



Forside

IS-304: 2024

Tittel: Tilgjengeliggjøring av nasjonale kartdata til bruk i AI-algoritmer

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen og Geir Inge Hausvik
Veileder	Geir Inge Hausvik
Oppdragsgiver:	Tietoevry og Kartverket

Studenter:

Etternavn	Fornavn
Vårdal,	Cassandra Hartvedt
Svartsund,	Gustav
Båshus,	Tørres Andreas
Bakken,	Victor
Saitova,	Larisa

Vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA	
Kan besvarelsen brukes til undervisningsformål?	JA	
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA	

Forord

Denne rapporten dokumenterer prosjektet *“Tilgjengeliggjøring av nasjonale kartdata til bruk i AI-algoritmer”* og er en videreutvikling fra tidligere bachelorprosjekt *“Automatisering av dataflyt fra nasjonale kartbaser til AI-algoritmer”*. Rapporten tar for seg de viktigste avgjørelsene, rammeverkene brukt i prosjektet og prosjektprosessen.

Først, så ønsker gruppen å rette en takk til Tietoevry og Kartverket som har bidratt med ressurser, støtte og oppfølging i dette prosjektet.

Vi ønsker å rette en spesiell takk til Sander Jyhne som har opptrådt som produkteier på vegne av Kartverket og som gav gruppen fritt spillerom til å gjøre dette prosjektet til sitt eget. Takk til Alexander Nossum og Norkart, som har bidratt som en ekstern samarbeidspartner, har arrangert demomøter, og kommet med verdifull kunnskap og generell assistanse. Takk til Roar Engen fra Tietoevry som har bidratt med verdifull kompetanse innenfor prosjektstyring og metodikk. Takk til Lars Fredrik Gyland for hans ekspertise og gode humør gjennom prosjektet. Takk til vår kontaktperson ved Kartverket, Amna Drace Biscevic. Vi ønsker også å takke UiA-veileder Geir Inge Hausvik og Hallgeir Nilsen for sine forelesninger, og takk til UiA for muligheten til å jobbe så tett på bedrifter i IS-304. Det har vært et meget spennende og lærerikt semester.

Avslutningsvis ønsker vi å rette en siste takk til alle som har bidratt og vist engasjement for prosjektet vårt. Deres interesse og støtte, har vært meget motiverende. Vi setter stor pris på tilbakemeldingene vi har mottatt. Takk for at dere har fulgt oss på reisen og bidratt til suksess i prosjektet.

KARTAI



NORKART



Abstrakt

Denne rapporten beskriver utviklingsprosessen for en applikasjon for generering av treningsdata til bruk for maskinlæringalgoritmer, og er en fortsettelse på et tidligere bachelorprosjekt fra våren 2023 med tittelen “*Automatisering av dataflyt fra nasjonale kartbaser mot AI-algoritmer*”. Prosjektet ble gjennomført i samarbeid med Tietoenvry og Kartverket og går inn under et større forskningsprosjekt med navn KartAI. Målet med prosjektet er å tilgjengeliggjøre genereringen av kartdata som videre kan brukes til å trene maskinlæringsmodeller. Rapporten dekker hva som ble gjort under prosjektperioden som gikk fra januar 2024 til mai 2024.

Prosjektrammene gitt av produkteier var veldig åpne, og derfor ble rammeverket for scrum anvendt for prosjektstyring. Scrum artefaktene brukt gjennom prosjektet har vært *daily standup*, *sprint planning*, *sprint reviews* og *sprint retrospective*. Disse har bidratt til en agil utviklingsprosess. Grunnet gruppens krav til kvalitet ble det etablert lignende krav som god dokumentasjon, oppholdelse av kodenstandard og andre kvalitetssikringstiltak.

Brukertester gjort på fjorårets versjon av applikasjonen viste at den var lite brukervennlig. Det var derfor ikke ønskelig å publisere denne versjonen på nett, ettersom frontend delen av applikasjonen måtte rekonstrueres. Gruppen utnyttet teknologier brukt gjennom tidligere prosjekt, slik som Python, HTML, CSS, JavaScript med Bootstrap og Leaflet.

Gruppen hadde manglende kompetanse innenfor tidligere brukte teknologier i prosjektet, slik som Python. Med grundig kompetanseheving og disiplinert jobbing ble det derimot utviklet støtte for flere typer datakilder, og i følge fylkeskartsjefen i Kartverket Agder var vi også først i Norge til å oppnå brukerdefinert styling av fasitdata.

En gjennomgang av den endelige løsningen kan bli sett her:

<https://www.youtube.com/watch?v=Ztaiu1blkx8&t>

Applikasjonen er tilgjengelig på nett:

<https://ngis.azurewebsites.net/>

Github repository kan sees her:

<https://github.com/kartAI/NGIS-training-machine-webclient/>

Innholdsfortegnelse

Forord.....	2
Abstrakt.....	3
Innholdsfortegnelse.....	4
1. Introduksjon.....	7
1.1 Prosjektbeskrivelse.....	8
1.2 Sentrale begreper.....	8
1.3 Gruppen.....	10
1.4 Interessenter.....	11
1.5 Mål og krav.....	11
2. Prosjektstyring.....	13
2.1 Metodikk.....	13
2.2 Kvalitetssikring.....	17
2.2.1 Intern kvalitet.....	17
2.2.2 Ekstern kvalitet.....	19
3. Prosjektgjennomføring.....	20
3.1 Valg av verktøy og teknologier.....	20
3.2 Sprintene.....	23
3.3 Kartlegging.....	24
3.3.1 Scope.....	25
3.3.2 Roadmap.....	25
3.3.3 Risiko og risikomatrise.....	27
3.3.4 Kapasitetsplan.....	28
4. Designprosessen.....	29
4.1 Kartlegging av brukerbehov.....	29
4.1.1 Brukertestning.....	30
4.1.2 Gjennomføring av brukertestene.....	30
4.1.3 Resultater fra brukertester runde 1.....	31
4.2.1 Tilgjengelighetstesting mot WCAG.....	31
4.2.2 Resultater fra automatisert tilgjengelighetstesting.....	31
4.3.1 Resultater fra heuristisk evaluering.....	32
4.2 Planlegging av UX/UI Design.....	34
4.2.1 Skissering og lo-fi prototyping.....	34
4.2.2 High-fi prototyping.....	34
4.3 Modellering av backend.....	35
4.3.1 Original dataflyt.....	35
4.3.2 Tiltent dataflyt.....	36
4.3.3 Endelig dataflyt og systemarkitektur.....	37
5. Utviklingsprosessen.....	39
5.1 Distribuerings av web-løsning.....	39
5.2 Implementering av nytt brukergrensesnitt.....	40
5.2.1 Forbedring av UI/UX basert på brukertester.....	40

5.2.2 Resultater fra brukertester runde 2.....	42
5.3 Implementering av nye datakilder.....	43
5.4 Refaktorering, dokumentasjon og testing.....	44
6. Resultat.....	45
6.1 Oppfylling av mål og krav.....	45
6.2 Applikasjonen og hvordan den brukes.....	46
7. Refleksjon.....	49
7.1 Prosjektarbeid.....	49
7.1.1 Agil arbeidsmetodikk.....	49
7.1.2 Prosjektstyringsverktøy.....	49
7.1.3 Kvalitetssikring.....	50
7.1.4 Ulike interessenter.....	50
7.1.5 Hybrid arbeidsmetode.....	51
7.1.6 Tilgang til ressurser.....	51
7.1.7 Personvern og GDPR.....	51
7.2 Erfaringer med teknologier og utvikling.....	52
7.2.1 Kompetansegap.....	52
7.2.2 Videreutvikling av et tidligere studentprosjekt.....	52
7.2.3 Bruk av ChatGPT og andre AI-verktøy.....	53
8. Konklusjon.....	54
Litteraturliste.....	55
Vedlegg.....	63
Vedlegg A - Gruppekontrakt.....	63
Vedlegg B - Utdrag fra kapasitetsplan.....	65
Vedlegg C - Utdrag fra loggførte timer.....	66
Vedlegg D - Utdrag fra risikomatrise.....	67
Vedlegg E - Utdrag med reviderte roadmaps.....	68
Vedlegg F - Utdrag fra sprint dagbok.....	69
Vedlegg G - Utdrag fra sprintplaner.....	70
Vedlegg H - Utdrag fra beslutningsoversikt.....	71
Vedlegg I - Utdrag fra brukerhistorier.....	72
Vedlegg J - Brukertest mal.....	73
Vedlegg K - Utdrag fra brukertest 1, våren 2024.....	74
Vedlegg L - Utdrag fra brukertest runde 2, våren 2024.....	75
Vedlegg M - Utdrag fra ordbok.....	76
Vedlegg N - Low fidelity skisse.....	77
Vedlegg O - Figma prototype.....	77
Vedlegg P - Instruksjon og forside.....	78
Vedlegg Q - Kodedokumentasjon.....	79
Vedlegg R - Selvevaluering.....	80
Vedlegg S - Uttalelse fra arbeidsgiver.....	83
Vedlegg T - MoSCoW analyse.....	85

Figurliste

Figur 1: Roadmap per 15.01.....	26
Figur 2: Roadmap per 25.03.....	27
Figur 3: Uttak fra risikomatrise.....	28
Figur 4: Kapasitetsplan.....	29
Figur 5: Siden for å skrive inn koordinater.....	31
Figur 6: Siden for settings.....	31
Figur 7: Gammel forside.....	33
Figur 8: Endelig modell av dataflyt til løsningen (Bredesen et al., 2023).....	36
Figur 9: Første dataflyt.....	37
Figur 10: Endelig dataflyt.....	38
Figur 11: Popup vindu fra settings siden	41
Figur 12: Input grenser.....	41
Figur 13: Ny og oppdatert versjon.....	42
Figur 14: Forside applikasjon.....	46
Figur 15: Valg av metode.....	47
Figur 16: Tegn på kartet.....	47
Figur 17: Settings.....	48

1. Introduksjon

Stedfestede data utgjør en stadig større andel av dataen som vi omgir oss med daglig. Fagfeltet Geografiske Informasjonssystemer (GIS) handler om å samle inn, analysere, visualisere og organisere stedfestede data (Universitetet i Bergen, u.å.). I Norge har vi hatt en lang tradisjon med geografiske informasjonssystemer. Eksempelvis ble SOSI standarden utviklet på 1980 tallet (National Archives, 2020). Dette var den første nasjonale standarden for GIS som ble utviklet i Norge, og siden den gang har det vært stor fremgang innenfor sektoren. Dagens GIS-systemer er stadig i endring og forbrukere forventer stadig bedre og mer nøyaktige geografiske systemer. For å kunne tilby disse systemene tar flere utviklere i dag i bruk kunstig intelligens i kombinasjon med GIS-verktøy for å øke effektiviteten og brukervennligheten i systemer som er avhengig av stedfestet data.

Bruken av kunstig intelligens innenfor geomatikk og GIS er ikke noe nytt. Siden tidlig på 2000-tallet har man utforsket mulighetene for å bearbeide stedfestede data ved hjelp av nevralt nettverk og maskinlæring (Voženilek, 2009, s 279-280). Kombinasjonen av kunstig intelligens og geografiske informasjonssystemer kalles GeoAI og fagfeltet har hatt en enorm vekst de siste årene (Esri, u.å.). Denne veksten skyldes trolig at kunstig intelligens har blitt allemannseie etter at OpenAI lanserte ChatGPT i 2020 (Marr, 2023). For mange har ChatGPT dermed blitt det fremste eksemplet på kunstig intelligens. Generativ kunstig intelligens blir også utforsket til bruk i GeoAI og flere aktører har sett på mulighetene for å bruke generativ kunstig intelligens til å bearbeide de enorme mengdene stedfestede data (Li & Ning, 2023).

I Norge har også Kartverket begynt å se på mulighetene til å ta i bruk kunstig intelligens i sine systemer. I første omgang gjennomføres pilotprosjektet KartAI som ser på mulighetene for å kunne bruke kunstig intelligens sammen med stedfestede data for å kunne tilby bedre tjenester til innbyggerne i Kristiansand. KartAI har flere underprosjekter, og denne rapporten beskriver arbeidet med ett av dem. Rapporten tar for seg arbeidet som er gjort i prosjektet "Tilgjengeliggjøring av nasjonale kartdata til bruk i AI-algoritmer". Prosjektet ble gjennomført vårsemesteret 2024 og er en avsluttende bacheloroppgave for studiet IT og informasjonssystemer ved Universitetet i Agder. Rapporten dokumenterer gjennomføringen av prosjektet fra planleggingsfasen til og med levering av det ferdige produktet, samt veien videre for prosjektet. Rapporten tar også for seg hvordan gruppen har jobbet med agile metodikker, tilegnet seg viktig domenekunnskap og brukt kompetansen opparbeidet gjennom studie til å videreutvikle prosjektet.

Først blir prosjektet fremstilt og viktige begreper innenfor geomatikk, geografiske informasjonssystemer og metodikk blir definert. Deretter presenteres gruppen, interessentene og målene for prosjektet. Rapporten dekker også hvilke avgjørelser som ble tatt i de ulike sprintene og utviklingsprosessen, fra brukertesting til ferdig produkt. Til

slutt diskuteres gruppens egen innsats i arbeidet samt refleksjoner rundt det som har blitt gjort gjennom prosjektperioden.

1.1 Prosjektbeskrivelse

Bachelorprosjektet som gruppen har gjennomført våren 2024 inngår i forskningsprosjektet KartAI. På denne tiden deltok totalt fire respektive studentgrupper fra både UiA Kristiansand og UiA Grimstad i prosjektet. Målet for KartAI er som følgende:

Hovedmålet til KartAI prosjektet er at saksgangen for byggesaker i kommunen skal bli mer effektiv. Dette skal gjøres ved å heve kvaliteten til eiendomsregisteret (matrikkelen) og kartbaser ved hjelp av kunstig intelligens. I tillegg skal det utvikles automatiserte prosesser som går i dialog med innbygger eller grunneier (KartAI, u. å).

Prosjektet som gruppen har valgt er en videreutvikling av et prosjekt fra våren 2023. Dette prosjektet gikk under tittelen "Automatisering av dataflyt fra nasjonale kartbaser mot AI-algoritmer", og applikasjonen som ble utviklet fikk navnet NGIS Training Machine. Hovedoppgaven til NGIS Training Machine er å hente og generere kartdata for bruk i maskinlæringsalgoritmer til kunstig intelligens (Bredesen et al., 2023 s. 2). Arbeidet gruppen har gjennomført dette semesteret har gått på å videre tilgjengeliggjøre disse dataene gjennom å gjøre applikasjonen mer brukervennlig for nye brukere samt å inkludere flere datakilder for kartdataene som genereres. Ved å inkludere flere datakilder får man muligheten til å velge områder fra hele Norge istedenfor de predefinerte områdene som før var tilgjengelige. I tillegg til dette har store deler av kodebasen blitt skrevet på nytt. Dette ble gjort for å gjøre koden mer brukervennlig, og lettere å videreutvikle i fremtiden.

1.2 Sentrale begreper

I denne prosjektrapporten blir det brukt mange begreper som er sentrale for å forstå innholdet. Disse begrepene er ikke del av dagligtalen og dette delkapittelet er skrevet for å gi et overblikk over de forskjellige fagbegrepene som er nødvendig å forstå for å kunne lese og forstå rapporten.

Web map service (WMS) - En WMS er "En internasjonal standard for utveksling av elektroniske kart over internett, utarbeidet av Open Geospatial Consortium. Ved hjelp av WMS kan en bruker bygge opp et kartbilde i egen nettleser ved å hente data fra en eller flere tilbydere av kartdata." (Mæhlum, 2023).

En bruker av en WMS kaller opp en WMS-tjener, der det blir spesifisert ønsket karttema og geografisk område. Forskjellige kartlag fra samme eller ulike kartdatabaser legges delvis usynlige på hverandre til et kombinert kart er vist frem i brukerens nettleser. Forespørselen

fra klient til server skjer ved at en URL med de tilhørende spesifikasjonene og riktige parametere sendes tilbake til klienten, slik at den kan se hva som er etterspurt (Mæhlum, 2023).

Styled Layer Descriptor (SLD) - Styled layer descriptor, også kalt SLD, jobber opp mot WMSen. Den utvider hvordan bildet fra WMSen kommer ut, ved hjelp av brukerdefinert “styling” som gjør at brukeren selv kan velge hvordan bildet skal se ut (Open Geospatial Consortium, u.å.).

Felles kartdatabase (FKB) - “FKB er en samling av primærdatasett som samles inn og forvaltes i fellesskap av Geovekst-partene i en kommune” (Kartverket, 2024). Alle kartdata og produktspesifikasjonene for Norge samles i denne felles databasen og kan hentes på [Geonorge](#) (Kartverket, 2024).

GeoTIFF - GeoTIFF er en filtype som inneholder geografiske metadata som beskriver innholdet i bildet, kartet, satellittbilde eller digitaliserte kart slik at de kan bli brukt videre i geografiske informasjonssystemer i form av en TIFF fil. Dataen som blir samlet i filen er blant annet kartprojeksjon, koordinater, datum og matematiske regler for kartprojeksjon. (HEAVY.AI, u.å.).

Cloud-Optimized GeoTIFF (COG) - Er en vanlig GeoTIFF fil som skal bli hostet på en HTTP filserver, med en intern organisasjon som lager mer effektiv “workflow” i skyen. Dette betyr at man har mulighet til å kun spørre etter delene av TIFF som filen man trenger istedenfor hele filen (Cogeo, u.å.).

Ortofoto - Ortofoto er et flyfoto tatt over et geografisk område. Dette bildet har de samme egenskapene som et kart da det følger samme målestokk. Hensikten med ortofoto er at det kan gi informasjon om landoverflaten som et kart ikke gir. Dette innebærer mønstre, farger og teksturer. Det mangler fortsatt mange muligheter som kart har, som generalisering og framheving av vesentlige elementer og temaer (Mæhlum & Berg, 2023).

KI (Artificial intelligence/AI) - “Kunstig intelligens er informasjonsteknologi som justerer sin egen aktivitet og derfor tilsynelatende fremstår som intelligent.” (Tidemann, 2023).

GIS - GIS står for Geografisk informasjonssystem. Disse verktøyene brukes for å visualisere og analysere romlige data. (Universitetet i Bergen, u.å.).

Nasjonalt geografisk informasjonssystem (NGIS) - “NGIS er en forvaltnings plattform der geografiske data blir direkte oppdatert i en sentral database hos Kartverket. Kartdatabasen gir alle brukere tilgang til ferske og kvalitetssikrede data.” (Kartverket, u.å.).

Vektordata og Rasterdata - Datasettene fra Geonorge kommer i flere typer formater ettersom forskjellige kartprogrammer bruker ulike formater (Geonorge, u.å.).

- **Vektordata** betyr at geometrien består av punkt, linjer eller flater. Et eksempel på vektordata er høydedata. Hvor høydekurver blir representert som linjer, og topper blir representert som punkter (Geonorge, u.å.).
- **Rasterdata** er et annet format som består av ruter og/eller pixler med en gitt verdi. "Et eksempel på dette kan være flyfoto eller en bildefil (TIFF/JPG) som består av en fargeverdi for hver piksel i bildet." (Geonorge, u.å.).

QGIS - QGIS er et open source geografisk informasjonssystem verktøy som er et offisielt produkt av "Open source geospatial foundation" (QGIS, u.å.). QGIS kan blant annet brukes til å visualisere, redigere og analysere eksisterende kart (QGIS, u.å.).

FlatGeoBuf (FGB) - FGB er et filformat som er binært enkodet og spesielt utviklet for geografiske data (Flatgeobuf, u.å.).

1.3 Gruppen

Studentgruppen er sammensatt av fem studenter fra bachelorprogrammet IT og informasjonssystemer ved Universitetet i Agder. Alle studentene går siste året og flesteparten av studentene har jobbet sammen siden første semester med gode resultater. Ettersom de fleste i gruppen har samarbeidet siden første semester så har gruppen en god gruppedynamikk. Det har gjort prosjektarbeidet enklere og mer effektivt samtidig som gruppemedlemmene er trygge på hverandre.

I dette bachelorprosjektet fordelte gruppen seg på følgende ulike roller og ansvarsområder:

Victor Bakken	Backend-utvikler og ansvarlig for cloud-løsningen (Azure)
Tørres Båshus	Prosjektleder, scrum-master, UX/UI-designer & frontend-utvikler
Larisa Saitova	UX/UI-designer & frontend-utvikler
Gustav Svartsund	Backend-utvikler og ansvarlig for rapportskrivningen
Cassandra Vårdal	Backend-utvikler og teknisk ansvarlig

Flesteparten av medlemmene arbeidet på kontorene til Tietoevry og Kartverket mens ett av gruppemedlemmene var i Nederland for mesteparten av semesteret. Dette har ført til at gruppen har måtte adaptere en hybrid arbeidsmodell. Dette hadde ikke vært mulig uten det gode samholdet og den tilliten gruppens medlemmer har utviklet til hverandre gjennom

tidligere semestre. Den hybride arbeidsmåten fungerte bra og førte ikke til store hindringer i prosjektperioden. Under en slik arbeidssituasjon er det viktig å sikre at arbeidet blir gjennomført. Det ble derfor skrevet en gruppekontrakt som kan leses i [vedlegg A](#).

1.4 Interessenter

Dette bachelorprosjektet inngår i KartAI prosjektet som er et samarbeid mellom Norkart, Kartverket, UiA og Kristiansand kommune (KartAI, u.å.). Samtidig har Tietoevry fungert som veileder innenfor metodikk og prosjektstyring og gitt gruppen tilgang til kontorplasser.

Tietoevry er et nordisk konsulentselskap med 24000 ansatte spredt på kontorer i over 20 land (Tietoevry, u.å.). Bachelorprosjektet ble gjennomført i samarbeid med Tietoevry som bidro med metodikk, administrasjon, kontorlokaler og andre ressurser.

Kartverket er en offentlig teknisk etat som har ansvaret for å forvalte geografisk og stadfestet informasjon og videreformidle denne i diverse tjenester (Kartverket, 2023). Kartverket fungerte som oppdragsgiver for bachelorprosjektet, og Sander Jyhne ved Kartverket fungerte som produkteier. Bachelorprosjektet er knyttet opp mot hans doktorgrad hos Kartverket.

