, men kan tas i bruk i stor skala. Dette er nødvendig for at produktet skal imøtekomme brukergruppens reelle behov.

Kapittel 11

Prototyper

Dette kapittelet går gjennom de to prototypene utviklet i løpet av prosjektet (en designprototype og en programmert prototype), hvilke valg som ble tatt i utformingen av disse, og hvorfor disse valgene ble tatt. Det går også gjennom endringene gjort fra den første prototypen til den andre, og motivasjonen for disse endringene.

11.1 Designfilosofi

Det er mange hensyn som må tas for å lage et produkt av høy kvalitet. Ved å ha disse klart for seg kan man både jobbe for å oppnå dem og evaluere om man har lykket. Suksesskriterie 1 for dette prosjektet, er som nevnt i kapittel 3 at «Produktet skal være brukervennlig, intuitivt og enkelt å bli kjent med for brukerne». Dette må ivaretas gjennom hele designprosessen, og bør reflekteres i alle prototypene i løpet av prosjektet. For å oppnå dette suksesskriteriet har designprinsippene til Donald Norman, beskrevet i «Design of Everyday Things» [41], blitt brukt som overordnet designfilosofi. Alle prototypene som ble utviklet i løpet av dette prosjektet har tatt utgangspunkt i prinsippene denne filosofien beskriver. Nedenfor følger en gjennomgang av disse prinsippene og hva de innebærer.

11.1.1 Konseptuell modell

En konseptuell modell er en forklaring, ofte forenklet, av hvordan noe fungerer. Eksempler på konseptuelle modeller er batteriikonet på en smarttelefon som tømmes etter hvert som batteriet blir utladet. Batteriet inni telefonen blir ikke fysisk mindre av at batteriet blir utladet, men idéen er den samme. En konseptuell modell er samtidig bare nyttig så lenge antakelsene den bygger på er sanne. Hvis telefonen din skrur seg av når den viser at den har 15% av batterikapasiteten igjen, fordi telefonen blir kald, bryter den konseptuelle modellen sammen. De konseptuelle modellene som er relevante for å lage gode design er også mentale modeller hos brukerne av produktet, og kan dermed være ulike hos ulike mennesker. Den konseptuelle modellen til et produkt kan formes av tilbydelighetene, antydningene, overføringene og begrensningene til produktet. Det er viktig å redusere sannsynligheten for at en bruker lager en konseptuell modell som ikke stemmer, blant annet ved å unngå tvetydige

kontrollmekanismer. Et kjøleskap med en kontrollenhet som går fra 1 til 7 kan tydes både som at tallene representerer hvor kraftig kjøleskapet skal kjøles ned (høyere betyr kaldere), eller som at tallene representerer kjøleskapets temperatur (høyere betyr varmere). En bedre kontrollenhet har nivåene 1 °C til 7 °C eller fra ett til tre snøkrystallikoner, da disse markeringene er entydige (flere snøkrystaller betyr kaldere, grader i Celsius indikerer temperatur) og dermed bare svarer til én konseptuell modell. Et godt design gjør all informasjon som er nødvendig for å lage en god konseptuell modell tilgjengelig for brukeren.

11.1.2 Tilbydelighet (affordance)

Tilbydeligheten mellom et objekt eller et element og en person er hva objektet/elementet tilbyr en person å bruke det til eller gjøre med det. En stol tilbyr en person å sitte i den, mens en knapp tilbyr en bruker å trykke på den. Tilbydeligheten mellom et element og en person kan variere utifra personens forutsetninger (en trapp tilbyr personer å gå opp og ned trappen, men dette gjelder ikke for rullestolbrukere). Det er derfor viktig å ta med brukernes attributter og forutsetninger når man analyserer tilbydeligheten til et objekt eller et element. Tilbydeligheten til et objekt eller et element er kort oppsummert hva objektet/elementet kan brukes til av en gitt person. Et godt design lar brukeren oppnå det ønskede resultatet gjennom tilstrekkelig og relevant tilbydelighet.

11.1.3 Antydning (signifier)

Mens tilbydeligheten til et element beskriver hva en bruker *kan* bruke elementet til, er antydningen til et element hvilken bruk elementet *kommuniserer* at det kan brukes til. Antydninger (signifiers) kan være både bevisste, som et skilt eller en merkelapp, eller ubevisste, som en dør som åpnes ved å skyve den innover med et håndtak som antyder at døren skal trekkes utover. Et godt design sørger for at et produkts tilbydeligheter antydes gjennom designet.

11.1.4 Overføring (mapping)

Overføring er sammenhengen mellom to sett med elementer. Dette kan for eksempel være sammenhengen mellom knapper og hvilke elementer knappene styrer. Naturlig overføring er når overføringen har fysiske sammenhenger eller andre sammenhenger som gjør det åpenbart hvordan overføringen fungerer. En volumspak som har den nederste punktet som det laveste volumnivået og det øverste punktet som det høyeste volumnivået er et eksempel på naturlig overføring. Det samme gjelder scrolling på en smarttelefon. Når en drar fingeren oppover på skjermen følger innholdet med fingeren i samme retning og i samme hastighet. Hva som regnes som naturlig overføring kan variere mellom ulike kulturer, men det er enkelte overføringer som er relativt universelle. Dette gjelder for eksempel fysisk nærhet mellom elementer som henger sammen. Et godt design har en naturlig overføring mellom kontrollelementer og resultatene av å bruke disse.

11.1.5 Tilbakemeldinger

Tilbakemeldinger forteller en bruker at en handling er registrert. Et eksempel er en heisknapp som lyser når du trykker på den, eller et lasteikon som viser at en nettside holder på å laste inn innhold etter at du har trykket på en knapp. Tilbakemeldinger kan komme i ulike former, som lyder, lys eller andre visuelle endringer. For at tilbakemeldingene skal være gode, må de være konkrete og kommunisere tydelig hva som skjer. En heis med en skjerm som viser hvilken etasje den er i, samt en pil som viser om den er på vei oppover eller nedover, er et eksempel på en god tilbakemelding. En fjernkontroll som piper med den samme tonen uavhengig av hvilken knapp du trykker på, er et eksempel på en dårlig tilbakemelding. Et godt design gir fullstendige og kontinuerlige tilbakemeldinger til brukeren om resultatene av handlingene deres og produktets nåværende tilstand.

11.1.6 Begrensninger

Det er fire relevante begrensninger når det kommer til design: Fysiske, kulturelle, semantiske og logiske.

- Fysiske begrensninger gjør det fysisk umulig å gjøre feil valg. Et AAA-batteri kan ofte settes inn i to retninger, men bare én av dem gjør at enheten batteriet settes inn i fungerer. I motsetning har 9V-batterier en sirkulær pol og en sekskantet pol. Dette gjør at batteriet bare kan monteres på én måte, og forhindrer feil.
- Kulturelle begrensninger forhindrer feil valg ved hjelp av kulturelle normer. Du trenger ikke en eksplisitt beskjed eller et signal for å la være å sykle inne på et kjøpesenter, da dette strider med en kulturell norm. Kulturelle normer kan derimot endre seg over tid (Sverige endret hvilken side av veien biler skal kjøre på i 1967, og de kulturelle normene ble dermed også endret).
- Semantiske begrensninger utnytter betydningen av en situasjon for å forhindre handling. En rød, stillestående mann signaliserer til fotgjengere at det ikke er trygt å krysse veien. Forbudt-skilt indikerer at noe, enten en handling (adgang, løping), et objekt (mobiltelefoner, droner) eller en kategori (uvedkommende, aldersgrupper), er forbudt.
- Logiske begrensninger indikerer hvilket valg som er det korrekte ut ifra logikk. Resirkuleringsbokser for pant har ofte runde, små hull, mens restavfall har store, firkantede eller runde hull og papiravfall har lange, smale hull. Både papirark og brusflasker passer i hullet for restavfall, men det er logisk at flaskene skal i hullet som har omtrent samme form og størrelse som flasken, mens papir skal i hullet som har en lignende form som et papirark.

Et godt design utnytter fysiske, logiske, semantiske og kulturelle begrensninger til å veilede handling og bistå i tolkningen av hvordan produktet skal brukes.

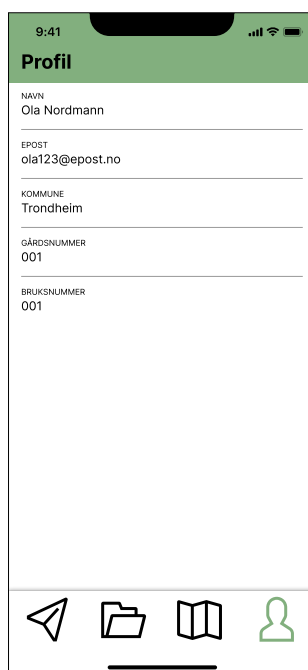
11.2 Prototype 1 - Designprototype

Applikasjonens design i prototype 1 er delt inn i fire kategorier med ulik funksjonalitet. Disse fire kategoriene byttes mellom ved hjelp av fire faner på bunnen av applikasjonen. De fire fanene representerer «ny tilsynstur», «fullførte tilsynsturer», «nedlastede kartutsnitt» og «profil». De fire knappene er synlige så lenge det er relevant å ha tilgang til funksjonaliteten som tilbys av en av de kategoriene som ikke for øyeblikket er valgt, og gjør det enkelt å bytte ulike kategorier.

Gjennomgående i designet er bruken av store knapper for hovedfunksjonalitet som beskriver med tekst hva de gjør. Dette er så det skal være enkelt å trykke på dem uavhengig av omstendighet. Disse er også fargekodet så det er lettere å skille dem fra hverandre, og så det skal være lett å identifisere hva de gjør (grønn for navigasjon, opprettelse, sette i gang prosesser, etc, rød for avbryt, slett, etc). Det er også mye bruk av ikoner. Fordelen med ikoner er at det kan gå raskere å forstå hva de betyr enn hvis de hadde vært erstattet med tekst, og de tar også ofte mindre plass. Disse grepene er med på å antyde hvilken tilbydelighet hver skjerm i applikasjonen har.

11.2.1 Profil

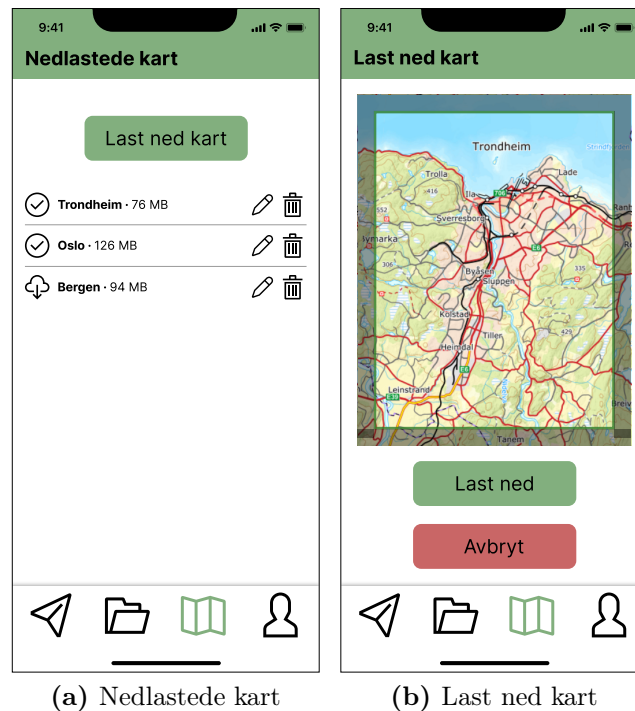
Profilsiden, vist i figur 11.2.1, består av navnet, e-postadressen, kommunen, gårdsnummeret og bruksnummeret til brukeren av applikasjonen. Denne informasjonen benyttes når en bruker skal starte en ny tilsynstur, og gjør at brukeren ikke trenger å registrere denne informasjonen manuelt for hver tur. Ved å lagre denne informasjonen blir prosessen med å starte en ny tilsynstur mer effektiv.



Figur 11.2.1: Fane 4: Profil

11.2.2 Nedlastede kart

Siden med nedlastede kart (figur 11.2.2a) består av en liste med kart brukeren har lastet ned. Ikonet på høyre side viser om kartet er ferdig lastet ned (sirkel med en hake) eller holder på å lastes ned (sky med pil som peker nedover). Dette er eksempler på tilbakemeldinger til brukeren som viser hva som foregår. Til høyre for ikonet vises kartets tittel og filstørrelse. Søppeldunkikonet på høyre side lar brukeren slette kart som ikke lenger er til nytte, og blyantikonet lar brukeren endre kartutsnittet som lastes ned. Disse ikonene følger normene for ikonbruk sletting og redigering, noe som trolig vil stemme med brukerens konseptuelle modell, og antyder til brukeren at disse handlingene er mulige. Når brukeren trykker på «Last ned kart»-knappen, navigerer applikasjonen til skjermen vist i figur 11.2.2b. Her kan brukeren laste ned et nytt kartutsnitt eller avbryte operasjonen. Hvis brukeren trykker på blyantikonet på figur 11.2.2a vises også skjerm 11.2.2b, men brukeren redigerer nå et kart som allerede er lastet ned i stedet for å konfigurere nedlastingen av et nytt kart.

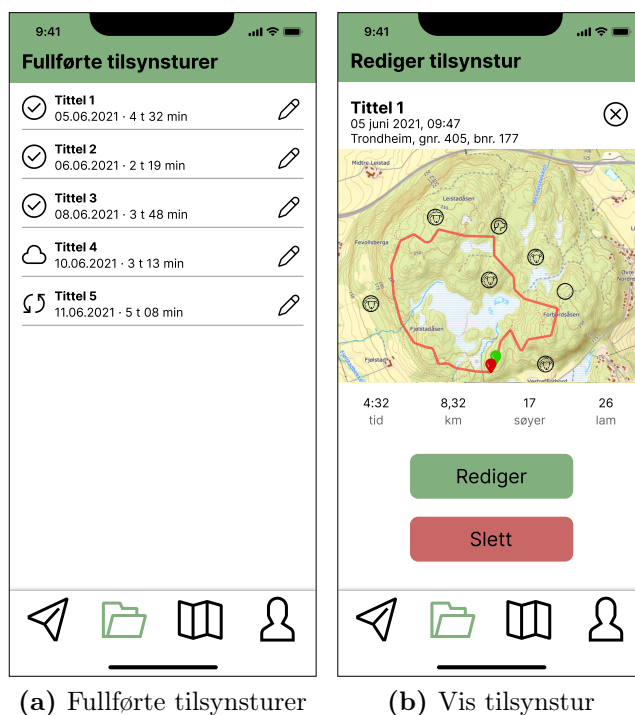


Figur 11.2.2: Fane 3: Kart

11.2.3 Fullførte tilsynsturer

Fane 2 (figur 11.2.3a) viser alle de fullførte tilsynsturene i en liste. Ikonet på venstre side viser om turen er tilgjengelig på enheten (sirkel med en hake), lastet opp til skyen (sky) eller holder på å lastes opp (to piler som går mot klokka). Dette gir brukeren innsikt i hva applikasjonen gjør, og hjelper brukeren å lage en korrekt konseptuell modell av applikasjonen. I tillegg gir det tilbakemeldinger til brukeren om tilstanden til applikasjonen. Hvert element i lista inneholder også tittelen til turen, datoen turen ble gjennomført på, og hvor lang tid turen varte. Til høyre er det et blyantikon som kan trykkes på for å redigere en gjennomført tilsynstur.