Norkart er et norsk teknologiselskap som tilbyr markedsledende løsninger innen kommunalteknikk, plan og geodata, og øvrig kart og eiendomsinformasjon til offentlig og privat sektor (Norkart, u.å.). Norkart bidro med kompetanse og ressurser for prosjektarbeidet som blant annet Microsoft Azure.

1.5 Mål og krav

Det overordnede målet for gruppens prosjekt var å videreutvikle NGIS Training Machine. Dette er en applikasjon for å generere treningsdata til maskinlæring som ble startet av en studentgruppe våren 2023. Deres hovedoppgave var å utvikle en selvbetjeningsportal der man kunne hente ut treningsdata (Bredesen et al., 2023, s.9). Gruppen ble presentert med følgende deloppgaver som en mulig videreutvikling av prosjektet

- Videreutvikle NGIS Training Machine til å støtte 3D-data og dynamisk AI-Styling på kartservere
- Utvikle COG-connector og DuckDB-støtte for input i KartAI-Stream
- Lage utvikler-rammeverk for MapAI 2.0 - Internasjonal AI-konkurranse.

Dette var målene som oppdragsgiver så for seg for bachelorprosjektet 2024. Målene samsvarte derimot ikke med produkteierens ønsker for prosjektet, og under gruppens første møte med produkteier fra Kartverket fikk ble det definert nye mål. I tillegg fikk

gruppen fritt spillerom til å definere egne mål for hva de følte at prosjektet burde inneholde. Da var det viktig å ta i betraktning målgruppen for applikasjonen.

Målgruppen for applikasjonen ville være folk med interesse i å bruke kartdata for trening av AI-modeller. Potensielle brukere ville dermed ha grunnleggende kunnskaper om kart og sannsynligvis også en grunnleggende forståelse eller interesse for hvordan å trene opp en AI-modell. Kartverket så for seg at denne applikasjonen skulle vekke større samfunnsinteresse for kart og GIS. Målene gruppen definerte innebar derfor forbedringer av brukeropplevelsen, da den daværende løsningen ikke var brukervennlig for personer uten teknisk bakgrunn.

Det første kravet fra produkteier var å gjøre løsningen tilgjengelig på nett gjennom Kartverkets Azure portal. Deretter var det ønskelig å supplere selve NGIS delen av applikasjonen siden denne krever diverse tilganger med flere andre datakilder. Gruppen samlet derfor alle innspillene fra produkteieren og sammen med produkteier ble det gjort om til en liste med mål for prosjektet.

Prioritert målsetting etter første møte med oppdragsgiver

1. Distribudere applikasjonen på nett via Microsoft Azure
2. Legge til nye datakilder for WMS, COG, FGB og NGIS
3. Refaktorere eksisterende kode til noe som er mer forståelig og lettere å videreutvikle
4. Forbedre brukervennligheten for applikasjonen

Disse målene skulle utgjøre scopet for gruppens bachelorprosjekt. Scopet ble sett på som flytende og punktene ble sett på som en prioriteringsliste.

For hvert mål lagde gruppen en eller flere brukerhistorier, og for hver brukerhistorie ble det definert et sett med akseptansekriterier som måtte bli møtt for at brukerhistorien skulle vurderes som oppfylt (Scrum Alliance, u.å.). Når alle akseptansekriteriene for et mål var oppfylt, ble målet vurdert som møtt.

Et eksempel er brukerhistorien for å kunne velge mellom ulike datakilder. Denne brukerhistorien tilhørte målsettingen "Legge til nye datakilder for WMS, COG, FGB og NGIS".

Tittel:	Velge mellom datakilder
Behov:	Som bruker vil jeg ha muligheten til å kunne velge hvilke datakilder treningsdataen min kommer fra.
Kriterier:	• Det er mulig for brukeren å velge mellom datakilder via en drop down eller knapper. • Datakilden som brukeren velger må kunne hente ned dataen brukeren spør etter. • Dataen som blir hentet ned må samsvare med området brukeren har valgt å hente data for.
Fordel:	Det å kunne velge mellom forskjellige datakilder gir brukeren muligheten til å skreddersy treningsdata etter eget behov.

Alle brukerhistoriene er tilgjengelig i [vedlegg I](#).

2. Prosjektstyring

Prosjektstyring kan defineres som “etablering av mål, planlegging av aktiviteter og oppfølging av gjennomføringen av et prosjekt” (Rolstadås, 2020). Dette er essensielt for å gjennomføre et vellykket utviklingsprosjekt.

I dette kapittelet beskriver gruppen metodikken og verktøy som ble brukt for å sikre at prosjektet ble styrt og gjennomført på en ansvarlig måte. Etter ønske fra oppdragsgiver valgte gruppen å følge metodikken scrum. Kapittelet vil først dekke de ulike delene av scrum metodikken og hvordan disse ble implementert i prosjektet. Deretter blir det beskrevet hvilke tiltak gruppen tok for å sikre kvalitet i prosjektet.

2.1 Metodikk

Agile metodikker har sett en enorm vekst i popularitet de siste årene blant en mengde selskaper som strekker seg fra IT og radio til traktor og vinproduksjon (Ribgy et al., 2016). Scrum metodikken utviklet på 90-tallet av Jeff Sutherland og Ken Schwaber, er et av de mest populære rammeverkene blant disse, og er særlig utbredt i IT-bransjen etter at fossefall metodikken gradvis har mistet popularitet. Scrum er et rammeverk designet for organisasjoner og kryssfunksjonelle team med omfattende prosjekter for å finne tilpasningsdyktige løsninger på komplekse problemer. Scrum gjør programvareutvikling enklere ved å bryte ned store, komplekse oppgaver i mindre deler som så gjennomføres i korte, iterative sykluser, også kalt sprinter. Denne metodikken gjør prosjekter mer forutsigbare og reduserer risiko (Sutherland & Schwaber, 2020).

Det var en forventning fra oppdragsgiver Kartverket og Tietoevry at gruppen tok i bruk scrum i prosjektet ettersom de selv brukte denne metodikken. I tillegg ga prosjektets flytende scope en stor grad av frihet til å styre prosjektet selv. Dermed hadde gruppen stor nytte av fleksibiliteten som scrum ga rom for. Tidligere erfaring med scrum og veiledning i scrum metodikken fra Tietoevry gjorde det til et naturlig valg for gruppen.

Rammeverket innebærer en mengde roller, artefakter og aktiviteter som må defineres for å få en helhetlig forståelse av hvordan den kan forbedre en utviklingsprosess. Rollene i et scrum team består av produkteier, scrum master og utvikler (Sutherland & Schwaber, 2020).

Produkteier - Produkteier er ansvarlig for å sørge for at scrum teamet kan yte maksimalt under utviklingen av produktet ved å sette klare rammer for hva sluttproduktet skal inneholde og hvilke oppgaver som burde ligge i produktbackloggen (Sutherland & Schwaber, 2020). Denne rollen ble tildelt doktorgradsstipendiaten som representerte Kartverket, og deltok under sprint review og sprint planning. Grunnet uforutsigbare omstendigheter opptrådte også delvis scrum master som produkteier. Vedkommende som hadde rollen som produkteier fra Kartverkets side gikk ut i pappapermisjon mot slutten av prosjektperioden. Kartverket ble enige med Norkart, som er involvert i KartAI prosjektet, om at Norkart skulle overta rollen. Byttet av produkteier medførte ikke store konsekvenser ettersom scope og prioriteringer allerede var fastsatt og prosjektet gikk inn mot sluttfasen. Produkteieren ga gruppen veiledning i hvilke oppgaver som burde prioriteres for hver sprint, og hadde også visse krav til hvordan funksjonaliteten i applikasjonen skulle virke.

Scrum master - Scrum masteren er ansvarlig for å sørge for at alle deltakende forstår og arbeider i henhold til retningslinjene som er etablert i rammeverket. Dette inkluderer å fasilitere daily scrum, sprint retrospective og sprint review (Sutherland & Schwaber, 2020). Scrum master var Tørres Båshus. Han fikk tildelt denne rollen da han har tidligere erfaring med å lede gruppen og anses å være en solid og klar leder. Ettersom gruppen har jobbet sammen over lengre tid, ble rollen som scrum master stort sett en administrativ rolle. Alle medlemmene i gruppen klarte å holde fokus og være ansvarlige nok til at scrum master aldri trengte å individuelt veilede arbeidet på noe vis.

Utviklere - Utviklerne er de resterende medlemmene av scrum teamet. Ansvarsoppgavene til utviklere er å utvikle og levere sluttproduktet. Dette innebærer blant annet å planlegge sprinter, gjennomføre oppgavene i backloggen og tilegne seg kompetansen nødvendig for å tilfredsstille de fastsatte kravene innen tidsrammen (Sutherland & Schwaber, 2020).

Gruppen valgte å dele seg inn i et frontend-team og et backend-team basert på de ulike styrkene og interessene til gruppemedlemmene. Dette sikret at prosjektet ville kunne se en kontinuerlig og effektiv fremgang på begge fronter, og at de individuelle oppgavene ikke ble for små. Det var også en kollektiv erfaring fra tidligere år at for mange personer på en

oppgave gjorde at medlemmene fort kom i veien for hverandre, og så måtte bruke tid på å få tilbake oversikt. Noe som senket kvaliteten på arbeidet.

Sprinter - I scrum metodikken blir arbeidet organisert i sprinter. Hver sprint er en iterativ prosess hvor teamet planlegger og utfører et sett med oppgaver definert i sprintbackloggen, tester, og til slutt presenterer et funksjonelt produkt til produkteier (Sutherland & Schwaber, 2020). En anbefaling fra Tietoevry var å gi hver sprint en varighet på to uker da dette var standarden hos dem. Gruppen konkluderte at dette var tilstrekkelig med tid til å utføre arbeidsoppgaver med god kvalitet og mulighet for fleksibilitet. Tilsammen ble prosjektet dermed fordelt på 9 sprinter. En sprint består av en rekke artefakter som blir beskrevet under, og en sprint er samlebetegnelsen for kombinasjonen av bruken av alle disse artefaktene.

Sprint planning - Sprint planning er et planleggingsmøte som holdes i starten av hver sprint. Under et slikt møte planlegger man hvilke oppgaver som skal gjøres i løpet av denne perioden og overfører disse fra produktbackloggen til sprintbackloggen (Sutherland & Schwaber, 2020). Hensikten med møtet er å kunne avdekke følgende spørsmål (Sutherland & Schwaber, 2020):

- Hva er hensikten med denne sprinten?
- Hvilket arbeidsoppgaver skal gjøres i løpet av denne sprinten?
- Hvordan skal arbeidet utføres?

Under møtet utarbeidet gruppen ett eller flere større mål som skulle løses i løpet av sprinten. Disse målene tilførte alle en verdi til prosjektet på ulike måter, enten det var innsiktsarbeid eller utvikling. Etter de første ukene med kompetanseheving omhandlet disse både frontend, backend, rapportskrivning og administrative oppgaver som ble fordelt basert på rollene beskrevet ovenfor. Planleggingen ble gjort i samarbeid med produkteier, som med tanke på gruppens tilbakemeldinger og innspill, fikk siste ordet på prioriteringen. De overordnede målene for hver sprint var basert på målsettingene for prosjektet.

Videre ble målene brutt ned til mindre oppgaver for sprintbackloggen. Ifølge rammeverket skal disse arbeidsoppgavene kunne gjennomføres på en dag eller mindre (Sutherland & Schwaber, 2020). Dette viste seg å være krevende å få til på backend delen av prosjektet, ettersom manglende praktisk erfaring med GIS-teknologier gjorde det vanskelig å vite hvordan oppgavene kunne brytes ned. I slike tilfeller ble det opprettet en større researchoppgave i backloggen i forbindelse med den overordnede backend oppgaven, som til sammen fikk allokert en dag eller to. Etter en grundigere utforskning og innhenting av eksterne meninger om hvordan oppgaven kunne løses, ble timer uthentet fra den overordnede backend oppgaven og omfordelt til mindre og mer spesifikke oppgaver.

Et eksempel på dette var gruppens implementering av datakildene COG og FGB. I starten av sprinten ble det opprettet to oppgaver med navn “Undersøk FGB” og “Undersøk COG” som samlet ble tildelt all tiden som var allokert til backend oppgaver. Etterhvert som gruppen fikk et klarere bilde av hvordan disse datakildene skulle implementeres ble det tatt timer fra de store “undersøk” oppgavene og allokert til mindre oppgaver som for eksempel “Last opp COG til Azure” og “Skriv funksjon som gjør kall til COG filen”. Oppgaver med tydeligere arbeidsforløp ble gitt et mer konkret tidsestimat basert på deres kompleksitet.

Daily scrum - Daily scrum er møter som holdes på omtrent samme tidspunkt hver dag. Målet med disse møtene er for gruppemedlemmene å besvare følgende tre spørsmål:

- Hva har du gjort siden sist?
- Hva skal du gjøre idag?
- Er det noe som hindrer deg fra å gjøre arbeidet du har planlagt?

Møtene sikrer at arbeidsflyten opprettholdes og fasiliteterer kommunikasjon i teamet minst én gang per dag (Sutherland & Schwaber, 2020). Gruppen valgte å holde sine daglige scrum møter omtrent klokken 8:15 hver arbeidsdag og dokumenterte disse i en dagbok. Dagboken kan leses i sin helhet i [Vedlegg F](#). Daily scrum ble bevisst holdt kort og lite detaljert. Hovedformålet med daily scrum var å definere dagens agenda, oppdage eventuelle hindringer fortest mulig og adressere disse.

Sprint review - Sprint review er et møte som holdes i starten av en sprint, der man diskuterer den tidligere sprinten sammen med produkteier og avgjør om målene for den forrige sprinten ble nådd. Sprint reviews er designet for å vurdere og eventuelt justere prosjektets scope, arbeidsoppgaver og arbeidsmetoder. For dermed å øke effektiviteten og kvaliteten på nåværende og fremtidige sprinter (Sutherland & Schwaber, 2020). I tillegg brukte gruppen disse møtene til å reflektere over time estimeringer og gjennomgå risikomatriksen. Gruppen holdt møtene første mandag i sprinten rett før sprint planning sammen med produkteier. Dette sikret at kommunikasjonen mellom teamet og produkteier, samt planlegging av veien videre, ble basert på en helhetlig forståelse av prosjektets status.

Sprint retrospective - Sprint retrospective er et kort møte holdt på slutten av hver sprint. Her diskuterte teamet sine positive og mindre positive erfaringer av arbeidet, og hvilke tiltak som burde tas for å heve kvaliteten for arbeidet i kommende sprinter (Sutherland & Schwaber, 2020). Formålet med dette var å drøfte løsninger som kunne effektivisere fremtidig arbeid.

Sprint backlog og produktbacklog - For å få oversikt over, og koordinere oppgaver gjennom hele prosjektet brukte gruppen to scrum artefakter, henholdsvis sprint backlog og produktbacklog. Github projects ble brukt for å holde oversikt over disse oppgavene. Med

grundig nok planlegging og forståelse for prosjektet skal produktbackloggen ideelt inneholde alle nødvendige oppgaver for hele prosjektet fra starten. Dermed vil det i teorien ikke være nødvendig å supplere med flere oppgaver. Dette ble ikke tilfellet for dette prosjektet da produkteier og Kartverket selv var usikre på gjennomførbarheten av spesifikke oppgaver. Dersom sprintbackloggen gikk tom plukket gruppen oppgaver fra produktbackloggen for å fylle opp de resterende timene i sprinten (Sutherland & Schwaber, 2020).

Definition of Done - Definition of done er en beskrivelse av tilstanden til en oppgave når det oppfyller kvalitetsmålene som kreves for produktet. Når en oppgave i produktbackloggen er ferdigstilt, oppfyller den definition of done (Sutherland & Schwaber, 2020). Definition of done ble også anvendt under sprint review for å avgjøre om et mål var nådd eller ikke.

2.2 Kvalitetssikring

Ifølge nasjonal digital læringsarena defineres kvalitetssikring som “systemene og aktivitetene som brukes for å oppnå den kvaliteten som kunden krever på produktet” (Industriskolen, 2020). Å vurdere kvaliteten i et komplekst prosjekt er en omfattende prosess, samtidig som det finnes en mengde ulike forståelser av hva det innebærer å ha god kvalitet i et prosjekt. Likevel er det klar enighet om at kvalitetssikring er avgjørende for å lykkes i et prosjekt. Følgende delkapittel beskriver hvordan gruppen definerer, evaluerer og forbedrer kvaliteten i prosjektet for å sikre at målene beskrevet i kapittel 1 blir oppfylt med riktig kvalitet.

Kvalitetssikring må gjøres både på et internt nivå for gruppen som utvikler og leverer tjenesten, og et eksternt nivå for sluttbrukerne av applikasjonen. Den interne kvalitetssikringen omfatter arbeidsmetoder og verktøy som blir tatt i bruk for å sikre effektiv prosjektstyring og gjennomføring. Slike prosesser skal blant annet bidra til riktig fremdrift, effektiv ressursallokering og samarbeid i prosjektet. I tillegg refererer intern kvalitetssikring av produktet, til hvorvidt systemets oppbygging tillater fleksibilitet, vedlikehold og fremtidig videreutvikling (Swarts, 2015). Ekstern kvalitet omfatter brukervennligheten av systemet og kundeverdien som sluttproduktet produserer for sluttbrukere (Sajith, 2023).

2.2.1 Intern kvalitet

Scrum - Bruk av scrum var et av gruppens mest sentrale kvalitetssikringstiltak. Denne arbeidsmetoden baseres på de tre grunnleggende prinsippene som er *transparency*, *inspection* og *adaptation*. Disse illustreres gjennom kontinuerlig kommunikasjon mellom interessenter, samt regelmessig inspeksjon og tilpasning av arbeidsprosessene og produktet under hver sprint (Sutherland & Schwaber, 2020). Scrum metodikken og de ulike artefaktene var en grunnleggende del av å sikre kvalitet og fremgang i gruppens prosjekt.

Risikomatrise - En risikomatrise er et verktøy innenfor risikostyring som blir brukt for å kartlegge og vurdere innvirkningen av potensielle risikoer som kan oppstå under gjennomføring av et prosjekt. Dette bidrar til en smidigere prosjektgjennomføring og effektiv ressursallokering (Aven T, 2022). Matrisen hjelper til å prioritere risikoene basert på sannsynlighet og konsekvens, og skaper grunnlag for å ta mer informerte valg i henhold til allokering av gruppens ressurser for en smidig arbeidsprosess. Rapporten går dypere inn i hvordan gruppen har vurdert og håndtert risikoer utover prosjektet i delkapittel [3.3.3 Risiko og risikomatrise](#). Risikomatrisen som ble utviklet for prosjektet er også lagt til som [vedlegg D](#) i rapporten.

Dokumentasjon - Gruppen utviklet dokumentasjon for store deler av prosjektet for å planlegge og styre arbeidet effektivt. Gruppens prosjekt krevde betydelig kompetanseheving. Dermed var dokumentasjonen svært viktig for å sørge for at gruppemedlemmene utviklet seg i takt med hverandre og for å minske kompetansegapet i gruppen. For dette ble det blant annet bearbeidet en ordbok for tekniske begreper og verktøy. Se [vedlegg M](#)

Tidlig i prosjektet ble det opprettet en sprintdagbok for å dokumentere daily scrum ut gjennom prosjektet. I dagboken skrev gruppemedlemmene ned hva de hadde gjort siden siste møte, planen for dagen og eventuelle problemer som hindret individuell fremdrift. Her ble også lagt til eventuelle øvrige notater. Sprintdagboken hjalp gruppen holde hverandre oppdatert og ga oversikt over fremdriften i prosjektet. Se [vedlegg F](#) for sprintdagboken.

Ved hvert sprint review ble en plan for den kommende sprinten skrevet. Planen inneholdt mål for sprinten, og hvilke oppgaver som skulle utføres. I planen ble det kartlagt om det var noen behov som måtte oppfylles for å nå målene, eller spesielle tiltak som gruppen måtte ta under sprinten. I plandokumentet så utarbeidet gruppen alle arbeidsoppgavene for sprinten. Målene som ble satt opp under sprint reviewet i planen ble vurdert som nådd eller ikke nådd under neste sprint review. Planen bidro effektiv kommunikasjon og fremdrift i prosjektet. [Vedlegg G](#) viser til alle planene gruppen la i de ulike sprintene.

Github - I tillegg til å bruke github som versjonskontrollsystem ble også flere av funksjonene i github aktivt tatt i bruk for å sikre kvaliteten i prosjektet. De viktigste av disse funksjonene var å sette opp klare regler internt i gruppen for hvordan og når man kunne slå sammen nyutviklet kode med den gamle kodebasen. Gruppen hadde en utviklings- og en main branch. Utviklingsbranchen ble brukt som et midlertidig lagringspunkt for ferdig kode mens main branchen ble brukt kun til kode som var deployet på Azure. For å kunne merge til utviklingsbranchen måtte koden gjennomgå og godkjennes av én annen utvikler i gruppen. For å merge til main måtte alle gruppens medlemmer være enige om at koden fulgte de interne retningslinjene og standardene som var satt for arbeidet. Main ble som hovedregel kun oppdatert før demo møter, der koden skulle være funksjonell uten bugs, slik at vi fikk frem et godt helhetlig bilde.

Kommunikasjon - Kommunikasjon er en viktig del av prosjektstyring og kvalitetskontroll. God kommunikasjon sikrer at alle gruppe medlemmene er på samme, og at ingen sitter uten arbeid i en lengre periode. Ettersom gruppen ikke alltid samlet seg fysisk var de daglige scrum møtene viktige for å holde hverandre oppdatert på fremgangen i prosjektet. I tillegg til disse daglige møtene ble kommunikasjonsverktøyet Discord brukt aktivt for å sikre regelmessig kommunikasjon uavhengig av sted. For å etablere klar separasjon av informasjon, ble ulike epics i prosjektet som for eksempel frontend og backend tildelt egne discord kanaler. Dette bidro til å strukturere arbeidsmiljøet for bedre oversikt, oppgavekoordinering og arbeidsflyt.

Unit testing - En annen viktig del av kvalitetssikringsprosessen for gruppen var å sørge for at koden hadde innebygd kvalitetssikring. Det var derfor en viktig prioritet for gruppen å utvikle unit tester. Formålet med unit tester er å sikre at alle funksjonene i koden fungerer og at man lett kan finne ut hvilke deler som ikke fungerer hvis man gjør endringer (Techtarget, u.å.). Gruppen brukte unit testing aktivt i prosjektet og utviklet tester for stort sett hele kodebasen på eget initiativ for å sikre at utviklingen ble lettere for både oss og fremtidige grupper.

Kodestandard - Kodestandarder er regler for utvikling som sørger for at koden er håndterbar, lesbar og skalerbar (University of St Andrews, u.å.). Å følge en slik standard sørger for at utviklere kan forstå, opprettholde og videreutvikle koden (Codacy, 2023). Gruppen valgte å videreføre prinsipper de hadde lært tidligere i utdanningen slik som cohesion, coupling, riktig navngivning av variabler, funksjoner med klare og konsis dokumentasjon. Kodestandarder varierer ofte ut fra forskjellige programmeringsspråk, og det ble derfor brukt mye tid på å lese seg opp på den aktuelle dokumentasjonen for å sikre kvalitet under utviklingen.

Pythonkoden i prosjektet er basert på det offisielle dokumentet “PEP 8 – Style Guide for Python Code”. Dette dokumentet beskriver hvordan man burde organisere koden samt skrive kommentarer med riktig syntax for å sikre høyest mulig kvalitet på koden (Rossum et al., 2023). For Javascript har gruppen fulgt retningslinjene for hvordan man burde skrive god og forståelig Javascript kode skrevet av Mozilla. (Mozilla, 2023)

2.2.2 Ekstern kvalitet

Brukertesting - For å sikre at applikasjonen tilfredsstiller målgruppens behov er det viktig å samle innsikt direkte fra brukere. Ved å observere brukeropplevelser og uventede systeminteraksjoner kan teamet avdekke hvilke løsninger som fungerer og hvilket som burde endres. Gruppen utviklet kooperative brukertester som ble utført av flere ansatte på Kartverket, som representanter for applikasjonens målgruppe. Tilbakemeldingene fra brukertestene satte grunnlaget for hvilke endringer som skulle gjøres på frontend delen av

applikasjonen. Gjennomføring av brukertester blir beskrevet i delpunkt [4.1.1 Om brukertestene](#).

Heuristisk evaluering - Heuristisk evaluering er en metode innen UX design for å teste og evaluere brukergrensesnitt mot et sett med prinsipper for design. Hvert prinsipp påpeker ulike aspekter av et brukergrensesnitt, og hvordan disse aspektene burde optimaliseres for en best mulig brukeropplevelse (Benyon, 2019, 246-247). Det finnes flere sett med prinsipper for design, men gruppen valgte å bruke Benyons 12 designprinsipper som medlemmene hadde fått en solid forståelse for gjennom studiet. For å sikre en grundig evaluering og forstå hvordan prinsippene kunne oppfylles i størst grad, ble den gjennomført av flere teammedlemmer individuelt. Deretter ble resultatene drøftet i fellesskap. Selve prinsippene blir introdusert i delpunkt [4.1.6 Resultater av heuristisk evaluering](#), mens implementering blir beskrevet i delkapittel [5.2 Implementering av nytt brukergrensesnitt](#).

Automatisert tilgjengelighetstesting - Automatiserte tilgjengelighetstester er skripter som utføres av programvareverktøy for å vurdere tilgjengeligheten av digitale systemer. Automatisert tilgjengelighetstesting sikrer kvalitet ved å bruke tilgjengelighetsstandards for å identifisere problemer som kan hindre personer med funksjonshemninger fra å bruke applikasjonen effektivt (loadfocus, u.å.). Bruk av automatisert testing sikret at gruppen fanget opp problemer i applikasjonen som er vanskelig å legge merke til med manuelle tester. Implementeringen testene er beskrevet i detalj i delkapittel [4.1 Kartlegging av brukerbehov](#).

3. Prosjektgjennomføring

Dette kapittelet inneholder sentrale punkter og drøfting om prosjektgjennomføring. Det beskriver valg av verktøy og teknologier som ble brukt til prosjektplanlegging, utvikling og kommunikasjon.

3.1 Valg av verktøy og teknologier

Dette delkapittelet beskriver hvilke verktøy og teknologier gruppen har valgt å ta i bruk. Flere av disse valgene er videreføringer av valg som ble gjort av studentgruppen som startet prosjektet våren 2023. Selv om valgene er videreføringer, hadde gruppen diskusjoner rundt disse teknologiene for å se om det var bedre måter å implementere prosjektet på.

Azure - Azure er Microsoft sin skybaserte plattform for infrastruktur, administrasjon, og det meste som trengs til forretningsapplikasjon. Disse tjenestene er skalerbare, og tar i bruk virtuelle servere med nettverk, fillagring og database-management. Dette er bare noen av

tjenestene Azure tilbyr en organisasjon, det er også mulig å lage og hente utviklingspakker, også kalt tredjepartsløsninger (Evelon & Smith, 2019). Gruppen valgte å bruke Azure etter ønske fra produkteier. Dette var en helt ny teknologi som gruppen måtte sette seg inn i. Det var Norkart som inviterte til bruk av Azure og gruppen fikk et budsjett på 800 kroner. Dette var nok for å ha en grunnleggende nettside, som dekket behovene for bachelorprosjektet.

Docker - Docker er “en plattform for å utvikle, sende ut og kjøre applikasjoner” (Docker, u.å.). Det er flere grunner til at docker ble valgt. Den første grunnen er at Docker applikasjoner kjører isolert og adskilt fra resten av applikasjonene på maskinen din. Dette øker sikkerheten i applikasjonen og sikrer også at hvis man gjør en feil i Docker-miljøet vil det ikke påvirke resten av maskinen som blir brukt til å kjøre det (Docker, u.å.). En annen grunn til at Docker ble valgt var for å sikre god kompatibilitet med Azure. Denne prosessen blir beskrevet nærmere i delkapittel [5.1 Distribuering av webløsning](#).

FastAPI - FastAPI er et moderne og samtidig raskt webbasert rammeverk basert på Python 3.8+ standarden (FastAPI, u.å.). Dette bachelorprosjektet bruker FastAPI til å knytte sammen frontend og backend i prosjektet. FastAPI var rammeverket som ble brukt av fjorårets studenter for å koble sammen backend og frontend og det ble også videreført av årets studentgruppe grunnet at det er raskt og enkelt å lære.

Figma - Figma er et designverktøy som lar designere samarbeide i sanntid for å designe applikasjoner (Staiano, 2022, s. 4). Dette verktøyet ble valgt fordi gruppen har mye erfaring med det fra tidligere semestre og kjenner til verktøyet godt. I tillegg er dette et gratis verktøy som gir brukere muligheten til å samarbeide i sanntid, noe som er verdifullt når deler av prosjektet ble gjennomført hybrid.

Github - Github er en plattform for å lagre kode, samt et versjonskontrollsystem og samarbeidsverktøy (Github, u.å.). Gruppen valgte å bruke Github fordi det var dette versjonskontrollsystemet vi hadde mest erfaring med fra tidligere i utdanningen. Github var også et naturlig valg ettersom alt av KartAI sin kode allerede er tilgjengelig på plattformen og gruppen fikk tilgang på et repository der.

Github Projects - Github projects er “et agilt og fleksibelt verktøy for planlegging og sporing av arbeid gjort i Github” (Github, u.å.). Github projects var ikke gruppens førstevalg når det kom til verktøy som skal bistå agil arbeidsmetodikk. Gruppen hadde fra tidligere mer erfaring med andre verktøy som Jira, men grunnet mangel på lisenser fra oppdragsgiver måtte gruppen bruke Github projects. Dette viste seg derimot å være et godt verktøy for å holde kontroll på arbeidet, og ble blant annet brukt til å planlegge sprinter, fordele oppgaver og gjøre tidsestimater. Github Projects var primærkilden for prosjektets oversikt.

HTML/CSS - HyperText Markup Language (HTML) og Cascading Style Sheets (CSS) la grunnlaget for brukergrensesnittet i prosjektet. HTML-kode brukes for å strukturere innholdet på nettsiden, og denne strukturen gir muligheten til å videreutvikle med stiler og lignende. (Mozilla Web Docs, 2023). CSS er et språk som definerer hvordan HTML dokumentet skal se ut, rent grafisk (Mozilla Web Docs, 2023). Disse teknologiene er en videreføring fra den forrige studentgruppen ettersom det var dette brukte til å utvikle frontend delen av sitt prosjekt.

Bootstrap - Gruppen brukte også bootstrap til å gi nettsiden stil i tillegg til egenskrevet CSS kode. Bootstrap rammeverket er et populært rammeverk basert på CSS som brukes til å utvikle interaktive og responsive nettsider til alle skjermstørrelser (W3Schools, u.å.). Dette var også en videreføring fra i fjor, men gruppen hadde allerede gode erfaringer med bootstrap rammeverket og var enstemmig enig om å videreføre det.

Javascript - Javascript er et programmeringsspråk som gjør nettsider mer interaktive og gir muligheten til å utvikle mer kompleks atferd (Mozilla Web Docs, 2024). Javascript er også et valg som ble videreført fra den forrige studentgruppen og gruppen så ingen grunn til å bruke et annet språk eller rammeverk grunnet at Javascript er enkelt og effektivt.

Postman - Postman er en API plattform for bygging og bruk av APler. Oppgaven til Postman er å forenkle stegene mellom kode og API-bruk, for eksempel testing (Postman, u.å.). I prosjektet ble Postman brukt til å teste de forskjellige FastAPI endepunktene til applikasjonen uten at man måtte gå gjennom alle stegene i applikasjonen og debugging av HTTP requests.

Python - Python ble brukt til å skrive kode for backend delen av prosjektet. Backend-delen var ansvarlig for å koble til diverse APler og håndtere import av biblioteker. Python ble valgt ettersom det støtter en enorm mengde biblioteker (Python, 2024.). Dette var også en videreføring fra den tidligere studentgruppen og det virket som et godt valg, grunnet store deler av KartAI prosjektet allerede er skrevet i Python.

Simple mail transfer protocol - Simple mail transfer protocol (SMTP) er en standard protokoll for å sende mail over nettverk (Cloudflare, u.å.). Denne protokollen er implementert i flere språk, blant annet i Python gjennom biblioteket smtpplib (Python, 2024). Gruppen valgte å bruke dette biblioteket til å sende mail istedenfor å bruke en ekstern tjeneste som SendGrid fordi den gjorde det enkelt å sende mail, og fordi informasjonen er mer tilgjengelig ved bruk av en SMTP protokoll. Norkart stilte med egendefinert mail til prosjektet, som gruppen kunne bruke.

Visual Studio Code - Visual Studio Code er et program for å redigere og debugge kode (Visual Studio Code, 2023). Gruppen valgte dette programmet fordi alle medlemmene

hadde gode erfaringer med det fra tidligere semester. Visual Studio Code har også et stort bibliotek av språk med svært god støtte. Noe som gjør debugging og utvikling enklere. Det er verdt å merke at alle medlemmene brukte samme program. Dette ble gjort bevisst med tanke på konsistens i prosjektet.

Wave - Wave er et webbasert grensesnitt som gir tilbakemelding på hvor brukervennlig et nettsted er med tanke på ting som kontrast, om skjermleser er implementert og lignende (Wave, u.å.). Gruppen valgte å ta i bruk dette verktøyet som en del av evalueringen av hvor brukervennlig nettsiden var.

Lucidchart - Lucidchart er et webbasert grensesnitt for å lage og samarbeide på diagrammer (Lucidchart, u.å.). Dette programmet ble brukt for å lage ulike diagrammer for å forklare applikasjonen.

3.2 Sprintene

Dette delkapittelet dekker sprintene for bachelorprosjektet. Selve metodikken og bruken av scrum er beskrevet i delkapittel [2.1 Metodikk](#). Dette kapittelet fokuserer kun på de viktigste hendelsene for hver sprint. Se [vedlegg F](#) for en fullstendig oversikt.

Prosjektperioden startet med en “presprint” hvor hovedfokuset gikk på å klargjøre målene for prosjektet. Disse målene blir beskrevet i delkapittel [1.5 Mål og krav](#) og [3.3.1 Scope](#). I tillegg til å få et klarere bilde over målene til prosjektet gikk resten av tiden bort på kompetanseheving innenfor domenet og diverse administrativt arbeid.

Hovedfokuset for sprint 1 var å fortsette med kompetanseheving innenfor domenet. Gruppen fokuserte på å heve kompetansen innenfor geomatikk, samt å få et overblikk over de ulike teknologiene som ble brukt av den forrige studentgruppen. Disse teknologiene ble beskrevet i kapittel [3.1 Valg av verktøy og teknologier](#).

I sprint 2 var hovedmålet å distribuere applikasjonen via Azure. Dette arbeidet blir beskrevet i delkapittel [5.1 Distribuerings av webløsning](#). I tillegg til å legge løsningen ut på Azure ble det brukt mer tid på kompetanseheving og å forstå hvordan de forskjellige komponentene i systemet hang sammen. Denne prosessen blir beskrevet i delkapittel [4.3 Modellering av backend](#). Samtidig startet gruppen med analysearbeidet for utviklingen av ny frontend løsning. Dette dekkes i delkapittel [4.1 Kartlegging av brukerbehov](#).

For sprint 3 var målet å få applikasjonen funksjonell via Azure. Selv om applikasjonen ble publisert på Azure i sprint 2, var deler av applikasjonen ikke funksjonell. I Sprint 3 implementerte gruppen datakildene WMS opp mot SLD med brukerdefinert styling. Dette arbeidet blir beskrevet i delkapittel [5.3 Implementering av nye datakilder](#). I tillegg til dette gjennomførte gruppen brukertestene for den gamle applikasjonen og begynte arbeidet

med å skissere den nye applikasjonen. Denne arbeidsprosessen er beskrevet i delkapittel [4.1 Kartlegging av brukerbehov](#) og [4.2 Planlegging av UI/UX Design](#).

I sprint 4 utvidet gruppen funksjonaliteten til funksjoner som ble implementert i sprint 3. I denne sprinten ble det implementert muligheten for at brukeren selv kunne velge styling på WMS-kallene. I tillegg ble cookies implementert for å tillate at flere kunne bruke applikasjonen samtidig. Den tekniske prosessen blir beskrevet i delkapittel [5.1 Distribuering av webløsning](#), mens arbeidet med GDPR og personvern dekkes i delpunkt [7.1.7 Personvern og GDPR](#). Arbeidet på frontend fokuserte på å ferdigstille skissene slik at utviklingsarbeidet kunne begynne i neste sprint.

Målet for sprint 5 var å implementere datakildene COG og FGB i applikasjonen. Dette arbeidet beskrives i delkapittel [5.3 Implementering av nye datakilder](#). I tillegg til dette ble det nye designet for applikasjonen implementert. De viktigste punktene her dekkes i delkapittel [5.2 Implementering av nytt brukergrensesnitt](#).

Sprint 6 ble en kortere sprint grunnet påskeferien. Hovedfokuset i denne sprinten var å fikse diverse bugs i koden samt å skrive store deler av denne rapporten.

Målene for sprint 7 var å implementere NGIS, satellittfoto og metadata. Disse oppgavene dekkes i [5.3 Implementering av nye datakilder](#). I tillegg til dette ble det sørget for at personvernerklæringen gruppen skrev fulgte internasjonale standarder for personvern og reglene for GDPR. Det oppsto derimot flere problemer i sprint 6 og gruppen fikk ikke tilgang på NGIS slik som forventet. Denne hindringen beskrives i delpunkt [7.1.6 Tilgang til ressurser](#).

Sprint 8 hadde hovedfokus på å gjøre ferdig de tekniske delene av prosjektet og å sørge for at dokumentasjon og unit testing var implementert riktig. Dette arbeidet beskrives i delkapittel [5.4 Refaktoring, dokumentasjon og testing](#). Det ble også brukt tid på å skrive videre på rapporten.

De resterende sprintene ble brukt på å finpusse og levere inn rapporten. Etter at rapporten ble levert inn falt fokuset på å øve til muntlig eksamen samt å presentere prosjektet i ulike fora som for eksempel Techlunsj for Technology professional hos Tietoenvry og internt hos Kartverket.

3.3 Kartlegging

Dette delkapittelet tar for seg generell kartlegging av prosjektet. Det innebærer risiko, scope og veikart for prosjektet. Her blir det begrunnet hvorfor arbeidet ble utført slik som det ble gjort, samt begrunnelser for prioriteringer internt i prosjektets gang.

3.3.1 Scope

Scope blir definert som “En komponent i prosjektarbeid som hjelper med å finne mål, oppgaver, strategier for arbeidsstyring, og begrensninger” (Adobe, 2022). Det å finne mål og oppgaver ble gjort tidlig i prosjektperioden sammen med oppdragsgiver. Disse oppgavene ble så prioritert sammen med oppdragsgiver slik at det var lettere å vite hvor gruppen skulle begynne arbeidet. Listen med mål ble først presentert i delkapittel [1.5 Mål og krav](#) og ble seende slik ut.

Målsetting etter første møte med oppdragsgiver:

- Distribudere applikasjonen på nett via Microsoft Azure
- Forbedre brukervennligheten for applikasjonen
- Legge til nye datakilder for WMS, COG, FGB, og NGIS
- Refaktorere eksisterende kode til noe som er mer forståelig og lettere å videreutvikle

Disse målsettingene definerer det ønskede scopet for prosjektet, men oppdragsgiveren var også veldig åpen om at så lenge gruppen var fornøyd med det endelige resultatet var han også fornøyd. Gruppen fikk i tillegg stadig innspill om andre ting som kunne implementeres i applikasjonen. Et eksempel på dette er implementeringen av Copernicus satellittfoto som var et ønske gruppen fikk midten av prosjektperioden. Dette arbeidet blir beskrevet videre i delkapittel [5.3 Implementering av nye datakilder](#)

En annen oppgave som heller ikke var en del av det originale scopet var håndteringen av GDPR og personvern. Selv om dette formelt sett faller inn under oppgaven “Distribudere applikasjonen på nett via Microsoft Azure” var det ikke et krav fra oppdragsgiver og var heller noe gruppen valgte å ta på seg for å få erfaring med domenet og å gjøre ting ifølge regelverket.

3.3.2 Roadmap

Gruppen utviklet under planleggingsdagen for sprint 1 et roadmap for hele prosjektet. Et roadmap er en visualisering av hvordan produktet kommer til å utvikle seg over tid (Davies, u.å.). Roadmap er spesielt viktig i et agilt prosjekt ettersom det sørger for at alle har en felles forståelse for hvordan oppgaver har blitt prioritert og når de skal gjennomføres (Davies, u.å.).

Roadmappen gruppen utviklet viser en oversikt over hovedaktivitetene i prosjektet, når de skal påbegynner og når de er estimert ferdig. Nedenfor er et roadmap som viser hvordan gruppen så for seg at prosjektet skulle bli gjennomført ved starten av bachelorprosjektet. Det ble utformet med utgangspunkt i det scopet som ble avtalt med produkteier fra Kartverket ved prosjektstart.

Veikartet per 15.01 (Sprint 1)	Presprint	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Eksamen
Kursstart												
Forståelse og kunnskap												
Webløsning												
Bruke SLD mot WMS i stedet for PostGIS												
Se på det å bruke FGB og COG												
Utvide med andre typer data												
Frontend												
Brukertesting												
Prototyping												
Overlevering og Innlevering												
Forberede muntlig eksamen												

Figur 1: Roadmap per 15.01

I starten av prosjektet var det beregnet mye tid til kompetanseheving, rapportskrivning og prosjektadministrasjon. Etter hvert som gruppen fikk hevet kompetanse om domenet, delte medlemmene seg i to grupper. En gruppe var ansvarlig for backend mens den andre var ansvarlig for frontend. På denne måten ville gruppen unngå at for mange jobbet med det samme, som kan føre til redusert effektivitet.

Det viste seg at gruppen ble ferdig med de ulike målene raskere enn forventet. Ved prosjektstart ble det estimert at målene kom til å ta to sprinter (4 uker) hver. UX/UI delen av prosjektet var noe gruppen trodde de skulle holde på med under hele prosjektet. Gruppen ferdigutviklet derimot hvert delmål på 1 sprint hver med ett unntak. Det ble også gjort omprioriteringer underveis. Nedenfor vises endelig roadmap som ble oppdatert 25.03 ved oppstarten av sprint 6. Alle veikartene gruppen lagde kan bli funnet i [vedlegg E](#).

Veikartet per 25.03 (sprint 6)	Presprint	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Eksamen
Kursstart												
Forståelse og kunnskap												
Webløsning												
Bruke SLD mot WMS i stedet for PostGIS												
Utvide med andre typer data og klasser												
Se på det å bruke FGB og COG												
Dokumentasjon & diverse												
Implementere NGIS + Diverse												
Frontend												
Brukertesting												
Prototyping												
Overlevering og Innlevering												
Forberede muntlig eksamen												

Figur 2: Roadmap per 25.03

3.3.3 Risiko og risikomatrise

Risiko er en potensiell hendelse som vil ha negativ påvirkning på prosjektets fremgang (Aven, 2023). Når det skal gjennomføres et så stort bachelorprosjekt med fem ferske utviklere er det naturligvis en del risiko involvert. Derfor er det viktig å identifisere disse og ta nødvendige steg for å nøytralisere dem slik at de ikke påvirker prosjektets fremgang eller resultat. For å få til dette ble det utviklet en risikomatrise. Dette er et diagram som oppsummerer risikoer som kan inntreffe og hvilke tiltak som kan forhindre risikoene samt hvilke konsekvenser en risiko kan ha (Aven, 2022).