Ved å trykke på et element i lista utenom blyantikonet blir du sendt til skjermen vist i figur 11.2.3b. Denne viser mer informasjon om tilsynsturen du trykket på, inkludert starttidspunkt, kommune, gårds- og bruksnummer, et kartutsnitt som viser ruten brukeren beveget seg i løpet av turen, og observasjonene som er registrert (vist med ikoner på kartutsnittet). Du får også se lengden til turen i kilometer, og applikasjonen viser hvor mange søyer og lam som ble registrert. Det er også knapper som gir mulighet til å redigere eller slette tilsynsturen, og disse er fargekodet i tråd med kulturelle normer.

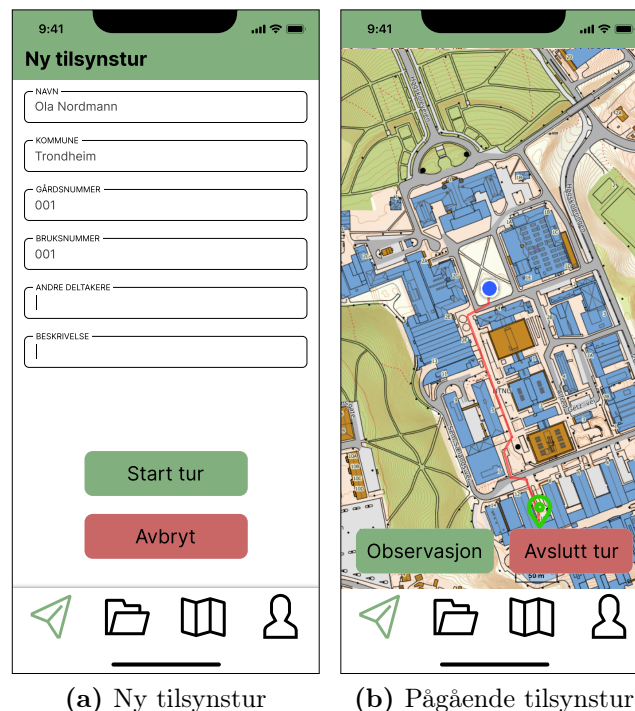


Figur 11.2.3: Fane 2: Fullførte tilsynsturer

11.2.4 Ny tilsynstur

Siden for å starte en ny tilsynstur (figur 11.2.4a) ligger under fane 1, og har felter for å skrive inn navn, kommune, gårdsnummer, bruksnummer, andre deltakere og en beskrivelse av turen. De av disse feltene som har verdier lagret under «Profil»-siden (figur 11.2.1) er forhåndsutfyllt, men kan endres. Feltene viser alltid både tittel på feltet og informasjonen som er fylt inn i feltet, og teksten oppdateres fortløpende etter hvert som brukeren taster inn informasjon. Dette signaliserer til brukeren at verdiene kan endres, og gir fortløpende tilbakemelding om at applikasjonen registrerer det brukeren skriver inn. Feltene som skal inneholde tall, presenterer brukeren et tastatur som bare består av tall i stedet for det vanlige tastaturet. Dette begrenser fysisk hvilke verdier brukeren kan taste inn i feltet.

Når brukeren trykker på «Start tur»-knappen, navigerer applikasjonen til skjermen vist i figur 11.2.4b. Denne skjermen består av et utsnitt av et digitalt kart som viser posisjonen til brukeren i tillegg til en rød strek som viser ruten brukeren har beveget seg siden turens start. Kartet har støtte for «pinch to zoom», og kan flyttes ved å dra det rundt med fingeren. Dette tilsvarer funksjonalitet som er tilstede i andre digitale kart, og samsvarer med brukerens konseptuelle modell av digitale kart. I tillegg til kartet presenteres brukeren med to knapper: En for å registrere en observasjon og en for å avslutte tilsynsturen. Knappene er fargekodet så det skal være lett å forstå hvilken knapp som avslutter turen. Når brukeren trykker på «Avslutt tur»-knappen, navigerer applikasjonen til skjerm 11.2.3a, hvor den fullførte turen vises. På den måten får brukeren en tilbakemelding om at turen er avsluttet og registrert i applikasjonen.

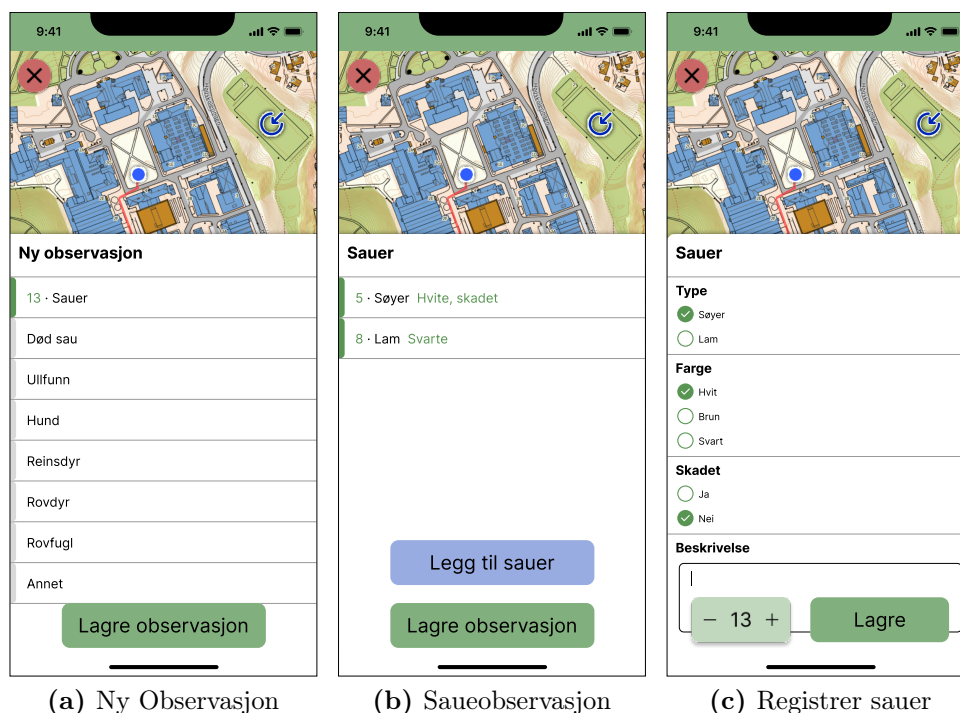


Figur 11.2.4: Fane 1: Ny og pågående tilsynstur

11.2.5 Observasjon

Når brukeren trykker på «Observasjon»-knappen i figur 11.2.4b, navigerer applikasjonen til side 11.2.5a. Her presenteres brukeren med en liste med informasjonen som kan registreres i observasjonen. Ved å trykke på ett av elementene i lista får brukeren mulighet til å registrere informasjon tilknyttet denne typen observasjon. Hvis brukeren trykker på «Sauer», navigerer applikasjonen til side 11.2.5b, hvor brukeren kan legge inn et antall sauer med forskjellige attributter. Hvis brukeren trykker på ett av de andre feltene, navigerer applikasjonen til en side tilsvarende 11.2.5c, hvor brukeren blir presentert med alternativer som er relevante for den gitte observasjonstypen.

Motivasjonen for å presentere brukere med en annen skjerm når de skal registrere sauer, er fordi det innad i én gruppe sauer er flere ulike undergrupper med sauer som tilhører ulike kategorier (søyer og lam, ulik farge på ull, skadet eller frisk). Det er relevant for bønder å kunne registrere hvilke kategori de ulike sauene hører hjemme i, og det må derfor være mulig å legge til flere grupper med sauer. Dette er ulikt fra for eksempel en rovdyrobservasjon. Rovdyr kan også observeres i grupper, men her er det ikke behov for å kategorisere de ulike rovdyrere i gruppa.



Figur 11.2.5: Registrer observasjon

På skjerm 11.2.5b kan brukeren av applikasjonen legge til et valgfritt antall sauer med ulike attributter. «Legg til sauer»-knappen sender applikasjonen til skjerm 11.2.5c, og her kan brukeren velge om sauene er søyer eller lam, hvilken farge de har, om de er skadet og hvor mange sauer som registreres. Antallet endres med pluss- og minusikonene på høyre og venstre side av tallet. Orienteringen stemmer

med den kulturelle normen for tallinjer, hvor tallene øker i høyre retning og synker i venstre retning. Minusikonet blir grått og slutter å fungere når tallet på sauer er «1», da det ikke er mulig å registrere 0 eller et negativt antall sauer. Dette er i tråd med prinsippet om antydning og begrensninger. All informasjonen som kan registreres om en sau er umiddelbart synlig, så hele observasjonens tilbydelighet antydes til brukeren. Hver informasjonskategori (type, farge og skadet) har også en forhåndsvalgt verdi, noe som forhindrer brukeren i å registrere ugyldige observasjoner hvor for eksempel «type» hverken er «søyer» eller «lam».

Både skjerm 11.2.5a og 11.2.5b viser også informasjonen som er registrert etter at brukeren har trykket «Lagre» på skjerm 11.2.5c. Skjerm 11.2.5b viser antall sauer som er registrert i hver kategori, om det er søyer eller lam, hvilken farge de har og om de er skadet eller ikke. Skjerm 11.2.5a viser ikke all denne informasjonen, men lar fortsatt brukeren se hvor mange sauer som er registrert totalt, og dette gjelder også for de andre typene informasjon som kan registreres i observasjonen. Dette designet lar brukere registrere flere ulike typer «observasjoner» (for eksempel ullfunn og hund) i samme app-observasjon, og designet forsøker å gjøre det tydelig hvilken informasjon som er lagt inn i en observasjon. Dette er gjort både ved å vise antall registrerte dyr, og ved å endre fargen på feltet til venstre i lista til grønn for de elementene som inngår i observasjonen.

11.3 Prototype 2: Expo-prototype

Prototype 2 bygger i stor grad på designet for prototype 1, men det er også gjort noen markante endringer. I tillegg er ikke all funksjonaliteten som er visualisert i prototype 1, implementert i prototype 2. Applikasjonen er fortsatt bygget opp rundt de fire fanene, men det er lagt til et sett med skjermer for å logge inn/registrere en ny bruker som man kommer til først. Logikken for å laste ned kart er ikke implementert i prototype 2, så fane 3 består bare av en fylltekst. De fire fanene har blitt fjernet når en tilsynstur pågår, da det på dette tidspunktet ikke bør være nødvendig å ha tilgang til de andre fanene i applikasjonen. Bruken av store knapper og ikoner har i stor grad blitt videreført. Applikasjonen som er utviklet har fått navnet «Beitespor», og oppgaven vil tidvis referere til dette navnet videre i oppgaven.

11.3.1 Autentisering

Når du åpner prototype-applikasjonen blir du først møtt med en skjerm som ber deg logge inn eller opprette en konto (figur 11.3.1a). Applikasjonen gir deg mulighet til å logge inn med Google, Apple, Facebook eller med e-post og passord. Ved å trykke på «Logg inn med Google/Apple/Facebook» blir du sendt til nettleseren din, som verifiserer at du er logget inn med den valgte kontoen, og at du har en Beitespor-konto tilknyttet denne kontoen. Hvis dette ikke er tilfellet får du spørsmål om du ønsker å opprette en ny konto. Hvis du ikke har en Beitespor-konto, kan du trykke på «Opprett konto»-knappen, og du blir da sendt til side 11.3.1b. Her har du mulighet til å opprette en konto, enten med Google,

Apple, Facebook eller e-post, og knappene fungerer på en liknende måte som knappene på «Logg inn»-siden. Du kan komme deg tilbake til side 11.3.1a ved å trykke på «Logg inn»-knappen. Ved å implementere autentisering gjennom tredjeparter som Google, Apple og Facebook, senkes barrieren for å opprette en konto, og det gir brukerne mulighet til å logge inn uten å trenge å huske et passord. Dette er med på å gjøre applikasjonen mer brukervennlig.

(a) Logg inn
(b) Ny bruker

Figur 11.3.1: Autentiserings skjerm

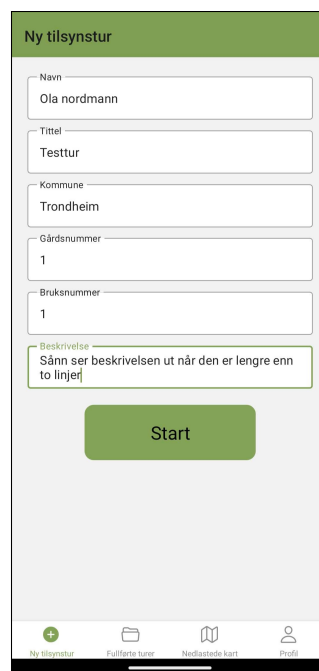
11.3.2 Ny tilsynstur

Når brukeren har logget inn, navigerer applikasjonen til skjerm 11.3.2, hvor brukeren har mulighet til å starte en ny tilsynstur. Dette designet er nesten uendret fra design 1, og de eneste forskjellene er at «Avbryt»-knappen er fjernet, «Andre deltakere»-feltet er fjernet, og «Tittel»-feltet er lagt til. «Avbryt»-knappen ble fjernet da det strengt tatt ikke er noe å avbryte før brukeren har startet en ny tilsynstur. «Andre deltakere»-feltet ble fjernet da den planlagte funksjonaliteten tilknyttet dette feltet ikke er implementert. «Tittel»-feltet er lagt til for å kunne definere tittelen som vises på «Fullførte tilsynsturer»-skjermen. Alle feltene på skjermen er redigerbare, og feltene «Navn», «Kommune», «Gårdsnummer» og «Bruksnummer» har i utgangspunktet de samme verdiene som informasjonen registrert på «Profil»-siden. Disse feltene kan også redigeres så de inneholder andre verdier.

Alle feltene er valgfrie å fylle inn, da dette gir mest mulig fleksibilitet til brukeren. Om dette er hensiktsmessig bør senere vurderes i forhold til kravene til dokumentasjon fra tilsynsmyndighetene. For eksempel kommune, gårdsnummer og bruksnummer er nødvendig informasjon for å identifisere saueieren, og dette

kan være nødvendig informasjon å ha med i en rapport. Hvis denne informasjonen er nødvendig, bør applikasjonen begrense brukeren fra å la disse feltene stå tomme før en ny tur settes i gang.

Applikasjonen har i prototype 2 de samme fanene som i prototype 1, nemlig «Ny tilsynstur», «Fullførte turer», «Nedlastede kart» og «Profil», men utseende på fanene har endret seg. Ikonene er byttet ut med ikoner som i større grad henger sammen med innholdet i fanen, og det er lagt til en tekst som forklarer hva hver fane gjør under hvert ikon. Dette gjør det enklere å forstå hvilken tilbydelighet applikasjonen har, og fanene antyder tydeligere hvilken funksjonalitet de har. I prototype 2 er det også innholdet til ikonet som fargelegges i den valgte fanen, i stedet for omrisset, noe som gjør det enda mer tydelig hvilken fane som er valgt.