Gruppen utviklet en slik risikomatrise og identifiserte flere forskjellige potensielle risikoer med prosjektet. Risikomatriksen var basert på en mal som gruppen fikk av Tietoevry. Her ble det listet mulige risikoer og effekter som disse kunne ha på prosjektet, samt mulige tiltak for å forhindre eller håndtere disse. Sannsynligheten for hver risiko ble målt på en skala fra 1-4, hvor 4 var meget høy og 1 var meget lav. Hva de forskjellige verdiene betyr blir fastslått i det fullstendige dokumentet for matriksen. Se [Vedlegg D](#).

Risikomatriksen ble brukt aktivt og revidert under hvert sprintplanleggingsmøte. Dette ble gjort for å holde kontroll på hvilke risikoer som var mest aktuelle, og hvordan vi kunne møte risikoene hvis de inntraff. Dette kapittelet beskriver noen av de viktigste risikoene gruppen måtte håndtere og hvordan de arbeidet med risikomatriksen kontinuerlig for å senke mulighetene for at en uønsket hendelse skulle inntreffe.

Overestimert prosjektomfang Ettersom gruppen ikke har tidlig erfaring med geografiske informasjonssystemer og må lære dette fra starten, er det fort gjort	3	3	9	Bedriftene og produkteier har vært tydelig på at vi skal kunne ta den siste avgjørelsen for prosjektomfanget. Ettersom det overordnede målet	3	1	3	Sett i gang en diskusjon med produkteier med en gang for å gjøre endringer så tidlig som mulig, oppdater prosjektplanen, <u>identifiser</u> årsaken til feilestimering og hvilke tiltak som må tas for å ta mer sikre beslutninger i fremtidige <u>sprint</u> .
--	---	---	---	--	---	---	---	---

Figur 3: Uttak fra risikomatrise

Den største risikoen gruppen møtte i starten av prosjektperioden var at ett av gruppemedlemmene ikke hadde mulighet til å bidra til arbeidet fysisk. I en situasjon som dette er det flere risikoer som følger med. Blant annet mangel på kommunikasjon eller at man føler seg utelatt av gruppen. For å forhindre at denne risikoen inntraff valgte gruppen å innføre flere forebyggende tiltak. For eksempel ble alle møter tatt over enten Discord eller Teams og gruppen prøvde etter beste evne å inkludere det siste medlemmet. Over tid viste det seg derimot at det gruppen trodde var en stor risiko i starten ikke var særlig relevant da flere av de andre gruppemedlemmene også jobbet hjemmefra og store deler av tiden og kommunikasjonen internt i gruppen var svært god.

En annen stor risiko som ble mer relevant for gruppen utover i prosjektperioden var mangelen på tilganger fra oppdragsgiver. Selve disse problemene blir beskrevet mer i detalj i delpunkt [7.1.6 Tilgang til ressurser](#) og delkapittel [5.3 Implementering av nye datakilder](#). Gruppen anså risikoen for at tilganger uteble som liten i starten av prosjektperioden fordi Kartverket hadde supplert tidligere studentgrupper med nødvendige tilganger. Etterhvert som prosjektperioden pågikk uteble derimot noen av tilgangene som var nødvendige for å få NGIS delen av applikasjonen til å fungere. Denne risikoen ble derfor større. Dette viser at selv om man planlegger for risikoer og gjennomfører de forebyggende tiltakene man har etablert, er risiko fortsatt et faktum som man ikke alltid kan kontrollere. Det vil alltid være en sannsynlighet for at uønskede hendelser inntreffer.

3.3.4 Kapasitetsplan

Gruppen opprettet også en kapasitetsplan etter råd fra veileder på Tietoevry. Kapasitetsplanen inneholder en oversikt over tilgjengelige timer for alle i teamet, og viktige møter holdt denne sprinten. Alle gruppemedlemmene sto selv ansvarlige for å føre potensielle tap av timer. Dette førte ikke til store problemer da alle i gruppen stoler på hverandre, med at arbeidet blir ferdigstilt.

Sprint 7															
Teamets tilgjengelighet april															
Allokering	Navn	Rolle	Man	Tirs	Ons	Tors	Fre	Lør/Søn	Man	Tirs	Ons	Tors	Fre	Lør/Søn	Totalt antall timer
			8	9	10	11	12	13/14	15	16	17	18	19	20/21	
100%	Victor	Utvikler	6,5	6,5	0	6,5	5,5		0	6,5	6,5	6,5	5,5		50
100%	Gustav	Utvikler	6,5	6,5	0	6,5	5,5		6,5	6,5	0	6,5	5,5		50
100%	Larisa	Utvikler (UX/UI)	6,5	6,5	0	6,5	5,5		6,5	6,5	0	6,5	5,5		50
100%	Cassandra	Utvikler	6,5	6,5	0	6,5	5,5		6,5	6,5	0	6,5	5,5		50
100%	Tørres	Prosjektleder	4	7,5	0	6,5	4		7,5	7,5	0	6,5	6,5		50
Totalt antall timer per dag			30	33,5	0	32,5	26		27	33,5	6,5	32,5	28,5		
Totalt antall utviklingstimer tilgjengelig for denne sprinten (ekskludert planlegging)															213,5
Planlegging															30
Totalt (inkludert sprintplanlegging)															243,5
Merk: Tørres er på Studierådsmøte på mandag 08.04, Demomøte torsdag 11. april															
Notat:															

Figur 4: Kapasitetsplan

Figuren over viser til hvordan en typisk kapasitetsplan så ut. Her kan man også se at gruppen har gjort opp for tidligere fravær. Dette har da blitt gjort grunnet målet om å jobbe minimum 25 timer uken for hvert medlem med prosjektet. Det er også lagt til litt mer informasjon om hva som kommer til å foregå i ukene under planen for å skape mer oversikt. Hele kapasitetsplanen er vedlagt i [vedlegg B](#).

4. Designprosessen

Dette kapitlet dekker de viktigste aspektene av designprosessen. Designprosessen inneholder analyse og evaluering av design for både frontend og backend. Kapitlet beskriver hvordan ulike metoder og verktøy bidro til konseptualiseringen av et nytt design.

4.1 Kartlegging av brukerbehov

Dette delkapitlet dekker gruppens kartlegging av brukerbehov gjennom automatisert og manuell testing av den gamle applikasjonen. Kombinasjonen av disse testmetodene ga gruppen en helhetlig forståelse for hva som burde implementeres i gruppens nye versjon.

Etter gjennomgang av applikasjonen utviklet av den forrige studentgruppen våren 2023 erfarte gruppen en rekke mangler som negativt påvirket brukeropplevelsen. Det ble blant annet ikke gitt noen forklaring til formålet med applikasjonen, forklaringer til ulike funksjoner i applikasjonen, anbefalt brukerinput eller tilbakemeldinger fra systemet. Gruppen kom fram til at applikasjonen ikke var intuitiv å forstå eller bruke. Dermed la gruppen til forbedring av UX og UI designet som et av hovedmålene i prosjektet. For å

fastslå hvordan den skulle forbedres måtte applikasjonen testes ut på potensielle brukere (Sandnes, 2022, s. 315).

4.1.1 Brukertesting

De ansatte ved fylkeskontoret for Kartverket i Kristiansand hadde profilen som mest passet målgruppen for applikasjonen ettersom de hadde svært varierende kunnskap innen geodata og kunstig intelligens, og flere av dem var potensielle brukere av applikasjonen. Ettersom målgruppen var ganske spesifikk ble kooperativ evaluering valgt som metode for brukertesting. Gjennom en kooperativ evaluering samler man inn mest mulig data fra hver enkelt test ved å oppfordre testbrukeren til å tenke høyt og fasilitere diskusjon mellom testobjekt og testleder for mer detaljerte resultater (Benyon, 2019, s. 250). Grunnet dette ble det ikke behov for å gjennomføre mange brukertester. Testene gikk ut på å markere det ønskede treningsområdet på kartet og deretter definere fordelingen av treningsdata og fasitdata for AI modellen.

Gruppen valgte å gjennomføre to runder med brukertesting. I første omgang testet gruppen applikasjonen som ble utviklet våren 2023 av den forrige studentgruppen. For denne utgaven ble det gjennomført tre brukertester. Senere i prosjektet ble det gjennomført en ny runde med brukertesting for applikasjonen som gruppen utviklet våren 2024. Og det igjen ble gjennomført ytterligere tre brukertester. Testresultatene for gruppens forbedrede versjon blir presentert senere i delpunkt [5.2.2 Resultat fra brukertesting av ny applikasjon](#). Formålet med brukertesting av den gamle applikasjonen var å kartlegge brukernes utfordringer, ønsker og behov for applikasjonen. Formålet med brukertesting av gruppens reviderte applikasjon var for å sjekke om behovene hadde blitt tilfredsstilt.

4.1.2 Gjennomføring av brukertestene

For å gjenspeile sluttbrukerne av applikasjonen valgte gruppen bevisst ut potensielle brukere med varierende erfaring med geodata og kunstig intelligens. For å redusere distraksjoner ble testene utført i private grupperom hos Kartverket. Testpersonene ble hentet inn enkeltvis for å gi hver person mest mulig oppmerksomhet, sikre en grundig observasjon av alle reaksjonene og i tillegg unngå gruppepåvirkning. Under testen deltok de to gruppemedlemmene ansvarlig for frontend. En hadde rollen som testleder, og var ansvarlig for å lede brukeren gjennom brukertesten og hjelpe til ved eventuelle misforståelser eller spørsmål. Den andre hadde rollen som observatør, og var ansvarlig for å notere ned reaksjoner og tilbakemeldinger fra testobjektet. De potensielle brukerne ble gitt et papir med en bestemt brukerhistorie, stegene som skulle tas for å gjennomføre brukerhistorien og tilslutt et sett med spørsmål for å fastslå inntrykk av applikasjonen og hvilke endringer de ville se i redesignet. Malen for brukertest og eksempel på utført brukertest kan finnes i [vedlegg J](#) og [vedlegg K](#).

4.1.3 Resultater fra brukertester runde 1

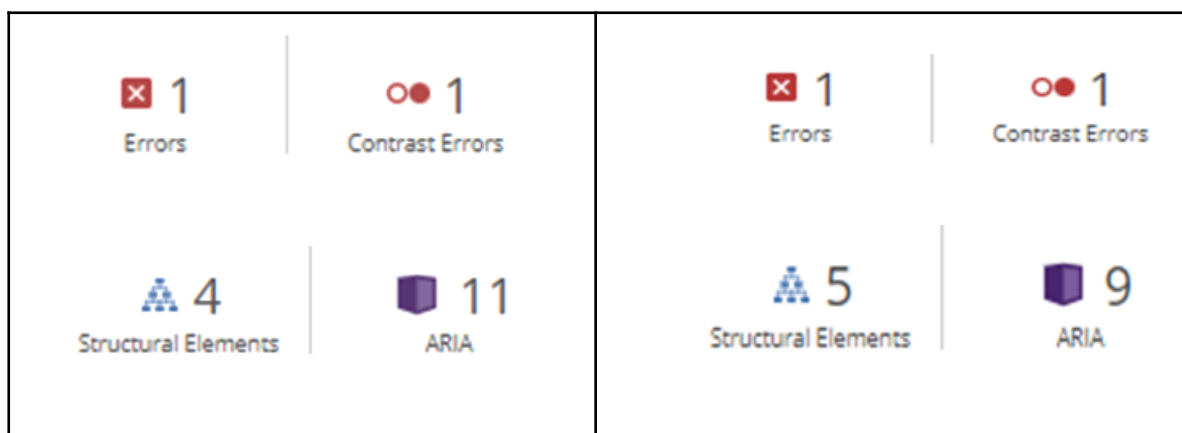
Brukertesting av den gamle applikasjonen viste at mange slet med å forstå hva som var formålet med applikasjonen og hvordan den skulle brukes. Testpersoner med lang erfaring innen GIS syntes det var enkelt å navigere seg rundt i applikasjonen og markere områder på kartet ettersom de hadde tidligere kjennskap til slike applikasjoner. I mellomtiden hadde de likevel problemer med å forstå betydningen av flere tekstfelt og hvilke verdier som skulle plottes inn. Det var også uklart hvordan disse verdiene i det hele tatt ville påvirke sluttresultatet. I tillegg ble det foreslått å implementere en søkefunksjon da det var tidkrevende å lete seg fram til ønsket område på kartet. Inntrykket av applikasjonen var nøytral, men det var tydelig for disse testbrukerne at den hadde stort forbedringspotensial.

Brukere med mindre erfaring innen GIS hadde flere formeningar. Stilen ble beskrevet som utdatert og lite profesjonell. I tillegg til å ikke forstå begreper og verdier var det også forvirring rundt applikasjonens formål. Det var vanskelig for testpersonene å navigere seg på kartet, og det ble brukt mye tid og energi på å prøve å markere et område. Til slutt ga en av personene opp. Inntrykket til disse testpersonene var generelt blandet og de så ikke verdien i å benytte applikasjonen i fremtiden.

4.2.1 Tilgjengelighetstesting mot WCAG

WCAG står for “Web Accessibility Guidelines” og er et sett med internasjonale retningslinjer utarbeidet av Web Accessibility Initiative (WAI). Retningslinjene utgjør en felles standard som skal tilgjengeliggjøre webinnhold for mennesker med funksjonshemmelser, som for eksempel synsnedsettelse eller lesevansker (Web Accessibility Initiative, 2005). For dette har gruppen brukt nettsiden Wave som er et verktøy for evaluering av webinnhold mot WCAG (Wave, u.å.). Figur 5 og 6 viser to sider som ble vurdert med en WCAG test. Videre kommer en forklaring av resultatene og hva de betyr for tilgjengeligheten av systemet.

4.2.2 Resultater fra automatisert tilgjengelighetstesting



Figur 5: Siden for å skrive inn koordinater

Figur 6: Siden for Innstillinger

ARIA - ARIA (Accessible Rich Internet Application Suite) er spesielle attributter som legges til i HTML-strukturen av en nettside for å gjøre den forståelig for hjelpeteknologier som skjermlesere. Funksjonshemmede brukere vil kunne forstå og navigere nettsiden ved at skjermleseren tolker og formidler sidens innhold til brukeren. Applikasjonen inneholder manglende og feil bruk av ARIA-attributter. Dette gjør det vanskelig for slike hjelpeteknologier å tolke innholdet riktig, noe som begrenser brukerens mulighet til å bruke applikasjonen effektivt (Nurthern & Cooper, 2024).

Structural elements - Structural elements refererer til elementene i HTML-markup som beskriver semantiske strukturen til applikasjonen. Denne semantikken brukes for å organisere all innhold og funksjonalitet på en måte som beskriver dens formål og rolle på nettsiden. Den valgte semantikken i applikasjonen ga et uklart eller feil bilde av sidens egentlige innhold. Dette gjør det vanskelig for blant annet nettlesere, søkemotorer og skjermlesere å forstå innholdet, og reduserer dermed også brukbarheten og synligheten av nettsiden på nett. (WAVE, u.å.)

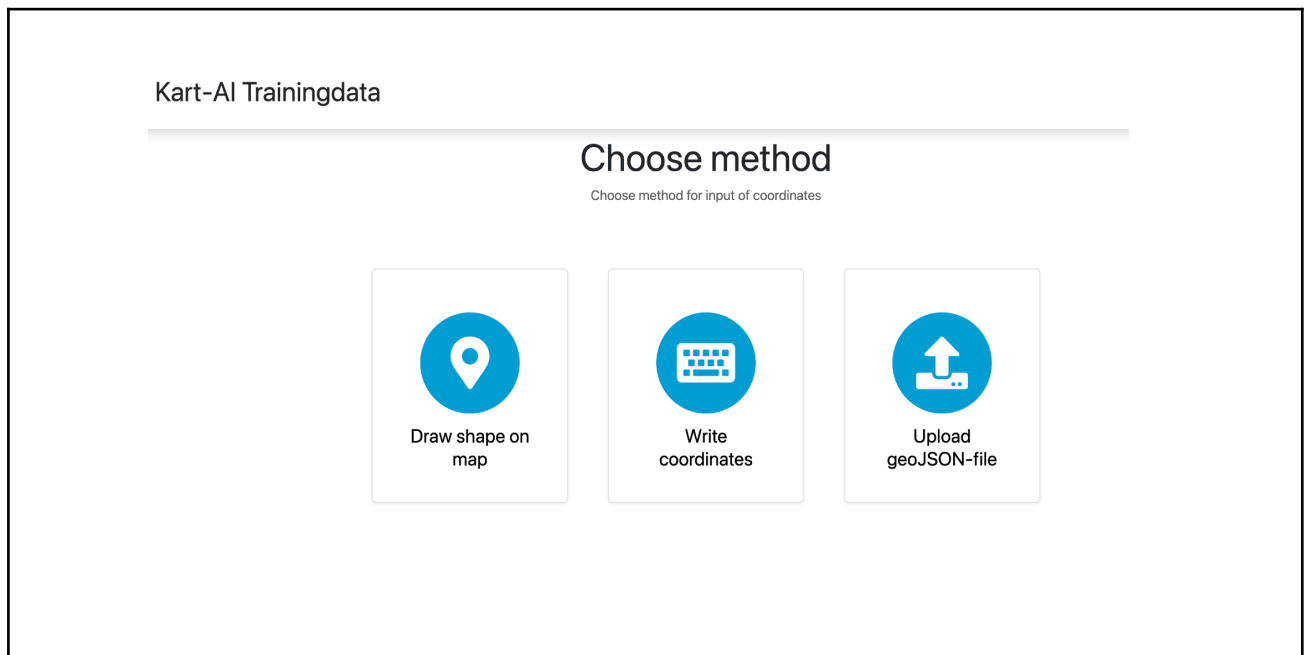
Errors - Errors viser til direkte brudd på retningslinjene i WCAG standarden (WAVE, u.å.). For denne applikasjonen ble det funnet errors som lav fargekontrast, mangel på etiketter for tekstfelt og mangel på alternativ tekst for knapper, bilder og ikoner. Dette påvirker igjen skjermlesere sin evne til å tolke applikasjonen, og skaper også problemer dersom bilder og ikoner ikke vil laste, og det ikke finnes alternative visuelle ledetråder for å forstå innholdet. Tilslutt kan dårlig fargekontrast gjøre det vanskelig å lese informasjonen som blir oppgitt.

Resultatene fra WAVE test viste at hovedutfordringene for tilgjengelighet i applikasjonen lå i systemets indre struktur, som forhindrer skjermlesere i å forstå applikasjonen. Dette er viktig fordi brukere med blant annet synsnedsettelse, bevegelseshemming, læringsvansker, lesevansker og andre kognitive funksjonsnedsettelse bruker skjermlesere for å navigere og utnytte applikasjonen. Utover dette var det ingen flere tilgjengelighetsproblemer, og applikasjonen samsvarte i stor grad med retningslinjene til WCAG. Dette kan forklares med at applikasjonen hadde en svært minimalistisk utforming med få funksjoner. Noe som førte til et begrenset testgrunnlag. Gruppen fikset de ovennevnte problemene gjennom bugfiksing hvor det ble lagt til flere semantiske HTML-tagger med ARIA-attributter, samt alternativ tekst for ikoner og kart.

4.3.1 Resultater fra heuristisk evaluering

Som nevnt i delkapittel [2.2 Kvalitetssikring](#) ble den gamle applikasjonen vurdert av gruppen opp mot Benyons designprinsipper. Følgende delkapittel beskriver kort hvilke prinsipper som applikasjonen bryter sterkt med og eksempler på hvordan.

Visibility - Visibility handler om at informasjon og funksjonalitet skal være synlig slik at brukeren vet hva som foregår i applikasjonen, og hva systemet forventer av dem (Benyon, 2019, 117). Applikasjonen manglet blant annet forklaring på alle begreper og metoder. I tillegg manglet det instruks for hvordan systemet brukes. Brukeren fikk ikke oppgitt hvilke områder som var frigitt og kunne dermed ikke være sikker på hvorvidt området de markerte ville gi dem reell treningsdata.



Figur 7: Gammel forside

Recovery - Recovery prinsippet refererer til å gi brukeren mulighet til å rette opp i sine feil og generelt reversere sine handlinger (Benyon, 2019, 117). Dersom brukeren markerte feil område på kartet var det ikke mulig å slette eller endre på markeringen. Det var heller ikke mulig for brukeren å gå tilbake til tidligere sider dersom de ville endre sine inputverdier.

Constraints - Dette prinsippet handler om å begrense brukerens handlingsfrihet for å sikre at de opprettholder systemkravene og unngår handlinger som kan medføre betydelige feil. (Benyon, 2019, 117). I den gamle applikasjonen var det ingen funksjonalitet som forhindret brukere fra å skrive inn tilsynelatende ugyldig innhold inn i tekstfeltene. Det var heller ingenting som stoppet brukeren fra å gå gjennom systemet uten å ha oppgitt noen informasjon.

Feedback - Feedback handler om å gi tilbakemeldinger til brukeren slik at de er klar over at handlingene deres er registret av systemet, og forstår hvordan systemet tolker disse handlingene (Benyon, 2019, 117). Brukeren fikk ingen tilbakemelding dersom koordinater eller annet input manglet eller var feil. Etter å ha trykket på order-knappen fikk brukeren ingen tilbakemelding om hvorvidt de vil motta dataene, hvor de vil bli tilsendt eller når.

4.2 Planlegging av UX/UI Design

Dette delkapittelet dekker designprosessen for brukergrensesnittet til applikasjonen. Gruppen har tidligere diskutert hvilke metoder de brukte for å samle inn tilbakemeldinger rundt om applikasjonen og vil i denne delen av rapporten gå over utviklingen av et nytt brukergrensesnitt. Fra skisser til en prototype.

Etter testperioden hadde gruppen en lang liste med oppgaver som skulle gjennomføres, og det var usikkert om gruppen ville få tid til å implementere alle endringene. For å sikre at de mest etterspurte endringene ble del av den nye applikasjonen ble det gjennomført en MoSCoW-analyse av alle frontend oppgaver. En MoSCoW analyse er et hjelpemiddel for å lettere kunne prioritere og sortere oppgaver i et prosjekt (Productplan, u.å.). Gruppen konkluderte at gjennomføring av oppgavene i kategorier “must have” og “should have” skulle være minimumskravet for å oppfylle det siste prosjektmålet “Forbedre brukervennligheten for applikasjonen”. MosCoW analysen kan sees i [vedlegg T](#).

4.2.1 Skissering og lo-fi prototyping

På bakgrunn av innsamlet data utarbeidet gruppen enkle skisser av den nye applikasjonen. Hensikten med skisser er å teste ut ideer for design, og bli enig om hvordan funksjonene skal struktureres på sidene (Rojas, 2023). Først ble det utarbeidet individuelle skisser og deretter ble det holdt gruppemøte for sammenligning og drøfting av hva som var bra med de forskjellige, og hvordan disse kunne kombineres. Basert på dette utviklet gruppen en lo-fi prototype av applikasjonen i Figma. Fokuset med lo-fi prototypen var å visualisere hvordan funksjoner og informasjon skulle plasseres på sidene for en intuitiv brukeropplevelse. Lo-fi skisser kan ses i [vedlegg N](#).

4.2.2 High-fi prototyping

Basert på skissene som ble laget, utviklet gruppen en high fidelity prototype for det nye designet i Figma. En high fidelity prototype er en prototype som simulerer det ferdige produktet gjennom visuelt design og innhold. Denne typen prototype er ofte veldig nær det endelige produktet (Figma, u.å.). Gruppen fokuserte først på å fikse utseende til prototypen og legge til nødvendig informasjon uten å gjøre den funksjonell. Denne versjonen ble så vist frem til produkteier under et møte for å få tilbakemeldinger og godkjenning, og det ble gjort et par endringer basert på tilbakemeldingene gruppen fikk. Denne fremgangsmåten sikret at gruppen ikke brukte tid på legge til funksjonalitet for et design som produkteier ikke var fornøyd med. Etter at designet var på plass ble det lagt et primært mål om å få prototypen i kjørende tilstand på Figma med nødvendig informasjon, layout, innhold og designelementer.

Selv om gruppen fikk god innsikt i hvordan applikasjonen skulle utformes var dette fortsatt et førsteutkast og designet til applikasjonen endret seg underveis i utviklingen. Grunnen til

dette er at gruppen måtte tilføye funksjonalitet for nye behov som dukket opp under utviklingsprosessen. Det ferdigstilte utkastet av prototypen kan sees i [vedlegg O](#).

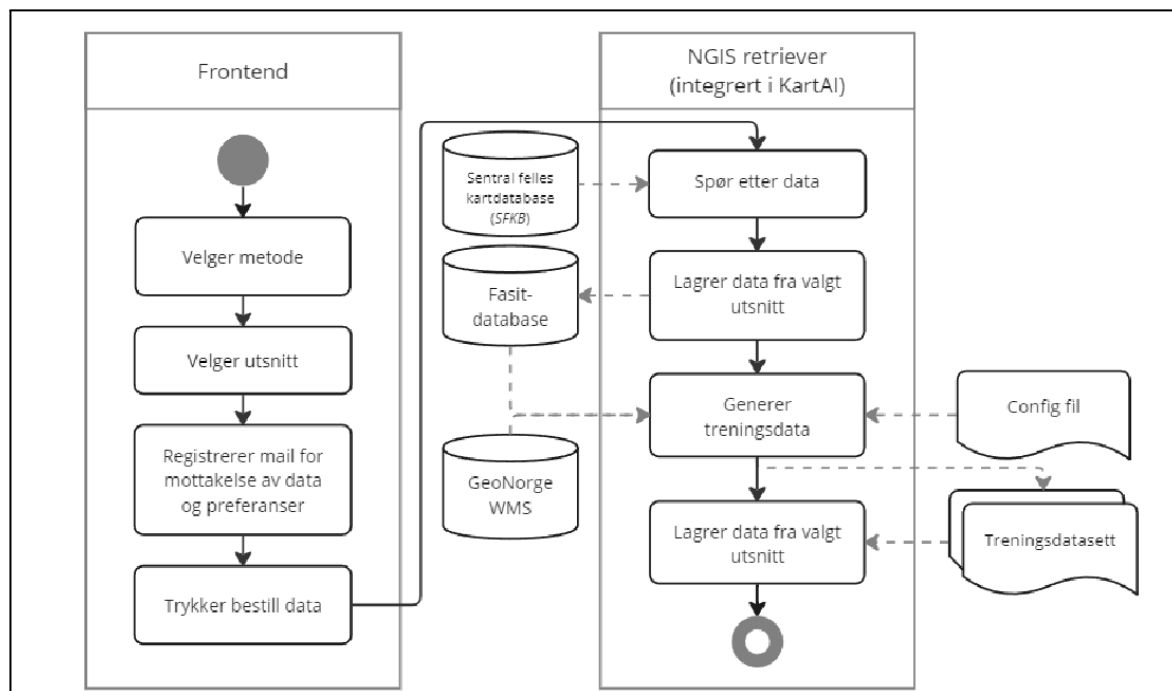
4.3 Modellering av backend

Dette kapitlet tar for seg modelleringsprosessen for backend av applikasjonen. Modellering handler om å skape en visuell fremstilling av de ulike komponentene som finnes i et system, hvordan disse henger sammen og hvordan data utveksles mellom dem (Visual Paradigm, u.å.). Dette var en avgjørende del av forberedelsesfasen gitt gruppens manglende forkunnskap om GIS og avansert systemutvikling. Ved å simulere alle prosessene i systemet kunne gruppen identifisere problemer og risikoer på forkant av implementeringsfasen for å spare både tid og ressurser. Modellering av systemet var et viktig hjelpemiddel for å forstå rollen til, og sammenhengen mellom ulike teknologier i applikasjonen, samt gruppens oppdrag i videreutviklingen. I tillegg til at selve modellene hadde en stor verdi for gruppen, var det også et viktig verktøy for kommunikasjon av systemkrav og fremdrift i utviklingen for de mange interessentene i prosjektet. Modellering ble også et viktig hjelpemiddel i løpet av prosjektperioden for å kunne kommunisere hvilke oppgaver som måtte gjøres og hvordan de forskjellige oppgavene hang sammen. Det å ha en modell av systemet som kontinuerlig utviklet seg i takt med at systemet ble mer komplekst, gjorde at vi enkelt kunne kommunisere om hvordan de ulike komponentene interagerer. Modellene som blir vist i dette kapitlet er ikke alle modellene som ble utviklet, men heller de ferdige modellene på ulike stadier i utviklingsprosessen. Gruppen ønsker å vise frem alle disse modellene for å fremheve at prosjektet har blitt større over tid og for å presisere at modellene har utviklet seg i takt med gruppens og prosjektets behov.

4.3.1 Original dataflyt

Ettersom gruppens prosjekt gikk ut på å videreutvikle både backend og frontend delen betydelig var det svært viktig at alle i gruppen først fikk en grundig forståelse for den foregående applikasjonen før det ble gjort betydelige endringer. Som utgangspunkt valgte gruppen derfor å ta i bruk en modell av dataflyten som ble utviklet av den forrige studentgruppen for bygge en solid forståelse for daværende løsning, og hvordan produkt eierens nåværende krav ville passe inn i modellen. Modellen er en hybrid mellom et aktivitet og et sekvensdiagram (Bredesen et al., 2023, s.27). Et aktivitetsdiagram beskriver flyten mellom de ulike aktivitetene eller stegene i en applikasjon (Visual Paradigm, u.å.) mens et sekvensdiagram beskriver hvordan de ulike objektene i et system interagerer (Bell, 2023). Denne måten å føre diagrammer på viste å være meget oversiktlig, og

gruppen valgte dermed å videreføre dette i utviklingen av egne modeller for dataflyt.

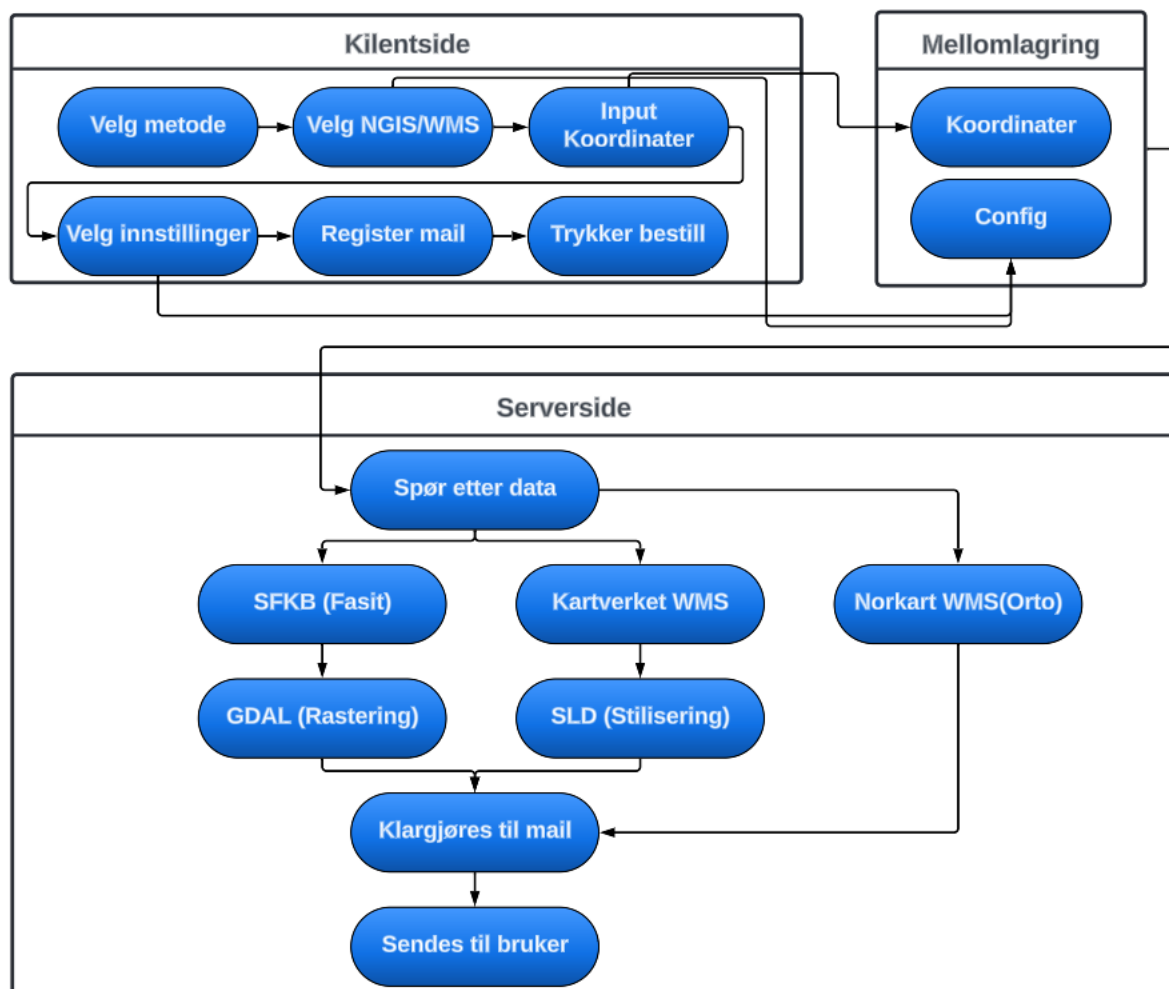


Figur 8: Endelig modell av dataflyt til løsningen (Bredesen et al., 2023).

4.3.2 Tiltentkt dataflyt

Det å ha en god forståelse for hvordan de forskjellige komponentene i systemet henger sammen, ble svært viktig i startfasen til prosjektet. Gruppen begynte derfor tidlig i prosjektperioden å analysere diagrammet fra figur 8. og hvordan dette diagrammet representerte de ulike delene av koden som var utviklet. Dette ble gjort i fellesskap slik at alle gruppemedlemmene hadde en solid forståelse for hvordan systemet fungerte på daværende tidspunkt og hvordan de planlagte endringene ville påvirke dataflyten i systemet.

Etter at alle hadde en felles forståelse for systemet på daværende tidspunkt ble det modellert et nytt diagram slik gruppen kunne se for seg at systemet ville se ut etter at prosjektperioden var ferdig.



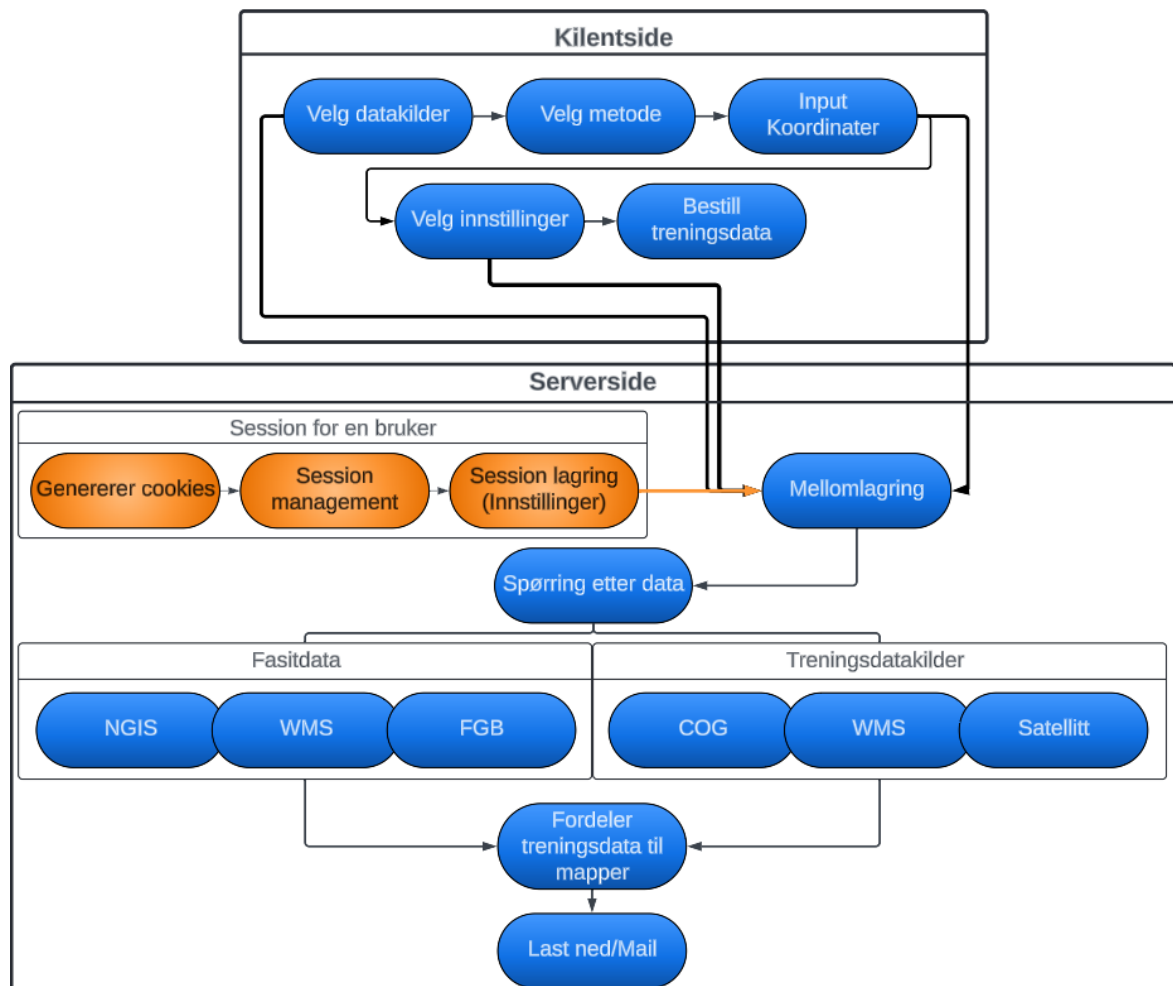
Figur 9: Første dataflyt

På daværende tidspunkt var gruppen usikker på hvor mye de ville få gjort i løpet av prosjektarbeidet og gjorde derfor ikke veldig store endringer på modellen. De største endringene fra den tidligere modellen var at brukeren skulle ha muligheten til å velge mellom bruk av SFKB (NGIS) og WMS som datakilde for fasit data, siden dette var ett av akseptansekriteriene for applikasjonen. I praksis betyr dette at både backend og frontend delen av applikasjonen ble utvidet for å kunne håndtere ulike valg som brukeren kunne ta. Deretter måtte applikasjonen prosessere og hente ned informasjon fra ulike datakilder basert på valget til brukeren.

4.3.3 Endelig dataflyt og systemarkitektur

Den endelige dataflyten til prosjektet kan sees i figur 10. De viktigste forandringene fra den første dataflyten er at det er utvidet med enda flere måter å hente ut både fasitdata og ortofoto på. Dette ble gjort for å kunne gi brukeren flere valg i hvilken informasjon de ville hente ut og hvordan. Den endelige dataflyten for applikasjonen har mange flere valgmuligheter når det kommer til valg av datakilder. En annen viktig endring er at applikasjonen kan brukes av flere personer samtidig, takket være informasjonskapsler.

Dette dekkes også mer i delkapittel [5.1 Distribuering av webbløsning](#). Denne modellen ble mer kompleks enn den gruppen hadde laget i starten av prosjektperioden som følge av at det ble tid til flere oppgaver og dermed et mer komplekst system. Denne modellen ble utviklet mot slutten av prosjektperioden for å sikre at fremtidige interessenter i prosjektet hadde en datamodell som representerte systemet de tok over.



Figur 10: Endelig dataflyt

Ettersom modellen kan virke kompleks har gruppen utarbeidet en stegvis forklaring for å tydelig kommunisere dataflyten i systemet til interessenter med mindre erfaring innen GIS:

- På klientsiden av applikasjonen velger brukeren hvordan koordinatene skal legges inn, hvilke datakilder som skal brukes for å hente data og hvilke parametere denne dataen skal inneholde. Brukeren har også mulighet til å laste opp en fil med predefinert metadata dersom de ønsker å gjenskape resultater fra et tidligere prosjekt.
- Serveren oppretter individuelle mapper for hver bruker der all brukerininput blir lagret og prosessert.
- Denne inputen blir mellomlagret i filer på serversiden frem til brukeren trykker på bestill knappen. Deretter hentes informasjonen opp av systemet og blir prosessert.

- Basert på brukerens valg henter applikasjonen ned informasjon fra de ulike datakildene og fordeler den i henhold til hvordan oppdragsgiver ville ha treningsgrunnlaget organisert
- Etter at informasjonen har blitt hentet ned har brukeren mulighet til å laste den ned eller få den tilsendt på mail før den så blir slettet fra serveren i samsvar med betingelsene fastsatt i applikasjonens cookie policy. Dette sikrer at applikasjonen ikke bruker mer lagringsplass enn strengt tatt nødvendig.

5. Utviklingsprosessen

Dette kapitlet dekker de viktigste punktene for utviklingsprosessen for både frontend og backend. Utviklingsprosessen blir forklart i forhold til målsettingene gruppen etablerte i delkapittel [1.5 Mål og Krav](#). Målsettingene var basert på krav fra oppdragsgiver og prosjektet ble strukturert etter disse målene.

5.1 Distribuering av webløsning

Den første oppgaven gruppen måtte løse var å distribuere den daværende applikasjonen på Azure. Siden applikasjonen på det tidspunktet kun hadde kjørt på lokal maskin var det usikkert om det måtte gjøres store endringer for å kjøre applikasjonen på nett.

For å løse denne oppgaven valgte gruppen først å analysere ressursbruken til applikasjonen. Det viste seg at selve applikasjonen var relativt enkel, men prosessen med å generere treningsdata gjennom KartAI krevde betydelige mengder ressurser. Dette fordi på daværende tidspunkt brukte applikasjonen KartAI kodebasen til å generere treningsdata. KartAI kodebasen har flere avanserte funksjoner for maskinlæring som strengt tatt ikke var nødvendig for gruppens formål.

Det ble derfor tatt valget om å koble prosjektet fra KartAI kodebasen som ble brukt av den forrige studentgruppen. Dette valget ble gjort for å spare på ressurser og for å gjøre applikasjonen mindre kompleks. Mesteparten av funksjonaliteten i KartAI koden var ikke nødvendig eller kunne gjenskapes på mindre ressurskrevende måter. For å erstatte denne koden skrev gruppen egne funksjoner som gjorde mange av de samme oppgavene med redusert ressursbruk og kompleksitet. Dette løste problemene som var relevante for oppgaven, og man slapp å ha all funksjonalitet for maskinlæring og lignende. Det å fjerne KartAI kodebasen fra prosjektet hadde også noen ulemper. Eksempelvis at gruppen selv måtte skrive kode for å konvertere og rasterisere data fra datakildene slik at de kunne brukes til maskinlæring. Dette var en oppgave som kunne vært unngått hvis man hadde beholdt KartAI kodebasen. Fordelene ved å fjerne den var derimot større enn ulempene.

Gruppen fikk tildelt et budsjett på 800 kroner i måneden fra oppdragsgiver for å dekke eventuelle kostnader ved distribuering av applikasjonen. Gruppen gjorde så en

gjennomgang av de forskjellige løsningene som tilbys gjennom Azure og fant en som passet gruppens og applikasjonens behov. Applikasjonen ble distribuert til Azure gjennom en docker container. Fordelen med en Docker container er at den støtter flere typer rollebasert autentisering (Microsoft, u.å). Noe som oppdragsgiver ønsket at fremtidige studentgrupper kunne implementere. En annen fordel med denne løsningen er at en docker container pakker sammen alle bibliotekene applikasjonen er avhengig av og dermed har mindre sannsynlighet for å møte på kompatibilitetsproblemer senere.

Det at applikasjonen nå var distribuert på Azure førte også til at gruppen måtte tenke på hvordan applikasjonen skulle håndtere flere brukere av gangen og hvordan personvern skulle håndteres. For å kunne håndtere flere brukere av gangen valgte gruppen å implementere cookies slik at systemet enkelt kunne identifisere hvem som bruker applikasjonen og allokere ressurser slik at flere kunne bruke applikasjonen samtidig.