Figur 11.3.2: Ny tilsynstur

11.3.3 Pågående tilsynstur

Etter å ha trykket på «Start»-knappen på skjerm 11.3.2, blir brukeren sendt til skjerm 11.3.3a. Denne skjermen har blitt betydelig endret fra prototype 1 (figur 11.2.4b). Det er flere grunner til dette. Av hensyn til overføringen mellom brukerens handling og reaksjonen til applikasjonen har registrering av nye observasjoner blitt endret fra en egen knapp til at brukeren trykker på kartet. Plasseringen til denne observasjonen blir da der brukeren trykket, og denne implementeringen er i større grad i tråd med prinsippet om naturlig overføring. Ulempen med denne endringen er at det kan være vanskeligere for brukeren å forstå hvordan en ny observasjon skal registreres, da applikasjonen i mindre grad antyder tilbydeligheten for å registrere nye observasjoner. Dette er noe som må undersøkes i brukertesten av applikasjonen. Knappen som lar brukeren avslutte tilsynsturen er midtstilt og har fått grønn farge, men er ellers uendret. Det er også lagt til en knapp øverst i venstre hjørne som kan brukes til å avbryte en

tilsynstur, hvor all registrert informasjon blir slettet. Hvis brukeren trykker her, navigerer applikasjonen tilbake til skjerm 11.3.2. Avbryt-knappen er rød, liten og på en mindre synlig plass enn «Avslutt tilsynstur»-knappen, da dette reduserer sannsynligheten for at en bruker trykker på denne knappen ved et uhell. Når brukeren avslutter en tilsynstur, navigerer applikasjonen til skjerm 11.3.10a, og den fullførte turen vises på bunnen av lista.



(a) Nystartet tilsynstur (b) Tilsynstur med rute

Figur 11.3.3: Pågående tilsynstur

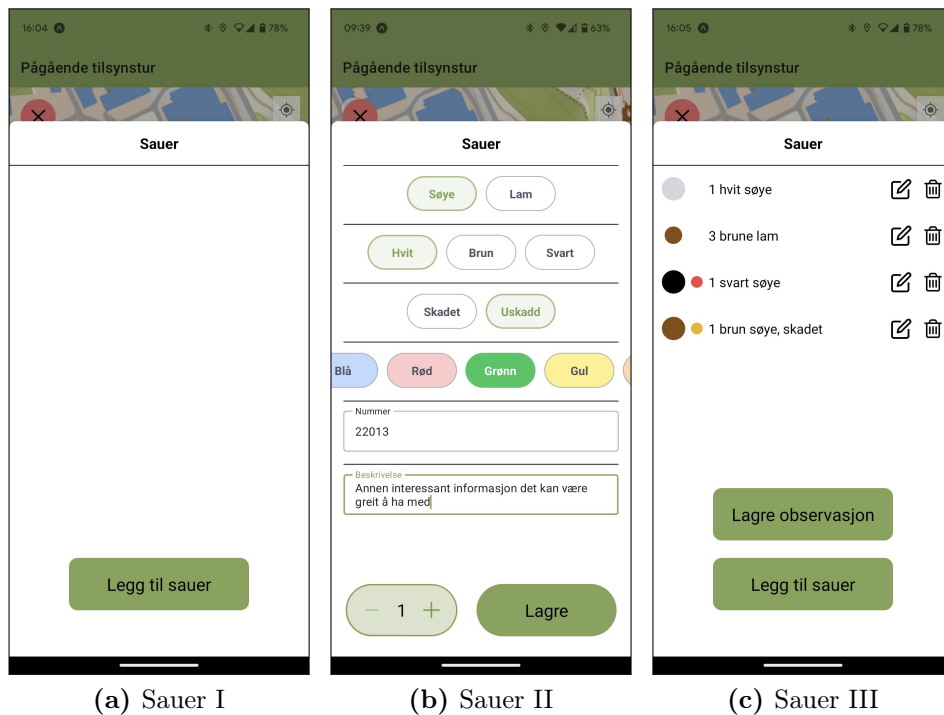
11.3.4 Observasjoner

Når brukeren i løpet av en tilsynstur trykker på kartet, skyves det opp et kort fra bunnen av skjermen som vist i figur 11.3.4, og brukeren kan så velge hva slags observasjon vedkommende ønsker å registrere. Dette er også en skjerm som har blitt endret fra prototype 1 (figur 11.2.5b). Etter evaluering av prototype 1 med veileder ble det klart at det ikke er nødvendig å tillate registrering av mer enn én type observasjon på en gang. Lista i prototype 1 har derfor blitt erstattet med en knapp for hver kategori. Det er også et annet sett med kategorier å velge mellom her enn i prototype 1. «Rovdyr» og «rovfugl» har blitt samlet i én kategori, mens kategoriene «hund», «reinsdyr» og «annet» har blitt fjernet. Grunnen til at enkelte kategorier ble fjernet, er at observasjoner innenfor disse kategoriene er mindre relevante med tanke på rapportering (se vedlegg A.1). All funksjonalitet som ikke var sentral til applikasjonens primærhensikt ble nedprioritert i utviklingen av prototype 2, og disse observasjonstypene ble derfor ikke implementert.

Når brukeren trykker på «Sauer»-knappen, endrer kortet utseende til en tom liste med overskriften «Sauer», og en knapp hvor det står «Legg til sauer», vist i



Figur 11.3.4: Ny observasjon



Figur 11.3.5: Observasjon av sauer

figur 11.3.5a. Ved å trykke på «Legg til sauer»-knappen blir brukeren sendt til skjerm 11.3.5b, hvor brukeren kan registrere en gruppe med sauer.

Funksjonaliteten til denne siden er nesten identisk som i prototype 1 (figur 11.2.5c), men siden ser ganske ulik ut. Brukeren har fortsatt mulighet til å velge om sauene som skal registreres er søyer eller lam, hvilken farge de har og om de er skadet eller ikke, men knappene har blitt større for å gjøre det lettere å trykke på dem. Det er også lagt til en mulighet for å registrere hvilken farge sauene har på slipset, samt et felt for å registrere sauens individnummer. Feltet for slipsfarge og individnummer er valgfrie å fylle inn, da det ikke alltid er lett å observere disse hvis observasjonen av sauene gjøres på avstand. Ved å gi brukerne mulighet til å registrere slipsfargen til sauene i applikasjonen, er det mulig å kontrollere om søyene har med alle lammene sine i flokken, eller om noen av dem mangler [42]. Dette er også med på å ivareta anbefalingen i rapportskjemaet fra Vestland (vedlegg A.1).

Når brukeren trykker på «Lagre»-knappen, navigerer applikasjonen tilbake til lista med sauer lagt til i observasjonen, og sauene som nettopp ble registrert vises på bunnen av lista (et eksempel på hvordan dette ser ut vises i figur 11.3.5c). Dette designet er også endret fra prototype 1 (figur 11.2.5b). Hver sau har en farget disk på venstre side i lista, som viser om sauene er søyer eller lam (størrelsen på disken) og fargen på sauene (fargen på disken). Denne informasjonen kommer også fram av teksten til høyre for disken. For noen av sauene er det en mindre disk til høyre for den største disken, og denne viser om sauene har registrert en slipsfarge. På denne måten kan man se den mest grunnleggende informasjonen om sauene som er registrert i observasjonen enkelt og oversiktlig. Dette gir også brukeren tilbakemelding om at sauene er riktig registrert av applikasjonen, og det gjør det enkelt å se om det er sauer som ikke er registrert (hvis det er en svart sau i flokken, og ingen av de venstre sirklene er svarte, er ikke sauene registrert). «Lagre observasjon»-knappen dukker først opp når brukeren har registrert minst én sau, da det ikke gir mening å lagre en observasjon hvor det ikke er registrert noen informasjon. Dette forhindrer brukeren i å ta valg som ikke er i tråd med tilbydeligheten til applikasjonen. Når brukeren trykker på «Lagre observasjon»-knappen, navigerer applikasjonen tilbake til kartet, og observasjonen som brukeren nettopp lagret vises på kartet (se figur 11.3.9 for hvordan de ulike observasjonstypene ser ut på kartet).

De andre observasjonene følger de samme prinsippene som beskrevet ovenfor, men med litt andre alternativer. Det er bare for observasjoner av sauer at det er mulig å registrere flere ulike typer sauer i én observasjon. Dette er fordi det under utviklingen av applikasjonen ble vurdert som usannsynlig at en bruker kommer over flere døde sauer, flere typer ull eller flere rovdyr av ulik art på samme sted. Når en død sau, et ullfunn eller en type rovdyr er registrert i en observasjon, er det derfor bare en knapp tilgjengelig, nemlig «Lagre observasjon» (se figur 11.3.6b, 11.3.7b og 11.3.8b). Det er heller ikke mulig å endre antall når en registrerer et ullfunn eller en død sau (figur 11.3.7a og 11.3.6a), men antallet kan endres for rovdyrobservasjoner (figur 11.3.8a), da det kan tenkes at man kommer over for eksempel en flokk med ulver. Disse valgene ble gjort for å alltid presentere brukeren med det minimale settet med muligheter for å kunne

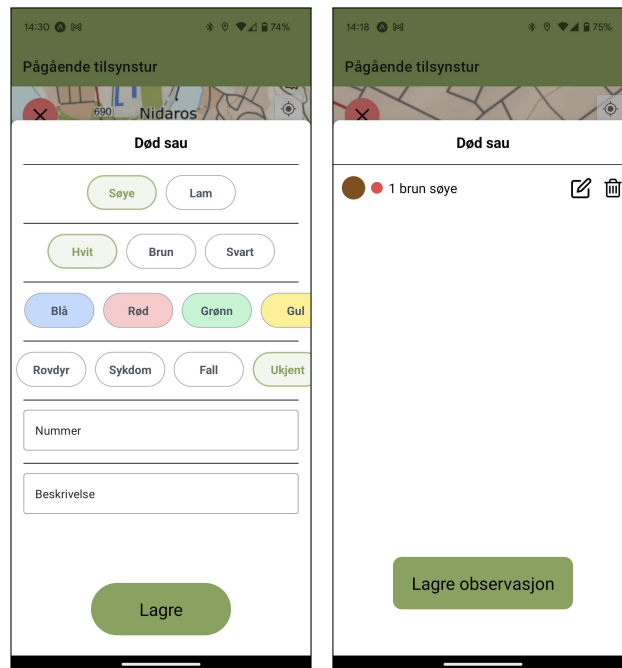
registrere relevant informasjon for alle observasjonstypene, og for å hindre forvirring.

Når det kommer til observasjoner av døde sauer, er muligheten for å registrere om sauen er skadet eller ikke byttet ut med antatt dødsårsak, da dette er informasjonen det er aktuelt å registrere for en død sau. For ullfunn er det bare ett tilgjengelig felt, hvor brukeren kan registrere fargen på ulla, og for rovdyrobbservasjoner er det også bare ett tilgjengelig felt, hvor dette brukes til å indikere hvilken type rovdyr som ble observert. I tillegg har alle observasjonstypene et beskrivelsesfelt hvor brukeren kan legge inn annen relevant informasjon som ikke omfattes av de forhåndsdefinerte feltene.

Når brukeren har registrert en observasjon, vises denne på kartet som et ikon. De ulike ikonene vises i figur 11.3.9. Observasjoner av sauer representeres av ikonet til venstre, som viser en sau i farger. Observasjoner av døde sauer representeres av det øverste ikonet, som viser en sau i svart hvitt, hvor ikonet også er snudd opp ned. Ullfunn representeres av det nederste ikonet, som viser en ulldott. Alle typer rovdyr representeres av ikonet til høyre, som viser et rovdyr fotavtrykk. Hensikten med disse ikonene er at det skal være enkelt å se hvilke observasjoner som har blitt registrert, og hvor de er registrert. Dette stemmer også med overføringen mellom kartet og den virkelige verden, da ikonene dukker opp der brukeren trykker på kartet når han starter registreringen av en ny observasjon, og dette representerer plasseringen til observasjonen i den virkelige verden. Det fungerer også som en tilbakemelding til brukeren på at observasjonene er registrerte.

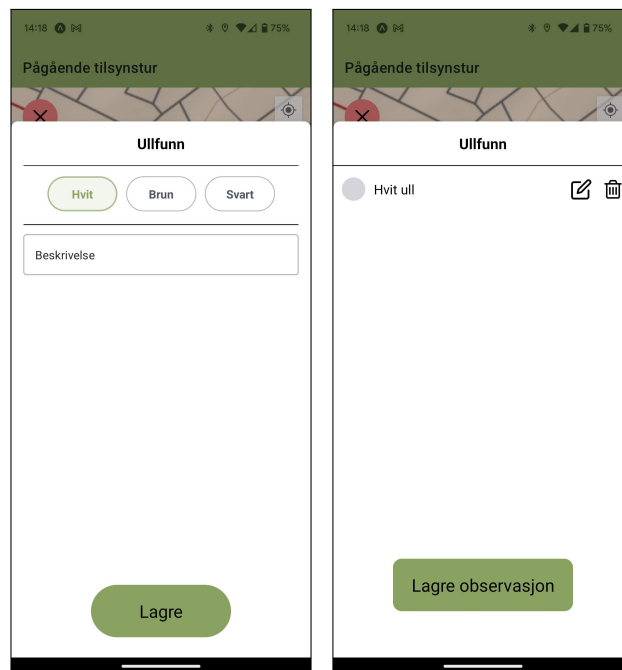
11.3.5 Fullførte tilsynsturer

Når brukeren trykker på «Avslutt tilsynstur»-knappen, lagres informasjonen som ikke ennå er lagret, turen avsluttes, og applikasjonen navigerer til skjerm 11.3.10a. Her vises en liste med turene brukeren har gjennomført, hvor den avsluttede turen vises på bunnen av lista. Designet her er nesten uendret fra prototype 1 (figur 11.2.3a). Blyant-ikonet er fjernet da mulighet for å redigere fullførte tilsynsturer ikke er implementert, og alle turene har det samme ikonet (sirkel med en hake) da funksjonalitet uten internett heller ikke er implementert. I tillegg til dato og varighet for turen er den totale avstanden til turen lagt til for hver tur i lista. Dette ble lagt til da det ble ansett som informasjon det er nyttig å ha enkel tilgang til. Hvis brukeren trykker på en av turene i lista, navigerer applikasjonen til skjerm 11.3.10b. Her vises et kart med oversikt over de ulike observasjonene som ble registrert i løpet av tilsynsturen, samt posisjonsloggen, men det er ikke mulig å hverken redigere eller slette turen i denne prototypen. Det er heller ikke mulig å se mer detaljert informasjon om tilsynsturen, som prototype 1 la opp til i figur 11.2.3b. Designet har ikke blitt bevisst endret på denne skjermen, men resten av funksjonaliteten og visualiseringen av informasjon har blitt nedprioritert av hensyn til annen funksjonalitet. En forbedret versjon av denne skjermen bør implementeres i en neste versjon av applikasjonen.