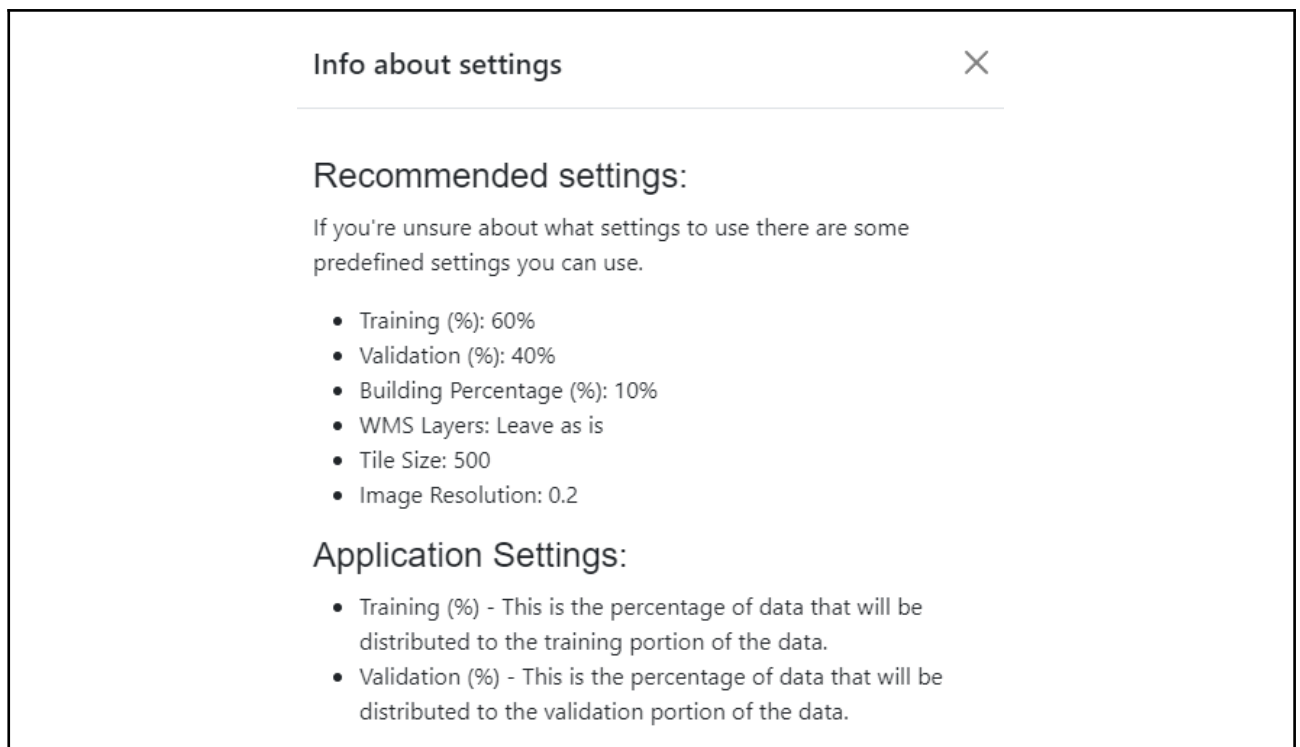
For å kunne håndtere cookies måtte det skrives en egen personvernerklæring. Dette er et juridisk krav og det finnes en rekke retningslinjer som må følges når systemet tillater persistens (Koch, u.å). Dette temaet blir diskutert mer i delpunkt [7.1.7 Personvern og GDPR](#).

5.2 Implementering av nytt brukergrensesnitt.

Dette delkapittelet tar for seg gruppens implementering av UI og UX komponenter i den nye applikasjonen. Kapittelet vil først beskrive endringene som ble gjort, og hvordan disse løser problemer som ble funnet under de ulike testene. Deretter blir resultater fra brukertesting av den nye applikasjonen analysert og sammenliknet med resultatene av den gamle.

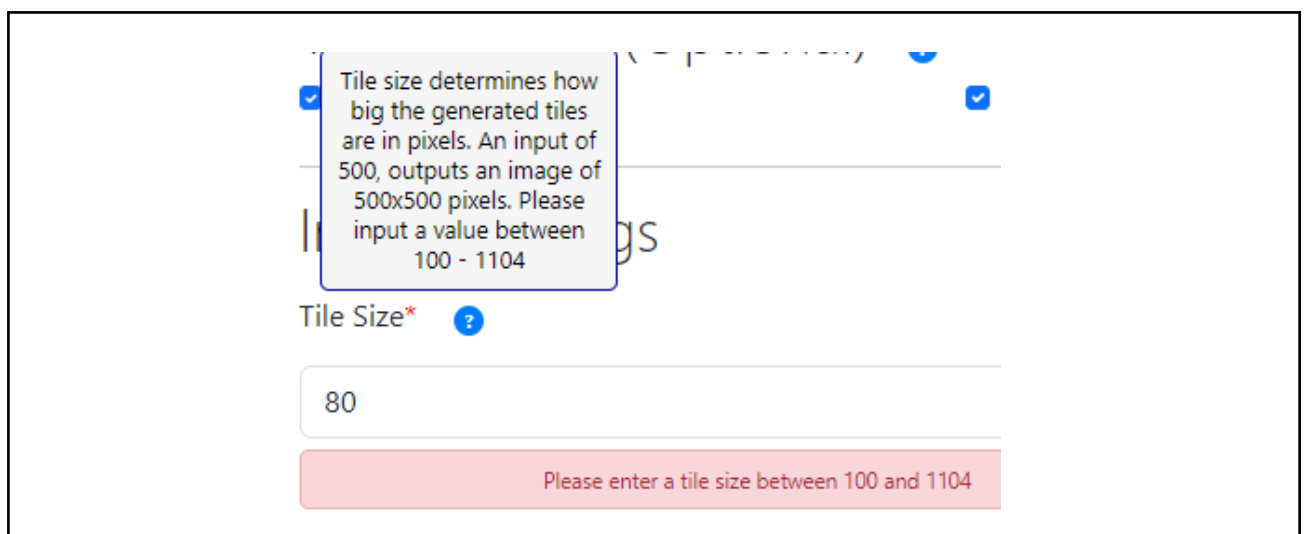
5.2.1 Forbedring av UI/UX basert på brukertester

Visibility - I den nye applikasjonen har gruppen implementert flere nye sider som en forside og infoside. Forsiden inneholder en kort og konsis beskrivelse av applikasjonens formål. Herfra kan brukeren velge å starte med å generere treningsdata eller lære mer om applikasjonen. Infosiden inneholder instruksjoner for hvordan den skal brukes, beskriver bakgrunnen for applikasjonen og informasjon om KartAI. Disse kan sees i [vedlegg P](#). Det er også implementert tooltips med forklaring på alle GIS-begreper. På hver side finner brukeren et popup vindu med informasjon om de ulike funksjonene på siden og anbefalt input for alle tekstfelt, figur 11. Disse endringene sikrer at alle brukere, uavhengig av teknisk bakgrunn, får en mest mulig lik forståelse av systemet og hvordan dataene deres vil påvirke sluttresultatet.



Figur 11: Popup vindu fra settings siden

Feedback og constraints - Det ble gjort en rekke endringer for å sikre korrekt innsamling av data og smidig interaksjon med applikasjonen. Gruppen utviklet blant annet feilmeldinger for tomme tekstfelt, ugyldig verdier, og det å fremheve obligatoriske felt med asterisk. Spesifikke tekstfelt som “training og building percentage” ble konfigurert til å kun akseptere numeriske verdier. På samme måte ble “upload geoJSON file” konfigurert til å kun tillate dette filformatet. Grunnet Kartverkets retningslinjer er ikke alle områdene i Norge frigitt for offentligheten. Under markering av kartområde får brukeren øyeblikkelig tilbakemelding dersom de har markert et område som ikke er frigitt.



Figur 12: Tile size input grenser

Flexibility og control - Flere nye funksjoner ble implementert i applikasjonen. På “methods” siden får brukeren nå mulighet til å velge mellom ulike datakilder som for eksempel FGB og COG, som figur 13 viser til. Mer om dette står i delkapittel [5.3 Implementering av nye datakilder](#). Dette var også et av hovedmålene definert av produkteieren for prosjektet.

The screenshot shows a web interface titled "Choose your data sources or upload a metadata file". It features three main sections at the top: "Choose label source*", "Choose orto source*", and "Upload a metadata file (Optional)". Each section has radio button options: "NGIS", "WMS", and "FGB" for labels; "WMS", "COG", and "Satellite" for ortho sources. A green notification bar in the center states "Data sources updated successfully!". Below these are three large buttons with icons: "Draw shape on map" (location pin), "Write coordinates" (keyboard), and "Upload geoJSON-file" (upload arrow).

Figur 13: Ny og oppdatert versjon

På “Draw shape on map” siden implementerte gruppen en søkefunksjon slik at man enkelt kunne finne frem til det ønskede området. Dersom brukeren ønsker å skrive inn koordinater selv har de nå flere koordinatsystemer å velge mellom. Videre på “settings” har gruppen også lagt til en rekke nye konfigureringsmuligheter for treningsdata. Som å definere image resolution og tile size til bildene, som kan sees på figur 12. Det er mulig å velge hvilke WMS layers man ønsker, som bygninger, veier og broer. Til slutt kan brukeren velge å laste ned dataen eller få den tilsendt på mail.

5.2.2 Resultater fra brukertester runde 2

Gruppen utførte tidligere nevnte brukertester for det nye designet. Resultatene viste at brukerne nå bedre forsto innholdet og formålet med applikasjonen. Brukertestene fremhevet likevel forskjellen mellom testpersoner med erfaring innen kunstig intelligens og testpersoner som ikke hadde det. Vedkommende som arbeidet med kunstig intelligens på Kartverket forstod applikasjonen intuitivt, leste lite og fylte raskt ut feltene for å generere treningsdata. De som hadde lite erfaring med kunstig intelligens brukte lengre tid og syntes det var vanskelig å plassere applikasjonen i en kontekst.

På grunnlag av de siste brukertestene ble det besluttet å fylle inn mer informasjon i de ulike elementene på applikasjonen, slik at en bruker uten erfaring med kunstig intelligens

skal forstå den bedre. Grunnet det var så kort tid igjen før overlevering av applikasjonen så bestemte gruppen seg ikke for å gjøre noen endringer på brukerflaten eller det visuelle.

5.3 Implementering av nye datakilder

Parallelt med at gruppen utviklet det nye designet til applikasjonen ble det også arbeidet på å legge til nye datakilder som det kunne genereres treningsdata fra. Gruppen implementerte en rekke nye datakilder i applikasjonen som alle gir en potensiell bruker muligheten til å generere treningsdata.

Datakildene COG og FGB ble implementert samtidig. COG filene ble lastet opp på Azure slik at applikasjonen har muligheten til å gjøre kall til den og hente ned bestemte deler av filen. Dette er fordelen ved en COG fil over en vanlig TIFF fil. Med COG kan man hente ned kun den data man trenger og sparer dermed både ressurser og tid. Fordelen med COG er at det kan gjøre spesifikke kall til hvert område, i motsetning til en større database. Dette fører til en reduksjon i bruk av ressurser samt tid for å laste ned nødvendige kartdata kraftig (Cogeo, u.å.).

FGB formatet er et relativt nytt filformat til bruk i GIS-applikasjoner. Fordelen med FGB over andre filformater er at det er designet for å kunne håndtere store mengder data og er optimalisert for bruk i GIS-applikasjoner (FME, u.å.). Ulempene med denne filtypen er at filen er lagret på serveren og må derfor oppdateres manuelt av dem som vedlikeholder den.

Det ble implementert støtte for ulike WMS servere. WMS sammen med SLD gir brukere muligheten til å selv angi stilen man vil ha på bildene man får tilbake fra applikasjonen. Denne typen teknologi er helt ny i GIS-sammenheng og gruppens prosjekt er trolig det første i Norge som tar det i bruk (L, F, Gyland. personlig kommunikasjon under et statusmøte, 20. mars 2024). Grunnen til at det var viktig for dette prosjektet var at for å kunne generere treningsdata på riktig måte må man ha et fasit grunnlag som består av kun svart hvit bilder. WMS`en har ikke mulighet til å gjøre dette uten hjelp av SLD. WMS`ene som ble implementert som datakilder var Norkart sin WMS for Ortofoto, og Kartverket sin WMS for label data.

Det var også et ønske fra en senioringeniør i Kartverket at applikasjonen hadde muligheten til å hente kartdata basert på satellittbilder. Dette ble implementert på samme måte som de andre datakildene. Man har nå muligheten til å velge å hente ut data fra Copernicus som gir satellittbilder. Copernicus er et system som tar i bruk et nettverk av flere satellitter som er en del av EUs romprogram. Disse satellittene overvåker miljøet på jorden til fordel for alle innbyggerne av EU (EU, u.å.).

En utfordring med dette er at bildene fra Copernicus ikke har like høy oppløsning som flybilder. Satellittbildene som er tilgjengelig fra Copernicus er ment for store områder. De har ikke detaljnivået til å kunne skille ut bygninger og andre lag. Denne delen av applikasjonen var kun en test av implementering og tilgangene til Copernicus har gått ut ved levering av denne rapporten. Dette er noe som kan videreutvikles ved en senere anledning av et annet team.

Gruppen hadde også som plan å starte med støtte for NGIS allerede i sprint 3. Det ble utsatt grunnet mangel på tilganger fra Kartverket. Gruppen fikk ikke tak i tilgangene de trengte for NGIS før slutten av sprint 7. Dette førte til at gruppen ikke fikk gjort nødvendige undersøkelser av NGIS og hvordan systemet fungerer. Det gruppen derimot fant ut var at områdene vi hadde blitt tildelt var i Vestfold i motsetning til Søm, og at dataen som kom tilbake fra SFKB var linjestrenger og ikke polygoner. Disse hindringene førte til at gruppen valgte å sette NGIS til side og heller fokusere på å forbedre de funksjonene som allerede hadde blitt implementert.

I tillegg til de nye datakildene ble det implementert en løsning slik at brukere har mulighet til å laste opp en metadata-fil fra tidligere prosjekter. Dette var et ønske fra oppdragsgivers for å gi brukeren muligheten til å gjenskape treningsgrunnlag man hadde laget tidligere og enkelt bruke dem igjen.

5.4 Refaktorering, dokumentasjon og testing

Gruppen brukte mye tid på å refaktorere og utbedre eksisterende kode. Den viktigste utbedringen som ble gjort var å sørge for at alle python- og javascript funksjonene i koden hadde god dokumentasjon som kunne bli brukt videre av andre studentgrupper eller interesserte i prosjektet. For å oppnå dette gjorde gruppen research på hvordan dokumentasjon burde skrives i de ulike språkene og sørget for at dokumentasjonen fulgte disse standardene.

Store deler av koden ble refaktorert slik at den lettere kan videreutvikles av fremtidige utviklere. Eksempel på dette var å flytte filene som lagrer brukerinput fra KartAI koden over til den egenskrevne koden. Prosessen med å åpne og skrive til filer ble også gjort mer generell da disse filene ikke lenger trengte å følge standarden til KartAI kodebasen. Et annet eksempel er at gruppen implementerte muligheten til å bruke Javascript moduler i prosjektet. Javascript moduler tillater gjenbruk av javascript kode ved at man kan importere funksjoner fra andre filer og lignende (Büşra Özmen, 2023).

Gruppen valgte på eget initiativ å implementere unittester i dette prosjektet. Dette ble gjort for å forsikre at koden skulle fungere som tiltenkt. Gruppens medlemmer hadde ikke mye erfaring med å skrive tester for en så komplisert kodebase. Dette gav oss verdifull erfaring på dette feltet. Testene som er skrevet dekker de viktigste funksjonene og burde gi

fremtidige utviklere muligheten til å enkelt finne ut av hva feilene er hvis det oppstår problemer med applikasjonen.

6. Resultat

Dette kapitlet tar for seg resultatet fra prosjektperioden og hvilke forbehold som må tas hvis prosjektet skal videreutvikles. Dette kapitlet beskriver også hvordan man kan bruke applikasjonen til å generere treningsdata og inkluderer også [instruksjonsvideo \(https://www.youtube.com/watch?v=Ztaiu1blkx8\)](https://www.youtube.com/watch?v=Ztaiu1blkx8) for bruk.

6.1 Oppfylling av mål og krav

Dette delkapitlet dekker hvorvidt målene som ble etablert i delkapittel [1.5 Mål og krav](#) ble møtt og om gruppen føler at løsningen ble godt implementert, eller om de har muligheter for forbedring.

Distribuere applikasjonen på nett via Microsoft Azure - Applikasjonen har blitt distribuert på nett via Microsoft Azure og den fungerer på denne plattformen. Selve prosessen med å oppdatere applikasjonen er enkel og burde ikke føre til store problemer videre. Det som derimot kan føre til problemer med Azure senere er at planen gruppen valgte i delkapittel [5.1 Distribuering av webbløsning](#) ikke har mye lagringsplass på Azure plattformen. Dette kan føre til problemer senere hvis applikasjonen for eksempel skal knyttes opp mot en database eller lignende. Med at applikasjonen nå er deployet på Azure gjør at første mål i prosjektet, “Distribuere applikasjonen på nett via Microsoft Azure”, har blitt oppfylt.

Forbedre brukervennligheten for applikasjonen - Brukervennligheten i applikasjonen har blitt forbedret ved at det nå er klarere hva som er formålet og hvordan den brukes. Felt har fått forklaringer, applikasjonen har fått en “Les mer” side som forklarer alle bruksområdene, og den følger blant annet flere av Benyons designprinsipper. Applikasjonen har potensial til å bli mer brukervennlig enn hva den er nå. For fremtiden er det anbefalt å fokusere på tjenstedesignet til applikasjonen, og redesigne den fra et brukerperspektiv basert på innsiktsarbeid, problemforståelse, og testing (Hvidsten et al., 2021, s. 13).

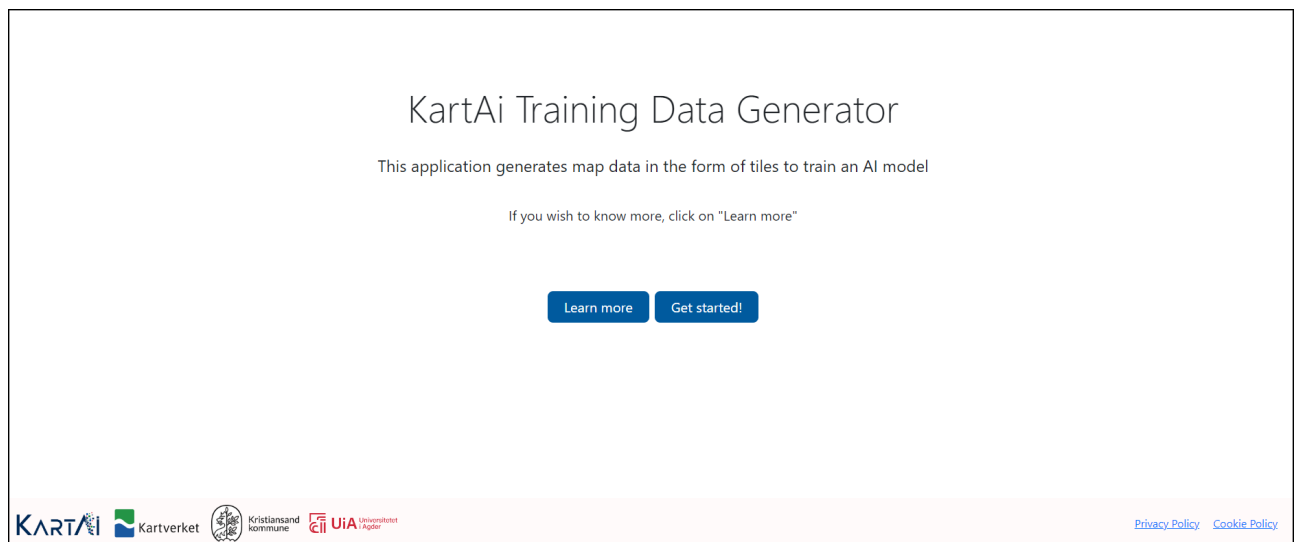
Legge til nye datakilder for WMS, COG, FGB og NGIS - Applikasjonen har blitt utvidet med flere datakilder og gruppen er veldig fornøyd med at de fikk til brukerdefinert styling ved hjelp av SLD mot WMS. Det at NGIS ikke fungerer er derimot et punkt der gruppen ikke oppnådde målet som ble satt i starten av prosjektet. Det er ulike grunner til at NGIS ikke ble implementert som ble dekket i delkapittel [5.3 Implementering av nye datakilder](#) og vil bli videre beskrevet i delpunkt [7.1.6 Tilgang på ressurser](#). COG og FGB ble også

implementert etter kriteriene som nevnt tidligere, dette ble gjort slik at man kunne hente ut data fra ytterlige datakilder om man ønsker dette. Gruppen fikk også til å implementere støtte for satellittbilder via Copernicus. Dette var et mål som ikke var definert i starten av prosjektperioden, men et ønske som kommer underveis i arbeidet.

Refaktorere eksisterende kode til å være lettere å videreutvikle - Koden som er tilgjengelig nå er forhåpentligvis enklere å videreutvikle. Dette er derimot en meget subjektiv vurdering som avhenger av en persons sitt kompetansenivå. Gruppen har strebet etter å gjøre denne delen av applikasjonen bedre og utviklet dokumentasjon som følger standarder samt implementert unit tester slik at applikasjonen blir lettere å feilsøke på et senere tidspunkt. Målet med å refaktorere koden kom senere inn i utviklingsprosessen da den eksisterende koden var utfordrende å sette seg inn i.

6.2 Applikasjonen og hvordan den brukes

Applikasjonen som overleveres ved slutten av prosjektperioden er en webapplikasjon der brukere har mulighet til å hente ut kartdata for trening av maskinlæringsalgoritmer fra ulike datakilder og få resultatene tilsendt på mail eller ved nedlastning. For å bruke applikasjonen kan man enten laste ned og kjøre kildekoden fra [Github](https://github.com/kartAI/NGIS-training-machine-webclient) (<https://github.com/kartAI/NGIS-training-machine-webclient>), dette krever eksterne tilganger fra Kartverket/Norkart. Eller gå gjennom Azure løsningen gruppen dannet for prosjektet. Nettsiden er hostet [her](https://ngis.azurewebsites.net). (<https://ngis.azurewebsites.net>)



Figur 14: Forside applikasjon

Figur 14 viser til hva en potensiell bruker først møter når man ankommer applikasjonen. Her får bruker allerede to valg. Han kan enten trykke "Learn more", som gir brukeren mer informasjon om applikasjonen, hvem som har vært med å laget den, og hvem som står

bak. Trykker brukeren seg videre i applikasjonen med “Get started!” så kommer en videre i applikasjonen.