(a) Død sau I

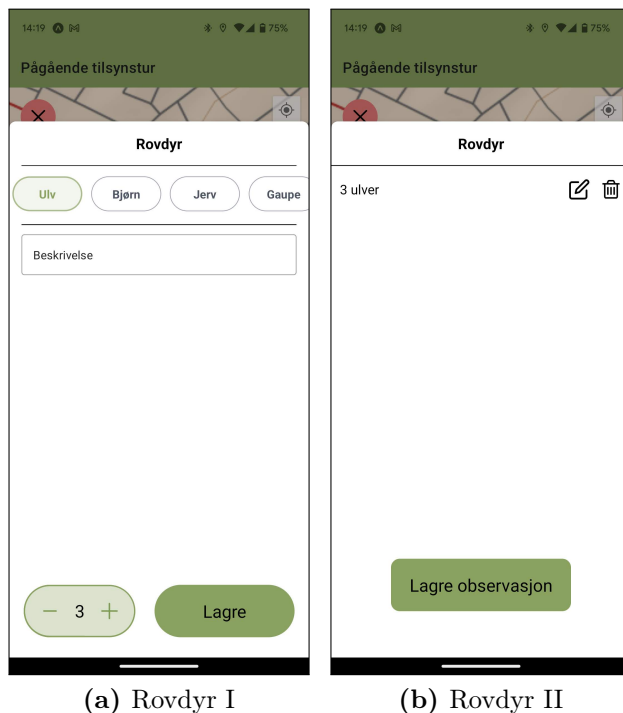
(b) Død sau II

Figur 11.3.6: Observasjon av død sau

(a) Ullfunn I

(b) Ullfunn II

Figur 11.3.7: Observasjon ullfunn



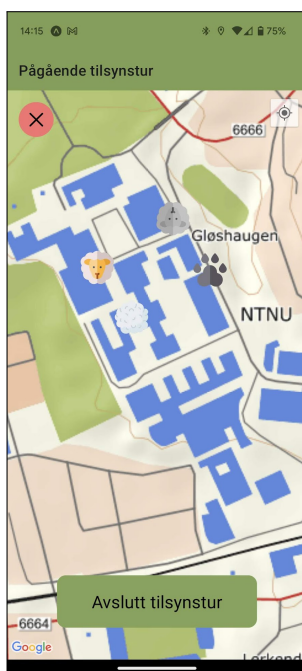
Figur 11.3.8: Observasjon av rovdyr

11.3.6 Nedlastede kart

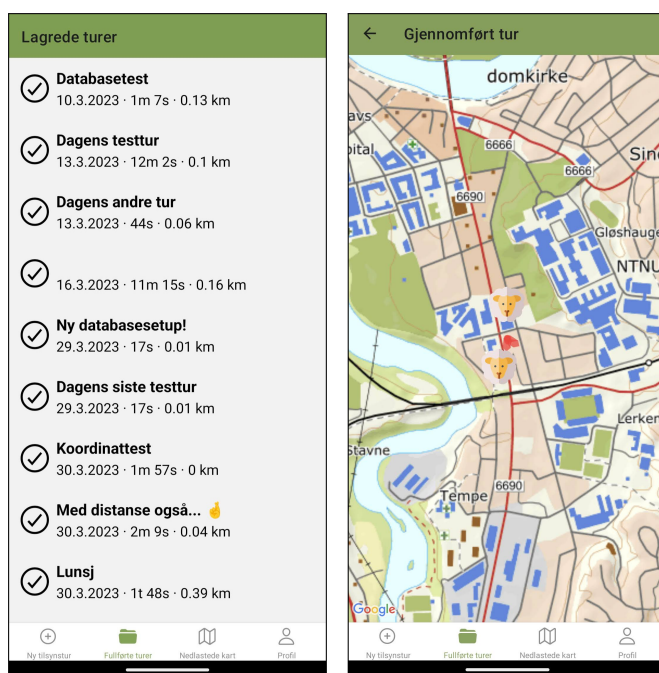
Implementering av nedlasting av kart ble nedprioritert for prototype 2, da dette ble ansett som sentral, men ikke essensiell funksjonalitet, og fordi det ble anslått at testing av resten av applikasjonen ikke ville forhindres av at kartdata bare var tilgjengelig når testenheten var koblet til internett. Denne skjermen inneholder i prototype 2 derfor bare en fylltekst uten funksjonalitet, som vist i figur 11.3.11.

11.3.7 Profil

Profilsiden (figur 11.3.12a) er ganske lik som i prototype 1 (figur 11.2.1), men med noen markante endringer. Det er lagt til to knapper, en «Rediger»-knapp og en «Logg ut»-knapp. «Logg ut»-knappen sender brukeren tilbake til skjerm 11.3.1a, og krever at brukeren på nytt logger inn for å komme tilbake til profilsiden. «Rediger»-knappen er lagt til for at brukeren skal ha mulighet til å redigere informasjonen på profilsiden. Dette var planlagt funksjonalitet allerede i prototype 1, men designet tok ikke høyde for hvordan dette skulle gjøres i praksis. Når brukeren trykker på «Rediger»-knappen, navigerer applikasjonen til skjerm 11.3.12b, hvor brukeren kan endre på alle feltene. Endringene lagres når brukeren trykker på «Lagre», og applikasjonen navigerer tilbake til skjerm 11.3.12a. Endringene som gjøres her endrer også de forhåndsutfylte verdiene på skjerm 11.3.2.



Figur 11.3.9: Observasjoner på kartet



(a) Liste med fullførte turer (b) Åpnet fullført tur

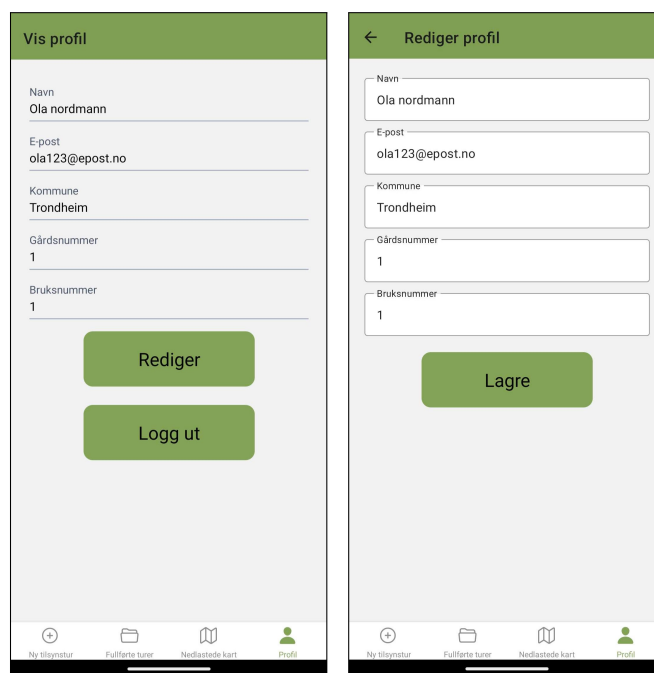
Figur 11.3.10: Fullførte tilsynsturer



Figur 11.3.11: Nedlastede kart

11.4 Oppsummering

Designprinsippene til Don Norman har vært sentrale i utviklingen av både prototype 1 og 2 i dette prosjektet. Ved å fokusere på disse prinsippene gjennom hele utviklingsprosessen og bruke dem for å analysere om konsepter er brukervennlige eller ikke, har designet blitt utviklet med brukeren i fokus. Målet har fra start til slutt vært å utvikle en applikasjon med all den nødvendige funksjonaliteten som skal til for å møte brukernes behov, men med minimal kompleksitet og et selvforklarende design. Både prototype 1 og 2 bærer preg av dette fokuset. Mange av elementene som inngår i applikasjonen har blitt valgt nettopp fordi de signaliserer funksjonalitet til brukeren, har en naturlig overføring og hindrer brukeren i å ta feil valg. I hvilken grad dette fokuset har ført til at applikasjonen er brukervennlig i praksis diskuteres i de to neste kapitlene.



(a) Vis profil

(b) Rediger profil

Figur 11.3.12: Profilskjærmer

Kapittel 12

Brukertest

12.1 Bakgrunn

For å teste i hvilken grad prototype 2 oppfyller suksesskriteriene beskrevet i kapittel 3, og for å kartlegge hvilke elementer av applikasjonen som bør forbedres, ble det utviklet en brukertest for å teste prototypen. Denne testen har som primærhensikt å kartlegge i hvilken grad prototypen imøtekommer suksesskriterium S1, nemlig at «Produktet skal være brukervennlig, intuitivt og enkelt å bli kjent med for brukerne». Sekundært vil testen kartlegge om prototypen oppfyller suksesskriterium S2, som spesifiserer at «Løsningen skal møte bøndenes behov for å registrere informasjon i forbindelse med tilsynsturer». Grunnen til at brukertesting benyttes for å vurdere disse kriteriene, er deres subjektive natur. I hvilken grad et produkt er brukervennlig og intuitivt er vanskelig å bevise objektivt, da disse attributtene primært eksisterer som en relasjon mellom en bruker og et produkt. Mens én bruker kan oppleve applikasjonen som intuitiv, kan en annen brukers opplevelse være at applikasjonen er kompleks og tar lang tid å lære. Ved å gjennomføre en brukertest på en bruker i målgruppen for applikasjonen, kan antakelsene som ligger til grunn for løsningen testes, og detaljer som er oversett kan oppdages.

12.2 Testsubjekt

For å kartlegge hvordan produktet oppleves for brukerne, ble det vurdert som viktig at testen ble utført på en person som er en del av målgruppen for produktet. Testpersonen som ble valgt, Trond Magnus Haugen, har tidligere jobbet som sauebonde på deltid i flere år, og har i løpet av et typisk år hatt rundt 30 sauer på utmarksbeite. Han har dermed innsikt i tilsynsprosessen, og er en relevant person å teste applikasjonen på, da applikasjonen potensielt kunne ha kommet til nytte for han i perioden han drev med sauehold. Haugen har ikke tidligere brukt noe form for applikasjon i forbindelse med tilsynsturer, og hans erfaring representerer dermed en typisk sauebonde.¹ Å bare teste applikasjonen på en bruker gir selvfølgelig begrenset informasjon om hvor godt applikasjonen

¹Beitesnap er den eneste kommersielle applikasjonen for dokumentasjon av tilsynsturer, og utviklingen av denne tjenesten stoppet opp etter 2019 på grunn av få brukere [20, 22]. Det er derfor naturlig å anta at de fleste bønder ikke har brukt en slik applikasjon.

oppfyller kravene for andre individer som inngår i brukergruppen, og applikasjonen skulle gjerne vært testet bredere. Samtidig ble det klart etter gjennomføringen av testen på bare én person at applikasjonen hadde behov for forbedringer på flere områder. Mer detaljert informasjon om hvor godt applikasjonen tilfredsstilte suksesskriteriene hadde blitt vanskelig å oppnå før disse problemene hadde blitt løst. Da prosjektets tidsbegrensninger ikke tillot flere iterasjoner av applikasjonen med påfølgende testrunder, var det ikke aktuelt å teste applikasjonen mer inngående.

12.3 Omfang

Fordi det i dag ikke eksisterer noen kommersielle alternativer til applikasjonen som har blitt forsøkt utviklet i løpet av dette prosjektet, ble det vurdert som naturlig å teste applikasjonen for seg selv, uten å sammenlikne den med alternative måter å registrere informasjon fra tilsynsturer på. Ulike bønder har også svært ulik praksis når det kommer til manuell registrering av informasjon og detaljnivået på denne under tilsynsturer. En test som sammenlikner dagens prosess med prototypen utviklet vil dermed i stor grad kunne påvirkes av hvordan testsubjektet registrerer informasjon manuelt i dag, og gi informasjon som ikke stemmer overens med erfaringen til andre brukere. Dessuten er suksesskriteriene til applikasjonen selvstendige, og vil derfor ikke påvirkes av hvor godt alternative løsninger presterer.

Testen ble utviklet så den kunne gjennomføres uten å bevege seg rundt i naturen. Bakgrunnen for dette valget var en kombinasjon av at posisjonssporingen har blitt testet fortløpende i løpet av utviklingen av applikasjonen, og at posisjonssporing i bakgrunnen ikke fungerer i prototypen som ble testet. Posisjonssporing av brukeren krever ingen interaksjon med applikasjonen fra brukerens side, med unntak av å starte og avslutte tilsynsturen, som uansett vil bli testet. Posisjonssporing når applikasjonen ikke er åpen og aktiv er ikke implementert i prototype 2 på grunn av begrenset tid, og denne fungerer derfor ikke når smarttelefonen er låst eller når Beitespor bare er åpen i bakgrunnen. Testingen ble gjort på en Google Pixel 7, da dette er telefonen applikasjonen har blitt utviklet på, og målet med denne testen var å få tilbakemeldinger fra en bruker på applikasjonens funksjonalitet og brukervennlighet. For å teste at applikasjonen fungerer som tenkt på ulike enheter, er andre testmetoder mer relevante.

12.4 Testmetodikk

Testmetodikken som ble brukt i brukertesten er basert på prinsippene Steve Krug beskriver i boken *Rocket Surgery Made Easy* [43]. Boka ble skrevet med fokus på å teste nettsider, men forfatteren er tydelig på at prinsippene i boka kan overføres til nesten alle typer produkter. Steve Krug argumenterer for at det er viktigere å gjennomføre suboptimale brukertester som gir en viss innsikt i utfordringene brukere har med produktet som testes, enn å risikere å ikke få testet produktet i det hele tatt. Han beskriver også en rekke steg en hver

brukertest bør bestå av:

1. Definer oppgavene brukeren skal løse.

Dette steget gjøres i god tid før testen skal gjennomføres, og går ut på å lage oppgaver som kartlegger om en bruker enkelt er i stand til å gjennomføre kjernefunksjonaliteten til produktet. For applikasjonen utviklet i dette prosjektet er det dermed naturlig å teste at de funksjonelle kravene med høy prioritet som er implementert i prototype 2 fungerer som de skal. Det er viktig at oppgavene er formulert nøytralt, så de ikke gir brukeren ekstrainformasjon som gjør det enklere å løse oppgaven, men bare beskriver oppgaver typiske brukere av applikasjonen vil ønske å kunne løse ved hjelp av applikasjonen.

2. Forberedelser før testen

Dette steget gjøres rett før testen starter, og går ut på å kontrollere at alt er klart før testen gjennomføres. Dette inkluderer å sjekke at applikasjonen er klar til å kjøre på en enhet, at brukeren er i stand til å utføre testen, og at eventuelt utstyr som skal brukes til opptak av interaksjon eller lyd fungerer.

3. Velkommen og introduksjon til testen

Her ønskes testdeltakeren velkommen, prosessen forklares og instruksjoner gis. Her er det flere elementer som er viktig å få fram, så Krug anbefaler sterkt å ha denne delen skrevet ned så detaljer ikke glemmes og formuleringen blir klar og tydelig. Elementer som bør være med i denne delen er

- Hensikten med testen.
- Hvor lang tid testen vil ta.
- At det er produktet som testes, ikke deltakeren.
- At man ikke vil ha mulighet til å bistå deltakeren i løpet av testen.
- En oppfordring til brukeren om å fortløpende forklare hva vedkommende tenker og gjør.

4. Oppvarmingsspørsmål

Å stille noen oppvarmingsspørsmål rett før brukertesten begynner har tre hensikter. For det første gjør det at brukeren blir komfortabel med å prate. For det andre viser det at du er interessert i å høre hva de har å si. For det tredje gir det nyttig informasjon om brukerens tekniske ferdigheter.

5. Gjennomgang av landingssiden

Her instrueres brukeren til å navigere rundt på landingssiden til nettsiden som testes, og gi noen umiddelbare tanker om hva som er hensikten med siden. Dette gir en indikasjon på om landingssiden tydelig formidler det ønskede budskapet til en bruker som nettopp har klikket seg inn på nettsiden. Dette steget er ikke nødvendigvis relevant for applikasjoner, da disse ikke alltid inneholder noe som kan sammenlignes med en landingsside. Når det er sagt har applikasjonens framstilling i applikasjons-markedsplassen en lignende funksjon som en landingsside.

Dette steget kan derfor brukes til å teste om deltakeren forstår hva hensikten til applikasjonen er ved å se på informasjonen som presenteres der.

6. Gjennomføring av testen

Det er i dette steget deltakeren forsøker å gjennomføre oppgavene som ble definert i forkant av testen. Dette gjøres ved at oppgavene leses høyt opp for deltakeren, en etter en. Etter at en oppgave er lest opp, forsøker deltakeren å løse oppgaven. Når en oppgave er løst, eller når det av andre grunner er naturlig å gå videre fra oppgaven, leses neste oppgave opp. I løpet av dette steget kan det også være lurt å notere ned stikkord eller spørsmål som kan stilles til deltakeren etter testen.

7. Oppfølgingsspørsmål

For å ikke forstyrre deltakeren er det lurt å vente med å stille oppfølgingsspørsmål til testen er ferdig. På denne måten slipper man også å bekymre seg for om man gir deltakeren hint om hvordan oppgaven kan løses. Når testen er ferdig er det derimot trygt å stille disse spørsmålene, for å få dypere innsikt i hva deltakeren tenkte eller oppfattet under ulike deler av testen. Dette kan være med på å oppklare deltakerens adferd og unngå usikkerhet i hva som faktisk skjedde i løpet av testen.

8. Avslutning

Si takk til deltakeren for at de deltok i testen, svar på eventuelle spørsmål de måtte ha, og gi kompensasjon for tiden de brukte på å delta i testen hvis dette er avtalt.

12.5 Gjennomføring

Ved gjennomføringen av brukertesten ble stegene beskrevet ovenfor fulgt med noen få unntak. Steg 4, oppvarmingsspørsmål, ble droppet da testdeltakeren og utvikleren som gjennomførte testen allerede kjenner hverandre. Testdeltakeren var dermed allerede komfortabel med å prate og visste at utvikleren som gjennomførte testen var interessert i å høre hva han hadde å si. Informasjon om brukerens tekniske ferdigheter var også kjent på forhånd. Steg 5, gjennomgang av landingssiden, ble også droppet, da applikasjonen ikke har noe som minner om en nettsides landingsside som en skjerm i applikasjonen. Applikasjonen var heller ikke publisert på en applikasjons-markeds plass, og det var dermed heller ikke aktuelt å teste inntrykket av applikasjonen i en slik setting. Oppgavene som ble brukt i brukertesten ligger vedlagt som vedlegg B.1. Nedenfor følger en beskrivelse av hvordan testdeltakeren løste de ulike oppgavene.

1. Opprett en bruker. Registrer kommune, gårdsnummer og bruksnummer.

Deltakeren løste denne oppgaven uten problemer. Opprett konto med Apple ble valgt som alternativ. Dette var mer tungvindt enn det ville vært i en reell situasjon, da deltakeren bruker en iPhone til vanlig. På sin egen telefon er han allerede logget på Apple-kontoen sin. Siden testen ble kjørt

på en Google Pixel 7, måtte han først logge på Apple-kontoen, og fikk deretter muligheten til å opprette en bruker.

2. Start en ny tilsynstur. Tittelen på turen skal være «Demotur», og beskrivelsen skal være «Brukertest av appen».

Også denne oppgaven ble løst uten problemer, men under testen ble flere utfordringer med applikasjonen observert.

- Tallfelt får automatisk verdien 0 hvis de er tomme. Dette skjer fordi det skal registreres et tall i disse feltene, men det var forvirrende for deltakeren at «0» ikke kunne slettes.
- Knappen hvor det står «Start tilsynstur» reagerer ikke når tastaturet vises.
- Kartet starter veldig zoomet ut, så det er vanskelig å se at posisjonen spores uten å zoome lengre inn.
- GPS-posisjoner bruker noen ganger litt tid på å markere akkurat hvor brukeren er. Dette gjør at posisjonsloggen i starten kan være unøyaktig.

3. Du får en viktig telefon, og blir nødt til å utsette tilsynsturen. Avbryt turen i appen.

På denne oppgaven trykket deltakeren på «Avslutt tilsynstur» i stedet for den røde «x»-knappen øverst i venstre hjørne.

4. Du er ferdig med telefonsamtalen, og starter en ny tilsynstur. Tittel og beskrivelse er valgfritt.

Deltakeren løste denne oppgaven uten problemer, men kommenterte at det hadde vært praktisk om det ikke var nødvendig å legge inn all informasjon på nytt hver gang en ny tilsynstur skal startes. Dette *er* mulig ved å legge inn informasjon på «Profil»-siden, som deretter vil fungere som foreslåtte verdier i feltene på «Ny tilsynstur»-siden. Utfordringen er at brukeren ikke havner på denne siden automatisk. Da det heller ikke er åpenbart hva denne siden gjør før man trykker på profilfanen, kan det ta lang tid før denne funksjonaliteten oppdages.

5. Du observerer en svart søye på toppen av stabburet. Søyen har rødt slips og en flenge i høyre side. Registrer observasjonen i appen.

Her brukte deltakeren flere sekunder før han forstod at observasjoner registreres ved å trykke på kartet. Før han fant ut dette forsøkte han for eksempel å trykke på den røde «x»-knappen, som avbrøt tilsynsturen. Det så ut som deltakeren litt tilfeldig trykket på kartet, og dermed skjønnte at det var sånn observasjoner ble registrert. Da han forstod dette var det tydelig for deltakeren at observasjonen ble registrert der han trykket på kartet. Registrering av informasjon gikk problemfritt, og deltakeren navigerte enkelt gjennom mulighetene her.

Brukeren trykket på et tidspunkt over kortet hvor informasjonen om observasjonen registreres ved et uhell, noe som resulterte i at kortet lukket

seg. Å trykke utenfor kortet fører til at kortet lukker seg, registreringsprosessen blir avbrutt, og all informasjon som er registrert går tapt. Det er litt for enkelt å gjøre dette ubevisst, noe som er en særdeles dårlig brukeropplevelse.

Deltakeren la merke til og kommenterte saueikonet som dukket opp på kartet når observasjonen ble lagret. Det var tydelig for deltakeren at dette representerte observasjonen som nettopp hadde blitt registrert og hvor den befant seg på kartet.

6. Du observerer nå en kongeørn som sirkler over nærmeste nabolomt. Registrer observasjonen i appen.

Deltakeren løste denne oppgaven uten problemer.

7. Du oppdager et sauekadaver midt på veien, rett ved innkjørselen din. Det er en brun søye, og det ser ut som den har blitt drept av en ulv. Den har et grønt slips.

Deltakeren løste denne oppgaven uten problemer.

8. Du finner nå en saueflokk i skråningen rett bak huset. Flokken består av både søyer og lam i forskjellige farger. Du ser to hvite og en svart søye. Du ser tre hvite lam, hvor ett av lammene halter på venstre bakben. Du ser også to svarte lam og ett brunt lam.

Deltakeren forstår raskt at flere sauer kan legges til i lista, og får dette til uten problemer. Han registrerer de tre hvite lammene samtidig, og skriver i beksrivelses-feltet at ett av dem er skadet i stedet for å registrere to som uskadet og deretter ett som skadet. Dette er relevant adferd å ta høyde for med tanke på hvordan applikasjonen bruker informasjonen som er registrert til å vise oppsummert informasjon om tilsynsturer og generere rapporter.

Deltakeren forsøkte flere ganger å scrolle ned til «Lagre»-knappen mens tastaturet var oppe. «Lagre»-knappen er skjult så lenge tastaturet vises, da den ikke registrerer trykk før tastaturet er fjernet, men det er tydelig at dette ikke nødvendigvis er en hensiktsmessig måte å håndtere denne utfordringen på.

9. Hvis du ikke ennå har gjort det, lagrer du nå observasjonen.

Da deltakeren trykket på «Lagre observasjon», gikk det noen sekunder uten at appen reagerte. Deltakeren trykket derfor på knappen på nytt. Dette resulterte i at det ble sendt to databasekall fra serveren, som gjorde at databasen havnet i en ugyldig tilstand med to like ID-er, siden brukeren utilsiktet lagret samme observasjon to ganger. Dette forårsaket flere feilmeldinger, og utvikleren som gjennomførte testen måtte gripe inn for å få databasen tilbake i en gyldig tilstand. Denne situasjonen oppsto på grunn av to svakheter som her kom tydelig fram.

Applikasjonen gir ikke alltid en umiddelbar tilbakemelding til brukeren etter at en knapp har blitt trykket på. Dette er tilfelle når en knapp forårsaker et serverkall, og applikasjonen venter på en respons fra serveren før den oppdaterer det som vises til brukeren. Dette gjør brukeren usikker

på om applikasjonen har fått med seg at knappen har blitt trykket på, og gjør også at applikasjonen virker lite responsiv.

Knapper som forårsaker et serverkall blir ikke deaktivert etter at de har blitt trykket på. Dette gjør at de kan trykkes på igjen før serverkallet er ferdig behandlet, og dette kan som observert i denne oppgaven føre til at applikasjonen havner i en ugyldig tilstand.

10. Til slutt oppdager du en brun ulldott flytende i midten av elva bak huset, og registrerer dette i appen.

På grunn av utfordringene i forrige oppgave ble tilsynsturen avsluttet, og denne oppgaven ble derfor hoppet over.

11. Du er nå ferdig med tilsynsturen, og lagrer denne.

Siden tilsynsturen allerede var avsluttet da vi kom til denne oppgaven, ble denne også hoppet over.

12. Se om du kan finne den lagrede tilsynsturen for å inspisere informasjonen du har registrert.

Deltakeren løste denne oppgaven uten problemer.

Da brukertesten var ferdig, fikk deltakeren også spørsmål om han mente det var noe som manglet i applikasjonen, eller om han hadde andre tilbakemeldinger. Han nevnte at det ikke var noen mulighet for å registrere gimmer (års gammel hunnsau) og andre type dyrerester en kadaver. Dette kan være relevant i enkelte situasjoner. Et eksempel er at hvis en drektig sau ved en feil blir sendt ut på beite før den har lammet, kan man komme over en morkake på en tilsynstur. Dette er det per nå ingen fornuftig mulighet for å registrere i applikasjonen. Ellers syntes deltakeren at løsingen fungerte bra, var enkel å bruke til å registrere informasjon, og dekket behovene for hva det er relevant å registrere på en tilsynstur.

12.6 Oppsummering

Brukertesting av applikasjonen er en viktig del av utviklingsprosessen, og å gjennomføre en strukturert brukertest på en bruker i målgruppen for applikasjonen ga verdifull innsikt i hvordan applikasjonen lykkes med det den forsøker å oppnå, men også hvordan den mislykkes. Det er flere deler av testprosessen som kunne vært forbedret, og dette diskuteres i neste kapittel. På tross av disse svakhetene har brukertesten som ble gjennomført i stor grad oppfylt hensikten med testingen.

Kapittel 13

Evaluering av brukertesting

Det er både styrker og svakheter med måten brukertesten i oppgaven ble gjennomført. Noen av svakhetene kunne vært unngått med bedre planlegging, mens enkelte svakheter er vanskelige å unngå på grunn av rammene til prosjektet.

13.1 Styrker

Brukertesten ble utført på en person i målgruppen til applikasjonen, og er dermed en meget relevant person å teste løsningen på. Kjennskapen til problemstillingen produktet forsøker å løse, samt erfaring med måten denne problemstillingen løses i dag, gjør at tilbakemeldingene fra testsubjektet er troverdige og pålitelige i forhold til evaluering av oppgavens suksesskriterier. Brukertesten ga også tydelige signaler om hvilke forbedringer som bør gjøres i applikasjonen, og dette var en viktig del av motivasjonen for å gjennomføre brukertesting i utgangspunktet.

Oppgavene som ble gitt til testsubjektet fungerte også godt. Det var lett for brukeren å forstå hva oppgaven betydde, mens brukerens utfordringer med å løse enkelte av oppgavene tyder på at de ikke signaliserte hvordan oppgaven skulle løses. Oppgavene dekket også all implementert kjernefunksjonalitet, og det var derfor klart etter testen hvilke applikasjonskrav som ikke var oppfylt, og hvilke deler av applikasjonen som trengte forbedringer.

13.2 Svakheter

Selv om brukertesten i hovedsak fungerte etter hensikten og ga relevante resultater, er det fortsatt betydelige svakheter med måten prosessen ble gjennomført på. For det første ble det bare gjennomført én testrunde. Det hadde vært hensiktsmessig å gjennomføre en testrunde tidligere i utviklingsprosessen, for å tidligere få tilbakemeldinger på hvilke deler av applikasjonen som bør revurderes eller endres. En konsekvens av dette kunne for eksempel vært at problemstillingen rundt treg respons på knapper som kaller serverfunksjoner hadde blitt problematisert tidligere. Ved å få denne tilbakemeldingen tidligere kunne for eksempel prinsippet om at applikasjonen fortløpende sender all informasjon til serveren blitt revurdert. Ved å lagre informasjon lokalt og bare

tidvis synkronisere denne med serveren kunne denne problemstillingen ha blitt eliminert, da lokal lagring går mye raskere enn å vente på en respons fra serveren.

Det er også en svakhet at applikasjonen ikke ble testet på flere deltakere. Selv om tilbakemeldingene om hvilke deler av applikasjonen som fungerte dårlig er ganske tydelige, hadde det vært nyttig å ha flere tilbakemeldinger på hvor godt delene av applikasjonen testsubjektet klarte å bruke uten problemer, fungerte. Mens observasjoner av brukerens erfaring med deler av applikasjonen som fungerer dårlig tydelig indikerer at disse bør endres, er ikke mangelen på kritikk av de andre delene av applikasjonen en garanti for at disse ikke inneholder problemer. Ved å teste applikasjonen bredere kunne dermed kravene som i dette prosjektet vurderes som tilfredsstillt blitt mindre usikre.

13.3 Resultater

Brukertesten ga tydelige signaler om hvilke deler av applikasjonen som må forbedres, og hvilke deler som tilsynelatende fungerer etter hensikten slik de nå er implementert. Disse beskrives i dette delkapittelet. Som Steve Krug argumenterer for i *Rocket Surgery Made Easy* [43] er det lurt å fokusere på bare de mest kritiske problemene som oppdages i brukertesten, da dette gir bedre fokus og gjør det mer sannsynlig at produktet raskest mulig forbedres mest mulig. Denne delen kommer derfor til å fokusere på de fem mest kritiske problemene når disse diskuteres.

13.3.1 Positive punkter

1. Opprettelse av bruker

Opprettelse av bruker og innlogging fungerer svært godt, og testen viste ingen klare svakheter med denne flyten. Testbrukeren oppdaget ikke muligheten til å registrere tilleggsinformasjon på profilsiden, men dette er ikke nødvendig for å bruke applikasjonen, og alle deler av applikasjonen fungerer selv om dette ikke gjøres. Innlogging og registrering av bruker fungerer veldig likt som hos de fleste andre applikasjoner som krever en brukerkonto for bruk av applikasjonen. Sannsynligheten for at det er betydelige skjulte svaketer som ikke ennå er oppdaget knyttet til denne delen av applikasjonen er derfor liten.

2. Registrering av observasjoner

Med unntak av at det kan være lite åpenbart at observasjonsflyten startes ved å trykke på kartet, fungerte registrering av observasjoner meget godt. Brukeren forstod raskt hvordan all beskrevet informasjon om hver observasjon skulle registreres, og klarte også å registrere flere sauer i én observasjon uten problemer. At noe informasjon ble notert i beskrivelsesfeltet i stedet for å bli registrert ved hjelp av feltene (se kap. 12.5, oppgave 8) kan by på utfordringer når det kommer til statistikk, men brukeren fikk uansett registrert all informasjonen beskrevet i alle oppgavene som ble gitt. Det ble også påpekt av brukeren at dette var et bevisst valg, da det var enklere å notere ned at ett lam var skadet i

beskrivelsesfeltet enn å registrere dette lammet separat. Hvis applikasjonen gjør det tydelig at informasjonen som registreres, brukes til å lage rapporter, vil brukeren trolig registrere så mye informasjon som mulig uten å bruke beskrivelsesfeltet for at rapportene skal bli korrekte.