Choose your data sources or upload a metadata file

Choose label source* ?
☐ NGIS ☒ WMS ☐ FGB

Choose orto source* ?
☐ WMS ☒ COG ☐ Satellite

Upload a metadata file (Optional) ?

Data sources updated successfully!


Draw shape on map


Write coordinates

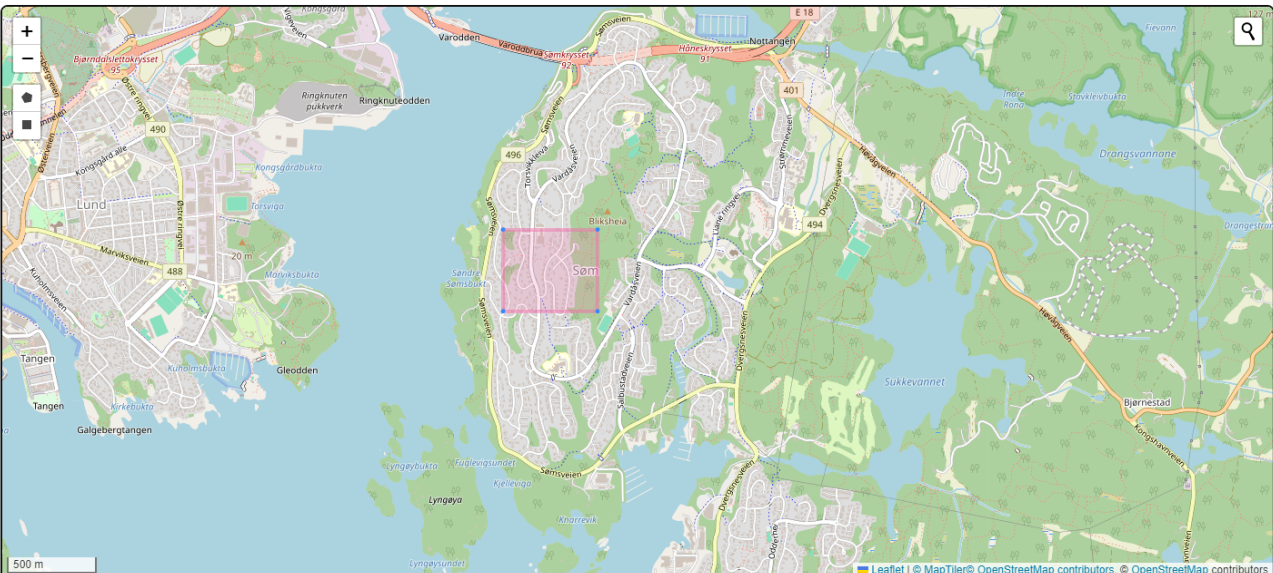

Upload geoJSON-file

Figur 15: Valg av metode

Hvis en bruker skulle gå videre fra forsiden så blir hen presentert denne siden. Her blir brukeren møtt med flere valg. Først må man velge datakilde, deretter metode for bilde generasjon. Dette gjør man enten med å tegne på kartet, skrive inn en streng med fire ulike koordinater, eller laste opp en GeoJSON fil. Det er også mulighet til å gjenskape et tidligere datasett, med opplasting av en metadata fil, som er inkludert i alle nedlastningene i applikasjonen. Blå spørsmålstegn ved siden datakilder forklarer forskjellen på dem.

Mark desired training area

Mark the desired training area(s) on the map using the rectangle or polygon shape. Click on the rectangle to shape it and click “Next” when you are done.



Figur 16: Tegn på kartet

Hvis en bruker skulle ønske å tegne på kartet eller legge inn koordinater manuelt, så blir hen møtt med siden i figur 16. Skulle en velge datakilder som ikke er WMS/WMS, så får brukeren muligheten til å se de tilgjengelige områdene disse dekker.

The screenshot shows a web form titled "Settings". It is divided into several sections:

- Define Application Settings:** Contains three input fields: "Training (%)*" with value 100, "Validation (%)*" with value 0, and "Building percentage (%)*" with value 5. Each field has a blue question mark icon.
- WMS Layers (Optional):** Contains three checked checkboxes with corresponding color swatches: "Buildings" (red), "Roads" (green), and "Bridges" (blue). Each checkbox has a blue question mark icon.
- Image settings:** Contains two input fields: "Tile Size*" with value 500 and "Image Resolution*" with value 0.2. Both fields have blue question mark icons.
- Dataset settings (Optional):** Contains two input fields: "E-mail (Optional)" with value "test@email.com" and "Dataset Name (Optional)" with value "testdataset".

At the bottom of the form are two buttons: a red "Cancel" button and a blue "Order" button.

Figur 17: Settings

I figuren over så kan brukeren legge inn ønskede innstillinger for treningsdataen. Hvis man skulle ønske å få datasettet tilsendt på mail, legger en ved en mailadresse. Det er også mulig å navngi datasettene allerede i applikasjonen slik man lett kan holde filene sine oversiktlige. Training og validation omgjør mappestrukturen levert etter brukerens behov. Building percentage er minimumskrav til hvor mange prosent bygninger som må være tilstede i dataen, for at det skal bli generert et bilde. Det er også mulig å fargelegge fasitdataen ved bruk av WMS layers, dette utgjør kun en forskjell dersom brukeren skulle velge WMS fasit. De siste parametrene omhandler bildene, slik som størrelsen på tiles, og oppløsningen på treningsgrunnlaget.

7. Refleksjon

Dette kapittelet dekker refleksjoner som gruppen har gjort i prosjektperioden. Gruppen har gjort seg opp refleksjoner rundt forskjellige temaer som hybrid arbeidsmetode, hvordan håndtere ulike interessenter i et prosjekt og hvordan mangel på kompetanse har blitt behandlet. Disse refleksjonene er delt opp i to hoveddeler der den første delen er erfaringer med prosjektarbeid og metodikk, mens den andre delen er erfaringer som handler om å ta i bruk og lære ny teknologi.

7.1 Prosjektarbeid

Dette delkapittelet dekker refleksjoner gruppen har gjort rundt prosjektarbeid. Dette innebærer scrum og metodikk, verktøy som ble brukt for å gjennomføre prosjektet og de ulike tiltakene som ble tatt for å sikre progresjon og kvalitet i prosjektet.

7.1.1 Agil arbeidsmetodikk

Det å arbeide agilt i en setting som har vært veldig lik en ordentlig jobbsituasjon har gitt gruppen ekstremt mange verdifulle erfaringer. Vi har prøvd å arbeide agilt i tidligere prosjekter, men har aldri helt fått metodikken til å fungere for oss, da de prosjektene har vært av mindre størrelse. Det å kunne dedikere 100% av arbeidstiden vår til en metodikk istedenfor bare noen få timer innenfor et emne har virkelig vist oss fordelene og ulempene med slik metodikk. Alle metodene og aktivitetene innenfor scrum rammeverket har tilført verdi til prosjektet og det føltes aldri som om rammeverket stod i veien for arbeidet slik det har gjort tidligere.

Den største utfordringen med å arbeide agilt har vært estimering og føring av timer. Gjennom hele prosjektperioden har vi hatt problemer med å estimere riktig timeantall på oppgaver og det har også hendt at vi har lagt til nye oppgaver under sprintene. Dette er et resultat av at vi ofte har arbeidet med komplekse temaer og ikke har hatt full oversikt over hva de forskjellige oppgavene innebærer. Estimatenes så derimot en forbedring utover prosjektperioden og gruppen ble også bedre på nedbrytning av større oppgaver i mindre elementer som kunne tilføres i backloggen.

7.1.2 Prosjektstyringsverktøy

Gruppen var i starten kritiske til å bruke Github Projects som prosjektstyringsverktøy, men det viste seg å fungere bra. Mye av skepsisen kom fra tidligere semesteret der Github Projects ikke hadde hatt den funksjonaliteten gruppen var ute etter. Disse problemene har blitt løst i mellomtiden og desto mer gruppen brukte verktøyet desto flere funksjonaliteter ble brukt og verdien av verktøyet ble større.

For kommunikasjon har gruppen brukt flere digitale kommunikasjonsplattformer, primært Discord. Digital kommunikasjon har fungert bra og har tilført stor verdi til prosjektarbeidet fordi vi hele tiden hadde et referat av hvordan arbeidet gikk og viktige ressurser for læring samlet på samme sted. Andre digitale kommunikasjonsplattformer som gruppen brukte var Teams og Google Meet.

Gruppen brukte også Google Disk til å lage dokumenter, presentasjoner og lignende. Disse verktøyene har gruppen brukt gjennom hele studieløpet og har vært uvurderlige. Det som er bra med å bruke slike webbaserte verktøy er at det også kan gi verdi for fremtidige studentgrupper som skal jobbe med prosjektet fordi de får tilgang til flere av våre planleggingsdokumenter i vedleggene.

7.1.3 Kvalitetssikring

Alle punktene nevnt under [2.2 Kvalitetssikring](#) har vært ekstremt viktige for å sikre at kvaliteten på prosjektet er der gruppen ønsker at den skal være. Disse verktøyene har vist seg å ha stor verdi for hele prosjektet. For eksempel er risikomatrisen blitt brukt aktivt og oppdatert jevnlig, dette har gitt gruppen muligheten til å kontinuerlig se på hvilke risikoer som truer gjennomføringen av prosjektet og justere kursen deretter. Det å følge en tydelig definert metodikk har også tilført mye til prosjektet, når vi har brukt scrum aktivt har vi redusert mengden tid vi bruker på planlegging og annet administrativt arbeid. Det var en stor fordel å kunne velge ut arbeidsoppgaver rett fra backloggen og vite at man alltid hadde noe å gjøre.

7.1.4 Ulike interessenter

Noe som var litt unikt med vårt prosjekt var at vi hadde mange interessenter som alle hadde sine egne meninger og innspill om hva applikasjonen burde inneholde. Det å skulle gjøre flere interessenter fornøyde var helt nytt for gruppen, tidligere prosjekter har kun hatt enten en interessent eller oppdragsgiver. Vi fant tidlig ut at det var viktig med åpen kommunikasjon med oppdragsgiver og de forskjellige interessentene for å holde forventningene til prosjektet realistiske. På grunn av disse interessentene var det også viktig å vurdere nye arbeidsoppgaver etter hva som var relevant for prosjektet. Gruppen følte etter en diskusjon at arbeidsoppgaver som ikke direkte samsvarte med hovedoppgaven, måtte nedprioriteres. Det viktigste var å ha et helhetlig prosjekt, både med programmering og prosjektrapport.

Denne problemstillingen førte også til vanskeligheter med å holde fremdrift tidlig i prosjektet. Når det hele tiden kom nye innspill fra forskjellige interessenter og vi enda ikke hadde et komplett bilde av hva som skulle gjøres var det vanskelig å prioritere oppgavene som ble gitt. Dette løste vi gjennom å fokusere på prioriteringene til produkteieren, og at han fikk avgjøre om en oppgave var relevant eller verdifull for gruppen å gjennomføre.

7.1.5 Hybrid arbeidsmetode

Prosjektet har gått gjennom flere faser, noen av disse fasene krevde mer samarbeid, mens andre krevde mer selvstendig arbeid innad i gruppen. I de fasene der det har vært mange små oppgaver, som krever lite samarbeid, har ofte flere av gruppemedlemmene valgt å arbeide hjemmefra. Dette har både fordeler og ulemper. Fordelene er at man kan sitte uforstyrret på et egnet sted og arbeide med oppgaver uten at man trenger å delta i den støyen som ofte oppstår når man jobber i grupper eller i kontorlokaler. Ulempene er at man ikke har like lett tilgang på hjelp fra andre gruppemedlemmer hvis man trenger det.

Dette har vi løst gjennom aktiv bruk av de kommunikasjonsverktøyene som er nevnt tidligere og det har ikke ført til store problemer.

Gruppen har også som tidligere nevnt i rapporten jobbet hybrid dette semesteret. Opplevelsene rundt hybrid har vært meget positive, og gruppen tror helhetlig at dette ikke har ført til ineffektivt arbeid, heller tvert imot. Dette samsvarer med teorien om sosial loffing som sier at man arbeider mindre effektivt i grupper enn når man jobber alene (Svartdal, 2020.). Gruppen føler de har klart å holde kommunikasjonen oppe i disse periodene, samtidig som den har både vært effektiv, konkret og faglig. Dette gode samarbeidet hadde antageligvis ikke hatt like gode resultater hvis gruppen ikke hadde vært trygge på hverandre fra tidligere prosjekter.

7.1.6 Tilgang til ressurser.

En annen viktig erfaring gruppen har gjort seg under arbeidet er med et så stort prosjekt med mange forskjellige interesser, settes det ulike krav og ønsker fra disse. Applikasjonen vi har laget krever mange tilganger og disse tilgangene var splittet mellom de forskjellige interessentene. Det var ofte også forskjellige personer innad i organisasjonene som hadde ulike tilganger vi trengte og det ble derfor mye venting, og kommunikasjon frem og tilbake på mail. Dette er en problemstilling fylkeskartsjefen i Kartverket også er klar over. Det handler om at det ikke er nok ressurser tilgjengelig til å bistå i et studentprosjekt ved siden av og det blir derfor litt nedprioritert. Det at det var såpass tidkrevende å få tilgang til de forskjellige ressursene, at det hendte fremgangen i prosjektet stoppet opp. Dette problemet ble bedre utover i prosjektperioden, men til tider var dette utfordrende. Det at vi ikke fikk de tilgangene vi behøvde tidlig i prosjektet, har ført til at oppgavene ble omprioritert underveis. Tidligere i rapporten er det nevnt hvordan vi har måttet flytte på utviklingen av NGIS på grunn av manglende tilganger.

7.1.7 Personvern og GDPR

Brukerøker blir lagret gjennom bruk av informasjonskapsler, også kalt cookies. I tillegg så ber applikasjonen om brukerens e-postadresse om brukeren ønsker å få tilsendt treningsdata til en e-postadresse istedenfor å laste den ned. Begge implementasjonene faller inn under personopplysningsloven som består av nasjonale regler og EUs personvernforordning (også kalt GDPR - General Data Protection Regulation). Personopplysningsloven handler om all innsamling, behandling og bruk av

personopplysninger (Datatilsynet, 2023). Generert treningsdata blir gitt en tilfeldig ID som kan kobles til brukerens ID til økten. Siden det er mulig å få denne tilsendt på mail, så blir dette et problem som faller under personopplysningsloven, hvor dataene blir midlertidig lagret. Vi kontaktet veileder ved Tietoevry for å få klarhet i gjeldende lover og regler. All informasjon om en bruker eller annen informasjon som kan føre tilbake til en bruker går inn under personopplysningsloven. Ved bruk av informasjonskapsler så må brukeren informeres om dette og brukeren må godkjenne det (Datatilsynet, 2019). Ettersom vi imidlertid lagrer en brukers e-postadresse så må brukeren være informert om hvordan denne informasjonen blir behandlet i applikasjonen og hvem som er ansvarlig for applikasjonen gjennom en personvernerklæring (Datatilsynet, 2023). Dette førte da til at det ble opprettet en cookie policy, samt en personvernerklæring (Privacy Policy).

7.2 Erfaringer med teknologier og utvikling

Dette delkapittelet tar for seg gruppens erfaringer med det å videreutvikle et prosjekt, kompetansegap, nye teknologier, og det å jobbe aktivt med forskjellige AI-verktøy.

7.2.1 Kompetansegap

Det at gruppen ikke hadde erfaring med geomatikk og GIS fra tidligere var en stor utfordring, spesielt i startfasen. Gruppen brukte veldig mye tid på å sette seg inn i de forskjellige begrepene og rammeverkene som blir brukt når man utvikler en applikasjon innenfor GIS. Vi fikk etterhvert et innføringskurs i disse temaene som var svært lærerikt, selv om gruppen følte at introduksjonen kom litt for sent i semesteret så var det uansett verdifullt for å få et mer helhetlig bilde av prosjektet, og sette det i kontekst.

Kompetansegapet når det kom til utvikling av kompliserte applikasjoner var også et stort hinder i starten. Det ble brukt veldig mye tid på å sette seg inn i hva og hvordan den forrige studentgruppen hadde jobbet og hvordan verktøyene de har valgt fungerer. Et eksempel på dette er FastAPI rammeverket der vi brukte en del tid på å utvikle mindre prosjekter for å forstå hvordan det fungerer istedenfor å hoppe rett over på NGIS koden.

7.2.2 Videreutvikling av et tidligere studentprosjekt

Det å videreutvikle et prosjekt har vært en interessant opplevelse for gruppen. Videreutvikling i en arbeidskontekst og videreutvikling i kontekst av et studentprosjekt er nok to vidt forskjellige ting og vi har erfart at det er utfordrende. Som tidligere nevnt har mye av koden som ble skrevet av fjorårets studenter enten blitt refaktorert eller direkte slettet. Dette er en konsekvens av at applikasjonens formål har endret seg, og mye av koden som ble skrevet i fjor ikke kunne brukes til å nå de målene vi hadde satt oss for dette prosjektet.

Prosjektet vi startet i januar hadde ikke blitt vedlikeholdt siden den andre studentgruppen overleverte det. Det var utfordrende å sette seg inn i eksisterende kode som ikke hadde god dokumentasjon samtidig som vi slet med kompetansegapet innenfor utviklingen, teknologi og GIS. Derfor har vi sørget for å dokumentere godt i vår kode, eksempel på dette kan sees i [vedlegg Q](#).

Vi er likevel takknemlig for innsatsen den forrige gruppen hadde lagt inn i prosjektet og store deler av koden på frontend ble videreført av oss. Dette førte til at vi kunne fokusere mer på tekniske og UI/UX oppgaver. Vi håper at det prosjektet vi overleverer til neste studentgruppe er lett å sette seg inn i og kan videreutvikles etter oppdragsgivers ønske uten problemer.

7.2.3 Bruk av ChatGPT og andre AI-verktøy

I startfasen var gruppen veldig imot å ta i bruk kunstig intelligens, grunnet stigmaet rundt bruk av det innenfor academia. Mye av denne skepsisen forsvant etter at temaet ble diskutert med oppdragsgiver og veileder fra Tietoevry. Grunnen til dette var at gruppen ble oppfordret til å bruke slike verktøy, fordi de ble aktivt brukt i jobbsammenheng, og gruppen ville tilnærme dette prosjektet et reelt arbeidsmiljø. ChatGPT har hjulpet til med å skrive deler av koden, samt idèmyldring for funksjonalitet i applikasjonen. Github Copilot, som er Github, Microsoft og OpenAI sin AI-modell innenfor utvikling, kan direkte implementeres i Visual Studio Code, og har vært et effektivt verktøy når det kommer til dokumentasjon i koden. Gruppen sitter igjen med en følelse av å ha benyttet disse verktøyene på en god måte, og er fornøyde med bruken av kunstig intelligens som et hjelpemiddel.

Selv om gruppen brukte generativ kunstig intelligens i prosjektperioden for utvikling var det viktig å merke seg at gruppemedlemmene var pliktet til å ha full forståelse av koden som ble lagt inn i kodebasen, og sikre at den ikke medførte sikkerhetsfeil eller mangler samtidig som den fulgte kodestandardene.

Det var også strenge regler for bruk av generativ kunstig intelligens når det kom til skrivingen av selve rapporten. Her var det ikke tillatt med noe bruk av kunstig intelligens da slike generative AI verktøy ikke klarer riktig kildehenvisning, og sliter ofte med å holde seg til fakta. Slike modeller er laget for å genere tekst, ikke til å finne tilstrekkelig informasjon.

8. Konklusjon

Denne rapporten har presentert gjennomføringen av bachelorprosjektet “Tilgjengeliggjøring av nasjonale kartdata til bruk i AI-algoritmer”. Prosjektet er en videreutvikling av et tidligere studentprosjekt våren 2023. Vi vil derfor takke den forrige studentgruppen som har lagt grunnlaget for vårt arbeid. Hovedformålet med prosjektet var å tilgjengeliggjøre kartdata for AI modeller og ta i bruk ny teknologi ved generering av kartdata.

Arbeidet vi har gjort er å distribuere applikasjonen på nett via Azure og lagt til muligheten for at flere kan bruke applikasjonen samtidig gjennom informasjonskapsler. I tillegg til dette har vi hatt fokus på å forbedre brukeropplevelsen og brukergrensesnittet til applikasjonen slik at den er lettere å bruke og ikke krever store mengder GIS-kunnskaper. Vi har også lagt til flere datakilder i prosjektet, blant annet COG, FGB og WMS mot SLD. Disse datakildene gjør det mulig å hente kartdata fra flere kilder og generere treningsdata basert på disse dataene.

Etter en samtale med fylkeskartsjefen ved Kartverket Agder fikk vi også vite at applikasjonen skal videreutvikles av fremtidige studentgrupper. Det er flere funksjoner som har blitt diskutert med oppdragsgiver som ikke har blitt implementert i ferdigstilt versjon. Blant annet må NGIS implementeres på nytt. Gruppen har startet arbeidet med å implementere NGIS og det er mulig at noe av koden som allerede er skrevet kan brukes videre. Denne koden er derimot ikke blitt testet grunnet dårlig tid og mangel på tilganger som ble beskrevet tidligere i rapporten.

Til slutt vil vi rette en takk til Kartverket og Tietoevry samt øvrige interessenter i dette prosjektet. Vi er utrolig heldige som har fått lov til å jobbe med et så spennende og potensielt banebrytende prosjekt. Dynamisk styling ved hjelp av SLD har aldri blitt implementert slik som vi har gjort det i dette prosjektet og ifølge fylkeskartsjefen for Kartverket Agder er dette første gang noen i Norge har gjort dette og muligens helt nytt på verdensbasis. Vi har fått muligheten til å bidra til et spennende prosjekt og det er vi takknemlige for. Vi håper at arbeidet som har blitt beskrevet i denne rapporten kan hjelpe med å utvikle gode karttjenester som gir hele Norge tilgang på nøyaktige og sikre kartdata.