3. Ikoner på kartet

Ikonene som vises på kartet for hver observasjon som er registrert fungerer også bra. Da disse dukker opp i det brukeren trykker på «Lagre observasjon»-knappen, er det enkelt å forstå at disse representerer observasjoner. De er også distinkte nok til at det er enkelt å skille dem fra hverandre, og plasseringen på kartet gjør det enkelt for brukeren å se om en observasjon allerede er registrert eller ikke.

4. Applikasjonsstruktur

Det ser også ut til at måten navigasjonen til applikasjonen er bygd opp på er intuitiv for brukeren. De fire fanene på bunnen kommuniserer klart hvilken funksjon de har, og det er enkelt å navigere mellom dem.

Testbrukeren hadde ingen problemer med å starte en ny tilsynstur, fullføre en pågående tilsynstur eller finne en oversikt over de fullførte tilsynsturene. Dette er også et design som brukes av mange andre applikasjoner med store brukermasser, så det er naturlig at brukere av applikasjonen utviklet i dette prosjektet allerede er kjent med denne måten å navigere gjennom en applikasjon på.

13.3.2 Punkter som må forbedres

Punktene i denne lista er sortert etter hvor kritisk det er at de forbedres, hvor det første punktet er mest kritisk. Dette gjør det tydelig hvor det er lurt å starte ved en neste iterasjon i utviklingen av applikasjonen. Hvor kritisk et problem er avhenger av to faktorer: Hvor sannsynlig det er at en bruker vil støte på problemet, og hvor store konsekvenser det får om problemet støtes på [43].

1. Det er mulig å forårsake ulovlige databasekall i applikasjonen

Dette problemet ble oppdaget da testbrukeren forsøkte å trykke på «Lagre observasjon»-knappen (figur 11.3.5c) på nytt etter å ha ventet i noen sekunder uten at applikasjonen ga noen tilbakemelding etter det første trykket. Dette er det største problemet med applikasjonen i dens nåværende tilstand, da brukeren enkelt kan forårsake en ugyldig databasetilstand og at applikasjonen slutter å respondere. Sannsynligheten er stor for at dette skjer ved normal bruk av applikasjonen, og konsekvensen av at det skjer er også stor.

2. Brukeren får ingen bekreftelsesdialog for irreversible valg

Når brukeren trykker på «Avbryt»-knappen eller «Avslutt tilsynstur»-knappen under en observasjon (figur 11.3.9), utføres resultatet av denne knappen umiddelbart. Dette er uheldig, da det ikke er mulig å gjøre om denne handlingen hvis en bruker ved et uhell for eksempel trykker på feil knapp. Hvis en sauebonde på slutten av en tilsynstur trykker på «Avbryt»-knappen i stedet for «Avslutt tilsynstur»-knappen, går all den

registrerte informasjonen tapt. Dette kan føre til at en tilsynstur på flere timer ikke fører til noen form for dokumentasjon. Sannsynligheten for at dette skjer er bare middels stor, men konsekvensene kan være meget store.

3. Å trykke på skjermen over kortet når en observasjon registreres avbryter registreringen av observasjonen

Se figur 11.3.5c for et eksempel på en tilstand hvor dette problemet er tilstede. Hvis brukeren trykker over kortet på denne skjermen, lukker kortet seg, og registreringen av observasjonen avbrytes. Dette fører til at all informasjonen brukeren har registrert går tapt, og dette er en meget frustrerende brukeropplevelse. Det er heller ikke sikkert det er åpenbart for brukeren hva som gjorde at kortet lukket seg, da det er relativt enkelt å komme borti skjermen over kortet. Sannsynligheten for at dette skjer er altså stor, og konsekvensen er middels stor, da mye informasjon kan gå tapt.

4. Mangel på tilbakemelding når knapper trykkes på

Når en knapp som forårsaker et serverkall blir trykket på, får ikke brukeren noen tilbakemelding fra applikasjonen på at noe har skjedd før applikasjonen mottar en respons fra serveren. Dette gjør at applikasjonen føles lite responsiv, og kan gjøre brukeren usikker på om applikasjonen fikk med seg at knappen ble trykket på. Dette problemet oppstår hver gang en bruker trykker på for eksempel «Start»-knappen (figur 11.3.2), og det er altså et problem samtlige brukere vil bli utsatt for. Konsekvensen er derimot ganske liten.

5. Standardverdier for «Ny tilsynstur»-skjerm

Det er ikke åpenbart hvilken hensikt profilsiden har, eller at feltene i «Ny tilsynstur»-skjermen (figur 11.3.2) automatisk fylles inn med verdiene fra denne siden. Dette er både fordi brukeren aldri blir sendt til profilsiden og fordi side 11.3.2 ikke forklarer hverken at feltene kan være forhåndsutfyllt eller hvor verdiene kommer fra. Det er ganske sannsynlig at en bruker ikke oppdager denne funksjonaliteten, og dermed fyller inn alle feltene hver gang en ny tilsynstur skal startes. Konsekvensen av dette er ikke veldig stor, men det kan føre til noe frustrasjon hos brukerne.

13.4 Oppsummering

Testing av applikasjonen er viktig både for å kartlegge hvilke krav som er oppnådd, og for å kartlegge hvor fokuset bør være på videreutvikling av løsningen. Testingen i dette prosjektet var en suksess når det gjelder kartlegging av fokuset for videreutvikling, da den ga klare indikasjoner på hva som er applikasjonens største svakheter. Denne hensikten er fokuset til Steve Krug [43] og den største motivasjonen for testing som en del av en vanlig produktutviklingsprosess. I et prosjekt som dette er det derimot viktig å også kartlegge i hvilken grad løsningen oppfyller kravene og suksesskriteriene som ble presentert ved starten av prosjektet. Testingen i dette prosjektet gir gode indikasjoner også på dette punktet, men det begrensede antallet tester og

testdeltakere gjør at det er tilknyttet usikkerhet rundt validiteten til disse resultatene.

Kapittel 14

Evaluering av løsningen

Den endelige prototypen varierer i hvilken grad den oppfyller kravene beskrevet i kapittel 7. Det er flere krav som er oppnådd, en god del krav som er delvis oppnådd, og noen krav som ikke er oppnådd i det hele tatt. Arbeidet med prototypen indikerer at det trolig er mulig å lage en løsning som i stor grad imøtekommer behovene til bøndene, samtidig som den legger til rette for registrering av tilstrekkelig informasjon til å generere tilfredsstillende rapporter til offentlige tilsynsmyndigheter. Samtidig er det tydelig at en slik løsning krever flere iterasjoner med forbedringer før den kan tas i bruk i praksis, og den endelige prototypen i dette prosjektet er et godt stykke unna et produkt som er klart til å tas i bruk kommersielt. Nedenfor gjennomgås kravspesifikasjonen til løsningen og hvorvidt disse kravene er tilfredsstilt eller ikke. Deretter vurderes løsningen opp mot suksesskriteriene beskrevet i kapittel 3.

14.1 Funksjonelle krav

Id	Beskrivelse	Status	Kommentar
F1	Brukeren skal kunne starte og fullføre tilsynsturer	Delvis fullført	F1.1, F1.2, F1.3 fullført
F1.4	Brukeren skal advares med en bekreftelsesdialog når vedkommende prøver å avbryte/avslutte turen	Ikke fullført	Nedprioritert på grunn av tid og prioritet
F2	Brukeren skal se et kart over Norge i løpet av tilsynsturen	Fullført	
F3	Brukeren skal kunne registrere observasjoner med tilstrekkelig informasjon i løpet av tilsynsturene	Delvis fullført	F3.1, F3.2, F3.3, F3.4, F3.5, F3.6 fullført
F3.7	Alle observasjonene skal ha støtte for å legge inn bilder	Ikke fullført	Nedprioritert på grunn av tid og prioritet

Id	Beskrivelse	Status	Kommentar
F4	Brukeren skal kunne se og redigere registrerte observasjoner i løpet av tilsynsturen	Ikke fullført	Nedprioritert på grunn av kompleksitet og tid
F5	Appen skal kontinuerlig spore posisjonen til brukeren i løpet av hver tilsynstyr	Delvis fullført	Implementert når appen er åpen, men ikke aktiv i bakgrunnen
F6	Brukeren skal kunne se en liste over gjennomførte tilsynsturer i appen	Delvis fullført	Gjennomførte turer vises i en liste
F6.1	Når brukeren trykker på en av turene i lista, skal appen vise et kart som inneholder observasjonene og ruten fra turen	Delvis fullført	Turer kan trykkes på, men de viser ikke all relevant info om tilsynsturen
F6.1.1	Når brukeren trykker på en av observasjonene på kartet, skal appen vise den registrerte informasjonen fra observasjonen	Ikke fullført	Nedprioritert på grunn av tid og prioritet
F7	Appen skal gi brukere mulighet til å generere en rapport som inneholder informasjon om de gjennomførte tilsynsturene	Ikke fullført	Nedprioritert på grunn av kompleksitet og tid
F8	Appen skal lagre gjennomførte turer på nett	Fullført	
F9	Appen skal kunne brukes uten internett	Ikke fullført	Nedprioritert på grunn av kompleksitet og tid
F10	Appen skal kreve innlogging for å kunne brukes	Fullført	

14.1.1 Kommentarer til funksjonelle krav

Mye av funksjonaliteten til applikasjonen er implementert, og det meste av denne fungerer også meget bra. For enkelte krav er det delmål med lav prioritet som ikke har blitt gjennomført (for eksempel F1.4 og F3.7), mens resten av delkravene er oppnådd og fungerer som de skal. Det er samtidig noen store krav som ikke er implementert, da dette ikke var mulig å få til innenfor tidsrammene til prosjektet (for eksempel F4, F7 og F9). Når ulike krav måtte settes opp mot hverandre på grunn av begrenset tid, ble kravene med høyest prioritet i stor grad gjennomført først. På enkelte tidspunkt ble derimot krav med høy prioritet (for eksempel F7) nedprioritert, da disse kravene representerte funksjonalitet som i liten grad påvirket andre krav. Hvis funksjonaliteten for å generere en rapport ikke er implementert, påvirker ikke dette hvordan resten av applikasjonen fungerer. På bakgrunn av dette ble funksjonalitet med lavere prioritet, men tett sammenheng med annen funksjonalitet, noen ganger utviklet før funksjonalitet med høyere prioritet.

14.2 Ikke-funksjonelle krav

Id	Beskrivelse	Status	Kommentar
IF1	Appen skal være tilgjengelig både på Android og iOS	Ikke fullført	Nedprioritert på grunn av kompleksitet og tid
IF2	Appen skal kunne brukes uten å lese instruksjoner eller dokumentasjon på forhånd	Delvis fullført	Enkelte deler av applikasjonen kan være vanskelige å forstå uten instruksjoner
IF3	Det skal ikke være mulig å forårsake server-feil ved å bruke applikasjonen	Ikke fullført	Oppdaget etter siste runde med utvikling
IF4	Appen skal bevare alle observasjoner og det meste av posisjonsloggen om den krasjer	Fullført	
IF5	Appen skal skalere dynamisk basert på antall brukere og mengden lagret informasjon	Fullført	

14.2.1 Kommentar til ikke-funksjonelle krav

Også når det kommer til ikke-funksjonelle krav ble enkelte krav nedprioritert på grunn av tidsrammen til prosjektet. Selv om applikasjonen er laget i et rammeverk som tillater at applikasjonen kjører på både iOS og Android [32], krever testing på iOS at man har tilgang til enten en iPhone eller en maskin som har operativsystemet macOS [33, 44]. Mye av koden fungerer automatisk både på Android og iOS, men fordi koden som skrives i React Native samhandler med APIene til Android og iOS, kan det være forskjeller i hvordan programmet

fungerer i praksis [32]. Det er derfor lite sannsynlig at løsningen fungerer som den skal på iOS uten at det gjøres noen tilpasninger. Siden utvikleren av applikasjonen ikke har tilgang til utstyr som tillater testing av applikasjonen på iOS, ble dette funksjonelle kravet ikke prioritert. Kravet er allikevel med i spesifikasjonen da over halvparten av smarttelefonene i Norge er iPhoner [45]. Med andre ord, enhver relevant applikasjon som skal ha en viss appell til brukergruppen må kunne brukes på iOS. Ved å utvikle applikasjonen i et rammeverk som støtter både iOS og Android, vil det være enklere å tilpasse applikasjonen så den også fungerer på iOS på et senere tidspunkt enn hvis den for eksempel hadde blitt skrevet i Kotlin, det primære programmeringsspråket som brukes på Android [46].

Når det kommer til de andre funksjonelle kravene som ikke ble tilfredsstilt i den endelige prototypen, handler det i mindre grad om nedprioritering, og i større grad om at applikasjonens mangler på disse områdene ikke ble tydelige før brukertesten var gjennomført. Applikasjonen ble gjennom hele prosjektperioden designet så den skulle kunne forstås uten instruksjoner, og så det skulle være enkelt å ta den i bruk. I hvilken grad dette er oppnådd er vanskelig å vite sikkert før produktet har blitt testet, så den eneste realistiske måten å oppfylle dette kravet fullstendig hadde vært å gjennomføre flere iterasjoner med testing og utvikling. Kravet om at applikasjonen ikke skal kunne forårsake serverfeil ble også forsøkt oppnådd i løpet av utviklingen. Også applikasjonens svakheter på dette området ble oppdaget i løpet av brukertesting, men i motsetning til applikasjonens brukervennlighet kunne dette problemet blitt oppdaget på andre måter. Brukerens interaksjon med applikasjonen utløste problemer som eksisterer på grunn av svakheter i implementeringen, og disse svakhetene hadde blitt tydelige ved å gjennomgå rutine for serverkall og hvordan disse håndteres. Av de ikke-funksjonelle kravene som ikke ble oppfylt var det altså enkelte krav som ble bevisst nedprioritert, mens andre krav ble forsøkt oppnådd uten suksess.

Kapittel 15

Evaluering av prosjektet

Kravene beskrevet i forrige kapittel ble utviklet basert på suksesskriteriene for applikasjonsprototypen. I dette kapittelet diskuteres hvorvidt disse suksesskriteriene er oppfylt, og problemstillingen besvares.

15.1 Evaluering opp mot suksesskriterier

S1: Produktet skal være brukervennlig, intuitivt og enkelt å bli kjent med for brukerne

Dette suksesskriteriet er til en viss grad oppnådd. Brukertesten som ble gjennomført etter utviklingen av prototype 2 viser at store deler av applikasjonen er intuitiv og enkel å bli kjent med. Samtidig er det enkelte elementer som gjør at produktet ikke er like brukervennlig som det burde være. Applikasjonen burde ikke gi brukeren mulighet til å forårsake problemer i databasen eller applikasjonens tilstand, flere handlinger som har store konsekvenser for brukeren bør forårsake en bekreftelsesdialog før konsekvensen inntreffer, og funksjonalitet med dårlig antydning bør forbedres (blant annet å registrere en observasjon ved å trykke på kartet og hensikten til profilsiden). Allikevel indikerer brukertesten at konseptene som designet til prototype 2 bygger på er gode og logiske. Det skal altså lite til å sørge for at applikasjonene i enda større grad oppfyller dette suksesskriteriet ved videreutvikling av løsningen.

S2: Løsningen skal møte bøndenes behov for å registrere informasjon i forbindelse med tilsynsturer

Dette suksesskriteriet er i stor grad oppnådd. Løsningen som er utviklet gjør brukeren som utfører en tilsynstur i stand til å registrere nesten all informasjon som per nå blir registrert uten tekniske hjelpemidler. I tillegg til mulighet til å registrere informasjonen som allerede registreres i dag, sporer også applikasjonen posisjonen til brukeren, selv om denne funksjonaliteten har noen mangler. I løpet av brukertesten ble det nevnt at det kunne være relevant å ha mulighet til å registrere dyrerester som en observasjon, og det kan også tenkes at det dukker opp andre typer observasjoner som ikke passer inn i en av de eksisterende kategoriene, men som allikevel kan være interessante å registrere. En kategori for «annet» bør derfor vurderes ved videre utvikling. Applikasjonen mangler også

mulighet for å legge ved bilder til observasjonene, og denne funksjonaliteten ville ført til at applikasjonen i enda større grad hadde oppfylt dette suksesskriteriet.

S3: Informasjonen registrert i produktet skal i tilstrekkelig grad dekke myndighetenes krav til dokumentasjon

Dette suksesskriteriet er oppfylt. Applikasjonen gjør brukerne i stand til å registrere all informasjon som er anbefalt å samle inn per dags dato (vedlegg A.1), i tillegg til ekstra informasjon om brukerens posisjon i løpet av turen, ved registrering av hver observasjon og observasjonens lokasjon. Det er med andre ord ingen grunn til å tro at informasjonen som samles inn ved å bruke applikasjonen ikke er tilstrekkelig til å dekke myndighetenes dokumentasjonskrav.

S4: Løsningen skal lages med moderne teknologi og være i stand til å videreutvikles og vedlikeholdes etter at prosjektet er ferdig

Dette suksesskriteriet er også oppfylt. Applikasjonen er bygget på moderne teknologi, og deler av applikasjonen har tatt i bruk funksjonalitet og teknologi som har blitt lansert i løpet av prosjektet. Teknologien som er brukt har også blitt oppdatert fortløpende i løpet av prosjektperioden. Applikasjonen i sin nåværende form kan enkelt videreutvikles og vedlikeholdes, og løsningen er bygget på en måte som skal gjøre det enkelt å lansere produktet kommersielt ved en senere anledning.

15.2 Evaluering opp mot problemstilling

Problemstillingen som ble presentert i starten av oppgaven var:

I hvilken grad er det mulig å utvikle en mobilapplikasjon som enkelt og effektivt lar bønder registrere detaljert lokasjonsbasert informasjon om sauer på utmarksbeite, som oppfyller myndighetenes dokumentasjonskrav og kan videreutvikles til en løsning som kan tas i bruk i praksis?

Som vist gjennom evalueringen av produktet og prosjektet, er det i aller største grad mulig å utvikle en mobilapplikasjon som møter disse målene. Prototypen som er utviklet er på mange måter allerede både enkel og effektiv å bruke, som bekreftet av brukertesten. Brukertesten bidro også til å kartlegge applikasjonens svakheter, og disse har klare og tydelige løsninger. Ved å implementere disse løsningene kan applikasjonen bli enda enklere å bruke, og det er tydelig at dette er oppnåelig. Den endelige prototypen lar bønder registrere både detaljert og lokasjonsbasert informasjon om sauer på utmarksbeite, samt andre interessante observasjoner som rovdyr og ullfunn. Det er mulig å gjøre denne informasjonen enda mer detaljert ved å legge til funksjonalitet for en «annet»-kategori og mulighet til å legge ved bilder i de registrerte observasjonene. Prototypen er laget slik at den kan utvides med denne funksjonaliteten. Informasjonen som er registrert ser ut til å tilfredsstille myndighetenes dokumentasjonskrav til

tilsynsturer, så prosjektet har vist at også dette er mulig. Løsningen er ikke tatt i bruk i praksis, men den er bygget på teknologi som er laget for at det skal være enkelt å publisere applikasjonen, og slik at dynamisk skalering skjer automatisk. Det er altså ikke bare mulig, men realistisk å kunne utvikle en slik applikasjon.

Kapittel 16

Konklusjon

I dette prosjektet ble det forsøkt utviklet en prototype til en applikasjon som kan bistå bønder med tilsyn av sauer på utmarksbeite, for å utforske om det er mulig å lage en løsning som er både effektiv, brukervennlig, og tilfredsstillende både bøndenes og tilsynsmyndighetenes behov for dokumentasjon. Prosjektet startet med en undersøkelse av hvordan tilsyn utføres og dokumenteres på det nåværende tidspunktet. Denne undersøkelsen viste at det mangler gode verktøy som kan bistå bønder med tilsynsprosessen. Det ble deretter utviklet en kravspesifikasjon og en designprototype for å vise hvordan en mulig applikasjon kunne se ut. Basert på kravspesifikasjonen, designprototypen og tilbakemeldinger fra veileder, ble det deretter utviklet en funksjonell prototype. Denne ble testet på en tidligere sauebonde, og resultatene fra denne testen la grunnlaget for en evaluering av prototypen. Prototypen ble evaluert ved å gjennomgå kravspesifikasjonen til løsningen og kartlegge i hvilken grad denne var oppnådd. I løpet av evalueringsprosessen ble også de største svakhetene med applikasjonen fremhevet.

Evalueringen av den endelige prototypen viser at applikasjonsprototypen på nåværende tidspunkt har flere svakheter som må forbedres for at applikasjonen skal tilfredsstillende de spesifiserte behovene. Samtidig er prototypen et godt utgangspunkt for videre utvikling av en applikasjon som kan tas i bruk i praksis, og resultatene indikerer at alle de spesifiserte kravene er oppnåelige. Prosjektet har altså vist at det i stor grad både er teoretisk og praktisk mulig å utvikle en løsning som tilfredsstillende både bøndenes og tilsynsmyndighetenes behov.

Kapittel 17

Videre arbeid

Løsningen som er utviklet i løpet av dette prosjektet har mange muligheter for videre utvikling. Prototypen har flere områder hvor den ikke oppfyller suksesskriteriene som er spesifisert i prosjektet, og det er naturlig å jobbe videre med applikasjonen til den tilfredsstillende disse kravene. Det er samtidig flere muligheter for implementering av ny funksjonalitet som ikke er foreslått i løpet av prosjektet. I tillegg til utvikling er det naturlig å også brukerteste både funksjonalitet som blir forbedret og ny funksjonalitet, for å finne ut om implementeringen fungerer etter hensikten. Dette kapitlet beskriver både nødvendige forbedringer og muligheter for ny funksjonalitet.

17.1 Oppfyllelse av eksisterende krav

For at applikasjonen skal kunne tas i bruk i praksis, er det flere av kravene som ikke ble tilfredsstillende av den endelige prototypen som må oppnås. Dette gjelder både funksjonalitet som ikke fungerte helt som den skulle, men også funksjonalitet som ble nedprioritert, og dermed ikke forsøkt implementert.

17.1.1 Forbedringer av eksisterende funksjonalitet

Som beskrevet i seksjon 13.3.2, er det flere betydelige problemer som bør løses for at applikasjonen skal være brukervennlig med funksjonaliteten som allerede er implementert. Et naturlig første steg i videreutviklingen av applikasjonen er dermed å løse disse problemene. Dette innebærer å sørge for at brukerens handlinger ikke kan forårsake ulovlige databasekall, at det er vanskeligere å avbryte registreringen av nye observasjoner, implementering av bekreftelsesdialog der det er relevant og lignende. Det er også naturlig å vurdere om applikasjonen primært skal lagre data lokalt på enheten, for så å synkronisere dette med en database på nett med jevne mellomrom. Dette vil gjøre applikasjonen mer responsiv ved at mindre tid går med til å vente på en respons fra serveren, og det vil også gjøre det enklere å få applikasjonen til å fungere uten internett på et senere tidspunkt.

17.1.2 Implementering av manglende funksjonalitet

Kapittel 14 beskriver hvordan enkelte krav ble nedprioritert under utviklingen av den endelige prototypen på grunn av prosjektets omfang. Dette er funksjonalitet som er inkludert i kravspesifikasjonen, men som ikke ble forsøkt utviklet. Etter at applikasjonens eksisterende funksjonalitet har gjennomgått den nødvendige videreutviklingen er det naturlig å jobbe videre med disse kravene. Dette innebærer å få applikasjonen til å fungere på iOS i tillegg til Android, å gi brukeren mulighet til å bruke applikasjonen uten internett, og funksjonalitet for å generere en rapport som inneholder informasjon om de gjennomførte tilsynsturene.

17.2 Ny funksjonalitet

I tillegg til funksjonaliteten beskrevet i kravspesifikasjonen til dette prosjektet, er det mange muligheter for annen funksjonalitet som kunne gjort løsningen enda bedre. Dette innebærer blant annet støtte for at flere brukere har tilgang til samme datasett.

Det er ofte flere enn én person som utfører tilsyn for et sett med sauer, enten det er fordi et beitelag samarbeider om å følge opp flere bønders sauer eller fordi bonden får hjelp fra familie, venner eller innleid arbeidskraft til dette. For å gjøre samarbeidet mest mulig sømløst kan det være nyttig at flere brukere kan registrere tilsynsturer for samme sett med sauer, og at alle disse brukerne kan se alle de gjennomførte tilsynsturene.

Det er også, som beskrevet i kapittel 5, mange sauer som er utstyrt med en form for sporingsutstyr som gir bonden en oversikt over hvor sauene befinner seg. Når en bonde skal utføre en tilsynstur kunne det vært nyttig å se disse posisjonene i applikasjonen, slik at det blir enklere å lokalisere sauene. I hvilken grad en slik integrasjon er mulig er utenfor omfanget til denne oppgaven, men om det er gjennomførbart hadde det trolig vært nyttig for brukerne av applikasjonen med en slik funksjonalitet.

Kildeliste

- [1] Statistisk sentralbyrå, *Beitebruk og seterdrift*, 2021. Hentet fra: <https://www.ssb.no/jord-skog-jakt-og-fiskeri/artikler-og-publikasjoner/beitebruk-og-seterdrift>.
- [2] Statistisk sentralbyrå, *Gardsbruk, jordbruksareal og husdyr*, 2023. Hentet fra: <https://www.ssb.no/jord-skog-jakt-og-fiskeri/jordbruk/statistikk/gardsbruk-jordbruksareal-og-husdyr>.
- [3] Landbruksdirektoratet. «Utmarksbeiting og rovvilt.» [landbruksdirektoratet.no](https://www.landbruksdirektoratet.no). Hentet fra: <https://www.landbruksdirektoratet.no/nb/jordbruk/miljo-og-klima/utmarksbeiting-og-rovvilt>.
- [4] *Forskrift om velferd for småfe*, 2005. [Online]. Hentet fra: <https://lovdata.no/dokument/SF/forskrift/2005-02-18-160>.
- [5] *Forskrift om erstatning når husdyr blir drept eller skadet av rovvilt*, 2014. [Online]. Hentet fra: <https://lovdata.no/dokument/SF/forskrift/2014-05-30-677>.
- [6] S. Dysthe, A. Kjerstad, «Effektiviserin av manuell oppfølging av sau på utmarksbeite», Masteroppgave i informatikk, Institutt for datateknologi og informatikk, Norges teknisk-naturvitenskapelige universitet, Trondheim, 2018.
- [7] *Lov om dyrevelferd*, 2009. [Online]. Hentet fra: <https://lovdata.no/dokument/NL/lov/2009-06-19-97>.
- [8] *Lov om matproduksjon og mattrygghet mv. (matloven)*, 2004. [Online]. Hentet fra: <https://lovdata.no/dokument/NL/lov/2003-12-19-124>.
- [9] Statsforvalteren i Trøndelag, «Regionale miljøtilskudd til jordbruket i Trøndelag 2022,» Statsforvalteren i Trøndelag, Steinkjer, Norge, 04.07.2022. Hentet fra: statsforvalteren.no/contentassets/7f7b2c780a794e009800b3c6c9e64585/veileder-regionale-miljotilskudd-trondelag-2022-versj-4.-juli.pdf.
- [10] Statsforvalteren i Møre og Romsdal. «Søknad om erstatning for rovviltskade i 2021.» [statsforvalteren.no](https://www.statsforvalteren.no). Hentet fra: <https://www.statsforvalteren.no/more-og-romsdal/miljo-og-klima/rovvilt/soknad-om-erstatning-for-rovviltskade-i-2021/>.
- [11] T. K. Henriksen, «Konkurransen i beite-tech hardner til: «Alle» vil passe på dyrene dine.» [Shifter.no](https://www.shifter.no). Hentet fra: <https://www.shifter.no/nyheter/konkurransen-i-beite-tech-hardner-til-alle-vil-passe-pa-dyrene-dine/165196>

- (Lastet ned: 26.05.2023).
- [12] K. Bergo, «Sporingssystemer: Her er tilbyderne.» Norsklandbruk.no. Hentet fra: <https://www.norsklandbruk.no/sporingssystemer-her-er-tilbydere/s/5-152-19573> (Lastet ned: 26.05.2023).
- [13] Telespor. «Produkt.» telespor.no. Hentet fra: <https://telespor.no/produkt/> (Lastet ned: 28.02.2023).
- [14] Findmy, «Findmy.» Findmy.no. Hentet fra: <https://findmy.no/no/agtech?type=sheep> (Lastet ned: 29.05.2023).
- [15] Findmy, «E-bjelle Model 2 BLÅ, GRØNN og RØD.» Findmy.no. Hentet fra: https://findmy.no/no/agtech/news/E-bjelle_Model_2_BLA,_GRONN_og_ROD?type=sheep (Lastet ned: 29.05.2023).
- [16] Spordata, «Gjetargut.» Gjetargut.no. Hentet fra: <https://gjetargut.no/> (Lastet ned: 30.05.2023).
- [17] Ture Apps. «Norgeskartportalen.» norgeskart.avinet.no. Hentet fra: <https://norgeskart.avinet.no/> (Lastet ned: 28.02.2023).
- [18] *GPS Exchange Format*, GPX 1.1, 2004. [Online]. Hentet fra: <https://www.topografix.com/gpx.asp>.
- [19] Fant AS. «En revolusjonerende beiteapp!» beitesnap.no. Hentet fra: <https://www.beitesnap.no/> (Lastet ned: 06.03.2023).
- [20] Fant AS. «BeiteSnap blir GRATIS for 2019!» beitesnap.no. Hentet fra: <https://www.beitesnap.no/gratis-2019/> (Lastet ned: 06.03.2023).
- [21] J. B. Sande. «Nyttig app til beitesesongen.» Bondevennen.no. Hentet fra: <https://www.bondevennen.no/aktuelt/nyttig-app-til-beitesesongen/> (Lastet ned: 06.03.2023).
- [22] Abaris Consulting AS. «BeiteSnap.» Apple.com. Hentet fra: <https://apps.apple.com/no/app/beitesnap/id1233671868> (Lastet ned: 06.03.2023).
- [23] *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*, ISO/IEC 25010, 2011. [Online]. Hentet fra: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010>.
- [24] *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*, ISO 9241-210, 2019. [Online]. Hentet fra: <https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1096929>.
- [25] S. Richard, P. LePage. «What are Progressive Web Apps?» Web.dev. Hentet fra: <https://web.dev/what-are-pwas> (Lastet ned 06.05.2023).
- [26] Figma, Inc. «It's time to design on your terms» Figma.com. Hentet fra: <https://www.figma.com/design> (Lastet ned: 06.05.2023).
- [27] Figma, Inc. «Prototype while you design, and vice versa.» Figma.com. Hentet fra: <https://www.figma.com/prototyping> (Lastet ned: 06.05.2023).

- [28] W. W. Royce, «Managing the development of large software systems,» presentert ved Proceedings of IEEE WESCON, Los Angeles, CA, USA, August 1970.
- [29] K. Beck et al. «Manifesto for Agile Software Development.» agilemanifesto.org. Hentet fra: <https://agilemanifesto.org/> (Lastet ned: 25.04.2023).
- [30] Github, Inc. «The tools you need to build what you want.» Github.com. Hentet fra: <https://github.com/features> (Lastet ned 05.05.2023).
- [31] A. Manchanda. «What is Software Architecture in Software Engineering?» NetSolutions.com. Hentet fra: <https://www.netsolutions.com/insights/why-software-architecture-matters-to-build-scalable-solutions/> (Lastet ned: 27.04.2023).
- [32] Meta Platforms, Inc. «React Native.» Reactnative.dev. Hentet fra: <https://reactnative.dev/> (Lastet ned: 03.05.2023).
- [33] Expo. «FAQ.» Expo.dev. Hentet fra: <https://docs.expo.dev/faq/> (Lastet ned: 03.05.2023).
- [34] Expo. «Instant updates.» Expo.dev. Hentet fra: <https://docs.expo.dev/deploy/instant-updates/> (Lastet ned: 03.05.2023).
- [35] tRPC.io. «Introduction», tRPC.io. Hentet fra: <https://trpc.io/docs> (Lastet ned: 03.05.2023).
- [36] Vercel, Inc. «What is Next.js?» Nextjs.org. Hentet fra: <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs> (Lastet ned: 03.05.2023).
- [37] Vercel Inc. «Serverless Functions Overview.» Vercel.com. Hentet fra: <https://vercel.com/docs/concepts/functions/serverless-functions> (Lastet ned: 27.04.2023).
- [38] Prisma Data, Inc. «What is Prisma?» Prisma.io. Hentet fra: <https://www.prisma.io/docs/concepts/overview/what-is-prisma> (Lastet ned: 03.05.2023).
- [39] PlanetScale, Inc. «What is PlanetScale?» PlanetScale.com. Hentet fra: <https://planetScale.com/features> (Lastet ned: 03.05.2023).
- [40] Clerk Inc. «Simple and Powerful Auth Built for React Native.» Clerk.com. Hentet fra: <https://clerk.com/solutions/react-native-expo> (Lastet ned: 04.05.2023).
- [41] D. A. Norman, *The Design of Everyday Things: Revised and Expanded Edition*. New York, NY, USA: Basic Books, 2013.
- [42] Oppland Sau og Geit, «Slips og fargekoder.» NSG.no. Hentet fra: <https://www.nsg.no/oppland/nyheter/slips-og-fargekoder> (Lastet ned: 15.05.2023).
- [43] S. Krug, *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. Berkeley, CA, USA: New Riders, 2010.
- [44] Expo. «iOS Simulator.» Expo.dev. Hentet fra: <https://docs.expo.dev/workflow/ios-simulator/> (Lastet ned: 01.06.2023).

- [45] StatCounter. «Mobile Operating System Market Share Norway.» Statcounter.com. Hentet fra: <https://gs.statcounter.com/os-market-share/mobile/norway> (Lastet ned: 01.06.2023).
- [46] Kotlin Foundation, «Kotlin for Android.» Kotlinlang.org. Hentet fra: <http://kotlinlang.org/docs/android-overview.html> (Lastet ned: 01.06.2023).

Vedlegg

Tillegg A

Rapportskjemaer

A.1 Rapportkjema for beitesesongen 2021 Vestland

Skjema for beitesesongen 2021

Tabellen **Oppsummering av beitesesongen** skal etter beitesesongen fyllast ut og sendast til leiaren i beitelaget. Tapstal som skal rapporterast er berre tap på utmarksbeite. Tap inne og på vår- og haustbeite på innmark skal ikkje takast med i denne statistikken. For å få best mogleg oversyn over tapsårsaker er det viktig at kvar bonde i si rapportering til beitelaget om hausten melder inn ei så fullstendig oppstilling av sikre og trulege tapsårsaker for sin buskap som mogeleg. Fylkesmannen ynskjer ei tilbakemelding om tapsårsaker når leiar søker om driftsmidlar (i kommentarfeltet eller som vedlegg).

Tilsynslogg kan nyttast for å sikre dokumentasjon på utført tilsyn med dyra i utmarka gjennom beitesesongen. Kvar tur skal noterast fortløpande i tabellen. Aktuelle opplysningar kan t.d. vera: daude og skadde dyr, tapsårsaker, observasjon av rovvilt, sauer som manglar lam, generell uro i beitet, beiteforhold med meir.

RAPPORTSKJEMA FOR TILSYN AV DYR PÅ UTMARKSBEITE 2021

Namn: _____

Organisasjonsnummer: _____

Oppsummering av beitesesongen

Dyr sleppt/ sanka på <i>utmarksbeite</i>	Sau		Lam		Geit		Storfe	
	Sleppt	Sanka	Sleppt	Sanka	Sleppt	Sanka	Sleppt	Sanka

Tilsyn i dagsverk per veke (gjennomsnittleg):	
--	--

Sikre/dokumenterte tapsårsaker :								
Flått	Alveld	Rev	Konge- ørn	Freda rovdyr	Ulukker	Svake dyr	Flugemakk	Anna
Trulege tapsårsaker :								
Flått	Alveld	Rev	Konge- ørn	Freda rovdyr	Ulukker	Svake dyr	Flugemakk	Anna
Kommentarar:								

Tilsynslogg for:

Beiteår:

Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	
Dato:	Rute/område:
Observasjonar:	

A.2 Rapport generert av Beitesnap



Beiterapport

For produsent

Produsentnummer: [REDACTED]

Navn: [REDACTED]

Adresse: [REDACTED]

Periode

Fra: 01.06.2018 Til: 30.10.2018

Tilsyn

Bruker	Startet	Avsluttet	Utført av	Notat	Timeforbruk
Vanlig tilsyn					
[REDACTED]	10.06.2018 16:26	10.06.2018 20:44	Tilknyttede		4 timer 18 minutter
[REDACTED]	14.06.2018 17:57	14.06.2018 21:09	Tilknyttede		3 timer 11 minutter
[REDACTED]	16.06.2018 17:55	16.06.2018 19:50	Tilknyttede	9geit og 16 killinger. Bytta 3 batt	1 time 54 minutter
[REDACTED]	17.06.2018 12:00	17.06.2018 14:00	Tilknyttede	Slapp kopplamma på setra. Ordna med vannet. Såg alle i skogskvea.	2 timer 0 minutter
[REDACTED]	27.06.2018 14:38	27.06.2018 17:43	Tilknyttede	Til Nedre Tverrglitra m saltstein. Rolig, tØrt	3 timer 4 minutter
[REDACTED]	27.06.2018 18:43	27.06.2018 20:51	Tilknyttede	Geita	2 timer 8 minutter
[REDACTED]	28.06.2018 17:30	28.06.2018 19:34	Tilknyttede		2 timer 3 minutter
[REDACTED]	02.07.2018 15:50	02.07.2018 16:51	Tilknyttede	Geita	1 time 1 minutt
[REDACTED]	02.07.2018 18:09	02.07.2018 20:25	Tilknyttede	Rolig. 60101 mangler 1 lam, ellers stemte det. Noe ravn sør BlØyvangen men fant ikke noe	2 timer 15 minutter
[REDACTED]	01.07.2018 09:00	01.07.2018 12:00	Tilknyttede	Torolf Hagen BlØyvang&aering;n	3 timer 0 minutter
[REDACTED]	27.06.2018 06:38	27.06.2018 17:43	Tilknyttede	Glitterdalen	11 timer 5 minutter
[REDACTED]	23.06.2018 10:00	23.06.2018 14:00	Tilknyttede	Snaupallen, t-stien, Frylibekken Øverst. Såg litt sau. Roleg	4 timer 0 minutter
[REDACTED]	05.07.2018 16:00	05.07.2018 18:00	Tilknyttede	Vann til sauene på setra	2 timer 0 minutter
[REDACTED]	16.07.2018 09:38	16.07.2018 10:37	Tilknyttede		0 timer 58 minutter

16.07.2018	16.07.2018	Tilknyttede	Så vomiskinn og ca 20 dyr	0 timer 44 minutter
09:38	10:23			
20.07.2018	20.07.2018	Tilknyttede		3 timer 0 minutter
16:07	19:08			
23.07.2018	23.07.2018	Tilknyttede		4 timer 38 minutter
09:35	14:13			
24.07.2018	24.07.2018	Tilknyttede	Sett ca 100 dyr. Rolig	4 timer 49 minutter
10:42	15:32			
24.07.2018	24.07.2018	Tilknyttede		0 timer 0 minutter
19:43	19:43			
24.07.2018	24.07.2018	Tilknyttede		0 timer 0 minutter
19:43	19:43			
24.07.2018	24.07.2018	Tilknyttede		3 timer 20 minutter
10:04	13:24			
25.07.2018	25.07.2018	Tilknyttede	Har sett ca 100 dyr mye Øverst langs gjØråa	5 timer 25 minutter
09:34	15:00			
26.07.2018	26.07.2018	Tilknyttede	KlufttjØnn	4 timer 51 minutter
11:07	15:59			
28.07.2018	28.07.2018	Tilknyttede	Fant en bjØlle i skogskvea. Telte over20 dyr.	0 timer 53 minutter
13:03	13:56			
29.07.2018	25.08.2018	Tilknyttede		649 timer 8 minutter
13:06	14:15			
29.07.2018	29.07.2018	Tilknyttede	Fra tjØnnbakken over Øvre Tverrglitra til RanglartjØnm og til tjØnnbakken mye sau RAGNHILD og Andreas gikk på SkjerellhØ	4 timer 58 minutter
13:07	18:06			
04.08.2018	04.08.2018	Tilknyttede	SØye og lam fast i Hovdemyra. Redda, god forfatning	1 time 14 minutter
11:45	12:59			
11.08.2018	11.08.2018	Tilknyttede	Dau ved start orange	2 timer 44 minutter
09:45	12:30			
12.08.2018	12.08.2018	Tilknyttede	Rundt Skjerellfjell. Kadaver i Steinbudalen, Ørn i Steinbudalen. Mye sau langt inn Vesllegret og mjØlrakkhaugan.	4 timer 14 minutter
10:33	14:48			
17.08.2018	17.08.2018	Tilknyttede	60101 og 60310 m 4 lam på BlØyvangen	2 timer 5 minutter
15:34	17:39			
14.08.2018	14.08.2018	Tilknyttede		2 timer 11 minutter
11:37	13:48			
23.08.2018	23.08.2018	Tilknyttede	30 dyr ved vannene på toppen	3 timer 1 minutt
10:39	13:40			
24.08.2018	24.08.2018	Tilknyttede	Sau hØgt oppe i Buåa på vestsiden ellers ikke sett sau	3 timer 24 minutter
09:59	13:24			
31.08.2018	31.08.2018	Tilknyttede		4 timer 12 minutter
12:58	17:10			
01.10.2018	01.10.2018	Tilknyttede		0 timer 0 minutter
14:50	14:51			
24.07.2018	24.07.2018	Tilknyttede	Nysetra- hytta	5 timer 13 minutter
11:00	16:13			
25.07.2018	25.07.2018			5 timer 0

10:00	15:00	Tilknyttede Hytta-Klufftjønn - hytta	minutter
29.07.2018	29.07.2018	Tilknyttede Tjønnbakken - skjerellhø- Glitterdalen- Ranglartjønn- Tjønnbakken	6 timer 0 minutter
12:00	18:00		
04.08.2018	04.08.2018	Tilknyttede Henta hjem geitene	1 time 0 minutter
16:30	17:30		
09.08.2018	09.08.2018	Tilknyttede Leita etter killing	1 time 0 minutter
16:00	17:00		
11.08.2018	11.08.2018	Tilknyttede Gått oppi med geitene	1 time 0 minutter
17:00	18:00		
17.08.2018	17.08.2018	Tilknyttede Jønnbrustulvegen-Leintjønnin-Sleotjønn-Olasvarden -attende	5 timer 40 minutter
09:20	15:00		
06.08.2018	06.08.2018	Tilknyttede Garmin logg	8 timer 19 minutter
07:30	15:49		
12.08.2018	12.08.2018	Tilknyttede Garmin logg	4 timer 3 minutter
09:05	13:08		
17.08.2018	17.08.2018	Tilknyttede	4 timer 8 minutter
10:02	14:10		
for vanlig tilsyn: 789 timer 24 minutter			
25.08.2018	25.08.2018	Tilknyttede Sanka 16 søyer og 27 lam	4 timer 40 minutter
09:31	14:12		
15.08.2018	15.08.2018	Tilknyttede Sanka 50204 i SØdorp (Skipperstøen)	4 timer 0 minutter
14:00	18:00		
26.08.2018	26.08.2018	Tilknyttede	1 time 42 minutter
10:03	11:45		
27.08.2018	27.08.2018	Tilknyttede 30km	8 timer 0 minutter
07:00	15:00		
30.08.2018	30.08.2018	Tilknyttede 3stk	8 timer 50 minutter
07:15	16:05		
04.09.2018	04.09.2018	Tilknyttede	1 time 45 minutter
12:15	14:00		
07.09.2018	07.09.2018	Tilknyttede	6 timer 7 minutter
08:00	14:07		
08.09.2018	08.09.2018	Tilknyttede	7 timer 0 minutter
07:00	14:00		
12.09.2018	12.09.2018	Tilknyttede	6 timer 0 minutter
08:00	14:00		
14.09.2018	14.09.2018	Tilknyttede	4 timer 0 minutter
09:00	13:00		
23.09.2018	23.09.2018	Tilknyttede	2 timer 48 minutter
13:12	16:00		
27.09.2018	27.09.2018	Tilknyttede	11 timer 0 minutter
05:00	16:00		
03.10.2018	04.10.2018	Tilknyttede Bjørnolia Musvoldalen m Andreas	25 timer 0 minutter
15:00	16:00		

Timeforbruk for sankning: 90 timer 52 minutter

Totalt timeforbruk: 880 timer 16 minutter



Kviknebaksida 25
2640 Vinstra

kontakt@beitesnap.no

Tillegg B

Brukertest

B.1 Oppgaver brukt i brukertest av prototype 2

1. Opprett en bruker. Registrer kommune, gårdsnummer og bruksnummer.
2. Start en ny tilsynstur. Tittelen på turen skal være «Demotur», og beskrivelsen skal være «Brukertest av appen».
3. Du får en viktig telefon, og blir nødt til å utsette tilsynsturen. Avbryt turen i appen.
4. Du er ferdig med telefonsamtalen, og starter en ny tilsynstur. Tittel og beskrivelse er valgfritt.
5. Du observerer en svart søye på toppen av stabburet. Søyen har rødt slips og en flenge i høyre side. Registrer observasjonen i appen.
6. Du observerer nå en kongeørn som sirkler over nærmeste nabo tomt. Registrer observasjonen i appen.
7. Du oppdager et sauekadaver midt på veien, rett ved innkjørselen din. Det er en brun søye, og det ser ut som den har blitt drept av en ulv. Den har et grønt slips.
8. Du finner nå en saueflokk i skråningen rett bak huset. Flokken består av både søyer og lam i forskjellige farger. Du ser to hvite og en svart søye. Du ser tre hvite lam, hvor ett av lammene halter på venstre bakben. Du ser også to svarte lam og ett brunt lam.
9. Hvis du ikke ennå har gjort det, lagrer du nå observasjonen.
10. Til slutt oppdager du en brun ulldott flytende i midten av elva bak huset, og registrerer dette i appen.
11. Du er nå ferdig med tilsynsturen, og lagrer denne.
12. Se om du kan finne den lagrede tilsynsturen for å inspisere informasjonen du har registrert.