Litteraturliste

- Adobe. (17. november, 2022). *Project scope — definition, best practices, examples, and more*. Adobe Experience Cloud. Hentet februar 8, 2024, fra <https://business.adobe.com/blog/basics/project-scope-definition-best-practices-examples-and-more>
- Aven, T. (10. oktober, 2022). *risikomatrise – Store norske leksikon*. Store norske leksikon. Hentet 9. mai, 2024, fra <https://snl.no/risikomatrise>
- Aven, T. (16. mars, 2023). *risiko – Store norske leksikon*. Store norske leksikon. Hentet 9. mai, 2024, fra <https://snl.no/risiko>
- Bell, D. (9. oktober, 2023). *Explore the UML sequence diagram*. IBM Developer. Hentet 26. april, 2024, fra <https://developer.ibm.com/articles/the-sequence-diagram/>
- Benyon, D. (2019). *Designing User Experience: A Guide to HCI, UX and Interaction Design* (4th ed.). Pearson Education Limited.
- Bredesen, I. M., Ekholt, T., Fugledal, N., Skjæveland, K. J. I., & Slettene, S. (22. mai, 2023). *Automatisering av dataflyt fra nasjonale kartbaser mot AI-algoritmer*. Automatisering av dataflyt fra nasjonale kartbaser mot AI-algoritmer. Hentet 8. februar, 2024, fra <https://kompetansetorget.uia.no/fullfoerte-oppgaver/automatisering-av-dataflyt-fra-nasjonale-kartbaser-mot-ai-algoritmer>
- Büşra Özmen, B. (8. november, 2023). *JavaScript Modules: Strategies to Improve Code Quality and Efficiency*. Medium. Hentet 7. mai, 2024, fra <https://medium.com/cstech/javascript-modules-e3ef549a9e33>
- Cloudflare. (u.å.). *What is the Simple Mail Transfer Protocol (SMTP)?* Cloudflare. Hentet 14. mai, 2024, fra <https://www.cloudflare.com/learning/email-security/what-is-smtp/>
- Codacy. (14. november, 2023). *Coding Standards: What Are They and Why Are They Important?* Codacy | Blog. Hentet februar 8, 2024, fra <https://blog.codacy.com/coding-standards>

Cogeo. (u.å.). *Why Cloud Optimized GeoTIFF?* COGEO. Hentet 16. april, 2024, fra

<https://www.cogeo.org>

Datatilsynet. (10. oktober, 2019). *Bruk av informasjonskapsler (cookies)*. Datatilsynet. Hentet 10. mai,

2024, fra <https://www.datatilsynet.no/personvern-pa-ulike-omrader/internett-og-apper/cookies/>

Datatilsynet. (27. juli, 2023). *Informasjon og åpenhet*. Informasjon og åpenhet | Datatilsynet. Hentet 10.

mai, 2024, fra

<https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/informasjon-og-apenhet/hvordan-skal-virksomheten-gi-informasjon/>

Datatilsynet. (23. november, 2023). *Om personopplysningsloven med forordning*. Datatilsynet. Hentet 5.

april, 2024, fra

<https://www.datatilsynet.no/regelverk-og-verktoy/lover-og-regler/om-personopplysningsloven-og-nar-den-gjelder/>

Davies, B. (u.å.). *Product Roadmap Guide: What is it & How to Create One*. Atlassian. Hentet 13. mai,

2024, fra <https://www.atlassian.com/agile/product-management/product-roadmaps>

Docker. (u.å.). *Docker overview*. Docker Docs. Hentet 30. januar, 2024, fra

<https://docs.docker.com/get-started/overview/>

Esri. (u.å.). *What Is GeoAI? | Accelerated Data Generation & Spatial Problem-Solving*. Esri. Hentet 2.

mai, 2024, fra <https://www.esri.com/en-us/capabilities/geoai/overview>

EU. (u.å.). *About Copernicus*. Copernicus. Hentet 15. mai, 2024, fra

<https://www.copernicus.eu/en/about-copernicus>

Evelon & Smith, J.-A. (1. mars, 2019). *Hva er Microsoft Azure? – Evelon*. Evelon. Hentet 22. april, 2024,

fra <https://www.evelon.no/artikler/hva-er-microsoft-azure>

FastAPI. (u.å.). *FastAPI framework, high performance, easy to learn, fast to code, ready for production*.

FastAPI. Hentet 26. januar, 2024, fra <https://fastapi.tiangolo.com>

Figma. (u.å.). *High-Fidelity Prototyping: What Is It And How Can It Help?* Figma. Hentet 9. mai, 2024, fra

<https://www.figma.com/resource-library/high-fidelity-prototyping/>

Flatgeobuf. (u.å.). *Flatgeobuf*. FlatGeobuf | flatgeobuf. Hentet 22. mars, 2024, fra <https://flatgeobuf.org>

FME. (u.å.). FlatGeobuf Reader/Writer. Hentet 6. mai, 2024, fra

<https://docs.safe.com/fme/html/FME-Form-Documentation/FME-ReadersWriters/flatgeobuf/flatgeobuf.htm>

Geonorge. (u.å.). *Formater*. Geonorge. Hentet 6. februar, 2024, fra

<https://www.geonorge.no/aktuelt/om-geonorge/slik-bruker-du-geonorge/formater/>

Github. (u.å.). *About Projects*. GitHub Docs. Hentet 30. januar, 2024, fra

<https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>

Github. (u.å.). *Hello World*. GitHub Docs. Hentet 30. januar, 2024, fra

<https://docs.github.com/en/get-started/quickstart/hello-world>

HEAVY.AI. (u.å.). *What is GeoTIFF? Definition and FAQs*. HEAVY.AI. Hentet 26. januar, 2024, fra

<https://www.heavy.ai/technical-glossary/geotiff>

Hvidsten, A., Rai, R., Helland, S., & Henriksen, T. (2021). *Introduksjon til tjenstedesign* (1st ed.).

Cappelen Damm AS.

Industriskolen. (27. mars, 2020). *Kvalitetssikring og kvalitetssystem - Produktivitet- og kvalitetsstyring*.

NDLA. Hentet 1. februar, 2024, fra

<https://ndla.no/subject:1:29212872-62d6-4555-89fd-b85e7f3f9411/topic:e7484706-399d-4d4b-a42b-44afd29f8dc8/resource:de5c5aac-5060-4845-9286-ee1739616074>

KartAI. (u.å.). *Om KartAI*. KartAI. Hentet 6. februar, 2024, fra <https://kartai.no/om-oss/>

Kartverket. (u.å.). *NGIS*. Kartverket. Hentet 26. januar, 2024, fra

<https://www.kartverket.no/geodataarbeid/ngis>

Kartverket. (24. mai, 2023). *Kva Kartverket gjer*. Kartverket. Hentet 8 februar, 2024, fra

<https://www.kartverket.no/om-kartverket/kva-kartverket-gjer>

Kartverket. (19. januar, 2024). *FKB produktspesifikasjoner*. Kartverket. Hentet 26. januar, 2024, fra

<https://www.kartverket.no/geodataarbeid/geovekst/fkb-produktspesifikasjoner>

Koch, R. (u.å.). *Cookies, the GDPR, and the ePrivacy Directive - GDPR.eu*. GDPR compliance. Hentet

15. mars, 2024, fra <https://gdpr.eu/cookies/>

- Li, Z., & Ning, H. (2023). Autonomous GIS: the next-generation AI-powered GIS. *International Journal of Digital Earth*, 16(2), 4668-4686. 10.1080/17538947.2023.2278895
- loadfocus. (u.å.). *Hva er Automatisert Tilgjengelighetstesting?* loadfocus. Hentet 12. mai, 2024, fra <https://loadfocus.com/no-no/glossary/what-is-automated-accessibility-testing>
- Lucidchart. (u.å.). *Lucidchart*. Lucidchart: Intelligent Diagramming. Hentet 7 mai, 2024, fra <https://www.lucidchart.com/pages/>
- Mæhlum, L. (3. januar, 2023). *WMS – Web Map Service – Store norske leksikon*. Store norske leksikon. Hentet 26 januar, 2024, fra https://snl.no/WMS_-_Web_Map_Service
- Mæhlum, L., & Berg, T. R. (13. november, 2023). *ortofoto – Store norske leksikon*. Store norske leksikon. Hentet 26. januar, 2024, fra <https://snl.no/ortofoto>
- Marr, B. (10. mai, 2013). *A Short History Of ChatGPT: How We Got To Where We Are Today*. Forbes. Hentet 2. mai, 2024, fra <https://www.forbes.com/sites/bernardmarr/2023/05/19/a-short-history-of-chatgpt-how-we-got-to-where-we-are-today/>
- Microsoft. (u.å.). *Docker Deployment on Azure*. Microsoft Azure. Hentet 30. januar, 2024, fra <https://azure.microsoft.com/en-us/products/kubernetes-service/docker>
- Mozilla. (2. oktober, 2023). *Guidelines for writing JavaScript code examples - The MDN Web Docs project* | MDN. MDN Web Docs. Hentet 14. mai, 2024, fra https://developer.mozilla.org/en-US/docs/MDN/Writing_guidelines/Writing_style_guide/Code_style_guide/JavaScript
- Mozilla Web Docs. (21. juli, 2023). *CSS: Cascading Style Sheets* | MDN. MDN Web Docs. Hentet 30. januar, 2024, fra <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Mozilla Web Docs. (28. august, 2023). *HTML basics - Learn web development* | MDN. MDN Web Docs. Hentet 30. januar, 2024, fra https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

Mozilla Web Docs. (23. januar, 2024). *What is JavaScript? - Learn web development* | MDN. MDN Web Docs. Hentet 31. januar, 2024, fra https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

National Archives. (18. januar, 2020). *PRONOM | Search by format*. PRONOM | Search by format. Hentet 2. mai, 2024, fra <https://www.nationalarchives.gov.uk/PRONOM/Format/proFormatSearch.aspx?status=detailRepo&id=2066>

Norkart. (u.å.). *Om oss - Sammen skaper vi smartere samfunn*. Norkart. Hentet 22. februar, 2024, fra <https://www.norkart.no/om-oss>

Nurthern, J., & Cooper, M. (23. januar, 2024). *WAI-ARIA Overview | Web Accessibility Initiative (WAI)*. W3C. Hentet 7. mai, 2024, fra <https://www.w3.org/WAI/standards-guidelines/aria/>

Open Geospatial Consortium. (u.å.). *Styled Layer Descriptor*. Open Geospatial Consortium. Hentet 21. februar, 2024, fra <https://www.ogc.org/standard/sld/>

Postman. (u.å.). *Postman*. Postman API Platform | Sign Up for Free. Hentet 7. mai, 2024, fra <https://www.postman.com/>

Productplan. (u.å.). *What is MoSCoW Prioritization? | Overview of the MoSCoW Method*. ProductPlan. Hentet 10. april, 2024, fra <https://www.productplan.com/glossary/moscow-prioritization/>

Python. (14. mai 2024.). *The Python Standard Library — Python 3.12.1 documentation*. Python Docs. Hentet 31. januar, 2024, fra <https://docs.python.org/3/library/index.html>

Python. (14. mai 2024). *smtplib — SMTP protocol client — Python 3.12.3 documentation*. Python Docs. Hentet 14. mai, 2024, fra <https://docs.python.org/3/library/smtplib.html>

QGIS. (u.å.). *Discover QGIS*. QGIS. Hentet 14. mars, 2024, fra <https://qgis.org/en/site/about/index.html>

Ribgy, D., Sutherland, J., & Takeuchi, H. (mai, 2016). *Embracing Agile*. Harvard Business Review. Hentet 31. januar, 2024, fra <https://hbr.org/2016/05/embracing-agile>

- Rojas, J. (14. september, 2023). *Learn How to Use Sketching as an Ideation Method*. The Interaction Design Foundation. Hentet 9. mai, 2024, fra <https://www.interaction-design.org/literature/article/etch-a-sketch-how-to-use-sketching-in-user-experience-design>
- Rolstadås, A. (14. april, 2020). *prosjektstyring – Store norske leksikon*. Store norske leksikon. Hentet 5. mars, 2024, fra <https://snl.no/prosjektstyring>
- Rossum, G. v., Warsaw, B., & Coghlan, A. (9. desember 2023). *PEP 8 – Style Guide for Python Code*. Python PEP. Hentet 14. mai, 2024, fra <https://peps.python.org/pep-0008/>
- Sajith. (10. juli, 2023). *Understanding the Four Types of Quality Assurance for Exceptional Results*. Medium. Hentet 10. mai, 2024, fra <https://medium.com/@sajiveva1112000/understanding-the-four-types-of-quality-assurance-for-exceptional-results-bef3168060fc>
- Sandnes, F. E. (2022). *Universell utforming av IKT-systemer* (3rd ed.). Universitetsforlaget AS.
- Scrum Alliance. (u.å.). *Acceptance Criteria: Everything You Need to Know Plus Examples*. Scrum Alliance Resources. Hentet 10. mai, 2024, fra <https://resources.scrumalliance.org/Article/need-know-acceptance-criteria>
- Staiano, F. (2022). *Designing and Prototyping Interfaces with Figma: Learn Essential UX/UI Design Principles by Creating Interactive Prototypes for Mobile, Tablet, and Desktop*. Packt Publishing, Limited. <https://books.google.no/books?id=GOBeEAAQBAJ&lpg=PP1&dq=prototyping%20tools%20for%20user%20interface%20design%20figma&lr&pg=PP1#v=onepage&q&f=false>
- Sutherland, J., & Schwaber, K. (november, 2020). *Scrum Guides*. Scrum Guide. Hentet 31. januar, 2024, fra <https://scrumguides.org/scrum-guide.html>
- Svardal, F. (27. mai 2020). *sosial loffing – Store norske leksikon*. Store norske leksikon. Hentet 13. mai, 2024, fra https://snl.no/sosial_loffing
- Swarts, E. (29. september, 2015). *Internal vs External Quality of Software*. Made Tech. Hentet 10. mai, 2024, fra <https://www.madetech.com/blog/internal-vs-external-quality-of-software/>

Techtarget. (u.å.). *What is Unit Testing? Definition fra WhatIs.com*. TechTarget. Hentet 10. april, 2024, fra <https://www.techtarget.com/searchsoftwarequality/definition/unit-testing>

Tidemann, A. (16. mai, 2023). *kunstig intelligens – Store norske leksikon*. Store norske leksikon. Hentet 26. januar, 2024, fra https://snl.no/kunstig_intelligens

Tietoevry. (u.å.). *Tietoevry skaper meningsfull teknologi som bidrar til å gjøre verden bedre*. Tietoevry. Hentet 8. februar, 2024, fra <https://www.tietoevry.com/no/om-oss/om-tietoevry/>

Universitetet i Bergen. (9. oktober. 2023). *Geografiske informasjonssystemer (GIS), kartografi og fjernmåling | Institutt for geografi | UiB*. Universitetet i Bergen. Hentet 26. januar, 2024, fra <https://www.uib.no/geografi/39289/geografiske-informasjonssystemer-gis-kartografi-og-fjernmA5ling>

University of St Andrews. (u.å.). *Code standards - Digital standards*. University of St Andrews. Hentet 8. februar, 2024, fra <https://www.st-andrews.ac.uk/digital-standards/code-standards/>

Visual Paradigm. (u.å.). *What is Activity Diagram? - When to Use Activity Diagram - Learn by Examples*. Visual Paradigm. Hentet 26. april, 2024, fra <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Visual Paradigm. (u.å.). *Why UML Modeling? - Importance of Modeling*. Visual Paradigm. Hentet 25. mars, 2024, fra <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/why-uml-modeling/>

Visual Studio Code. (5. februar, 2024). *Visual Studio Code Frequently Asked Questions*. Visual Studio Code. Hentet 30. januar, 2024, fra <https://code.visualstudio.com/docs/supporting/faq>

Voženílek, V. (2009). Artificial intelligence and GIS: mutual meeting and passing. *2009 International Conference on Intelligent Networking and Collaborative Systems*. IEEE. 10.1109/INCOS.2009.83

W3Schools. (u.å.). *What is Bootstrap*. W3Schools. Hentet 31. januar, 2024, fra https://www.w3schools.com/whatis/whatis_bootstrap.asp

WAVE. (u.å.). *Wave Documentation*. Wave. Hentet 7. april, 2024, fra <https://wave.webaim.org/api/docs?format=html>

Wave. (u.å.). *WAVE Web Accessibility Evaluation Tools*. WAVE Web Accessibility Evaluation Tools.

Hentet 19. april, 2024, fra <https://wave.webaim.org/>

Web Accessibility Initiative. (juli, 2005). *WCAG 2 Overview | Web Accessibility Initiative (WAI)*. W3C.

Hentet 19. februar, 2024, fra <https://www.w3.org/WAI/standards-guidelines/wcag/>

WebAIM - Web Accessibility in Mind. (9. april, 2020). *Introduction to WAVE*. Youtube. Hentet 14. mai,

2024, Hentet fra <https://www.youtube.com/watch?v=ITUDiTgAZY0&t=214s>

Vedlegg

Vedlegg A - Gruppekontrakt

Bachelorprosjekt 2024

Gruppens deltakere:

Victor Bakken
Tørres Andreas Båshus
Larisa Saitova
Gustav Svartsund
Cassandra Hartvedt Vårdal

Ved første prosjektmøte 10.01.2024 ble gruppens medlemmer enige om følgende:

Mål: Målet for prosjektet er å gjennomføre et prosjekt av høy kvalitet som fører til toppkarakter på bacheloroppgaven.

Gruppens regler:

- Alle avgjørelse skal gjøres i felleskap
- Ved uenighet skal avstemning benyttes. Flertallet bestemmer.
- Alle må delta aktivt.
- Alt produsert materiale skal bli kvalitetssikret og korrekturlest av et annet gruppemedlem (kode og tekst)
- Dersom vi har problemer/spørsmål skal vi ikke vente på et møte
- Discord er hovedplattform for kommunikasjon.
- Møteplikt på alle møter. Unntak gis ved gyldig fravær som er godkjent av gruppens øvrige deltakere.
- Hvis man sitter fast på en oppgave, eller ikke kan levere til oppsatt tid, skal de øvrige gruppemedlemmene informeres så tidlig som mulig slik at hjelp kan oppsøkes
- Kun produsere originalt innhold. Plagiat er uakseptabelt
- Alt innhold må følge akademiske regler. Kildehenvisninger følger APA 7 standarden
- Alle gruppemedlemmer er pliktige til å delta (fysisk eller digitalt) på alle daily scum mellom 8-16 uavhengig av fravær. Kan bruke skjønn ved spesielle forhold
- Hvert enkelt gruppemedlem plikter å fullføre tildelt oppgaver til oppsatt tid. Dersom dette ikke lar seg gjøre så må de øvrige i gruppens informeres så tidlig som mulig

- Gruppens medlemmer forventes å jobbe med bachelorprosjektet mellom kl. 08 - 16 på plass, om ikke annet er avtalt med gruppens medlemmer.
- Brudd på denne kontrakten vil medføre en muntlig eller skriftlig advarsel til vedrørende fra resten av gruppen og veileder/emneansvarlig vil bli informert
- Fravær uten grunn er ikke gyldig fravær og medlemmet skal få advarsel
- Alle advarsler dokumenteres
- Ved fjerde gangs advarsel blir den advarte ekskludert fra gruppen.
- Bruk av ChatGPT eller andre former av kunstig intelligens skal holdes strengt utenfor sluttrapporten sine rammer.
- Gjentatte overtredelser kan medføre at vedrørende gruppemedlem ekskluderes fra gruppen

Ansvarsfordeling som følger:

- Tørres Andreas Båshus - Scrum master, prosjektleder & utvikler
- Victor Bakken - Utvikler
- Larisa Khasanbekovna Saitova - Utvikler
- Gustav Svartsund - Utvikler
- Cassandra Hartvedt Vårdal - Utvikler

Prosjektleder har ansvar for:

- Å kalle inn til møter
- Delegering av oppgave

Victor Bakken

Victor Bakken

Tørres Andreas Båshus
Tørres Andreas Båshus

Larisa Saitova

Larisa Saitova

Gustav Svartsund

Gustav Svartsund

Cassandra H. Vårdal

CASSANDRA HV

Sted og dato: Kristiansand, 10.01.2023

Vedlegg B - Utdrag fra kapasitetsplan

Sprint 8																
Teamets tilgjengelighet april/mai																
Allokering	Navn	Rolle	Man	Tirs	Ons	Tors	Fre	Lør/Søn	Man	Tirs	Ons	Tors	Fre	Lør/Søn	Totalt antall timer	
			22	23	24	25	26	27/28	29	30	1	2	3	4/5		
100%	Victor	Utvikler	6,5	6,5	0	6,5	5,5		6,5	0	0	6,5	5,5		43,5	
100%	Gustav	Utvikler	6,5	6,5	0	6,5	5,5		6,5	0	0	6,5	5,5		43,5	
100%	Larisa	Utvikler (UX/UI)	6,5	6,5	0	6,5	5,5		6,5	0	0	6,5	5,5		43,5	
100%	Cassandra	Utvikler	6,5	6,5	0	6,5	5,5		6,5	0	0	6,5	0		38	
100%	Tørres	Prosjektleder	6,5	4	0	7,5	4		7,5	0	0	7,5	4		41	
Totalt antall timer per dag			32,5	30	0	33,5	26		33,5	0	0	33,5	20,5			
Totalt antall utviklingstimer tilgjengelig for denne sprinten (ekskludert planlegging)															177	
Planlegging															32,5	
Totalt (inkludert sprintplanlegging)															209,5	

Merk: 30 april er innleveringsfrist for IS-305, demomøte torsdag 2. mai.

Notat:

Sprint 9																
Teamets tilgjengelighet mai																
Allokering	Navn	Rolle	Man	Tirs	Ons	Tors	Fre	Lør/Søn	Man	Tirs	Ons	Tors	Fre	Lør/Søn	Totalt antall timer	
			6	7	8	9	10	11/12	13	14	15	16	17	18/19		
100%	Victor	Utvikler	7,5	7,5	6,5	0	0		7,5	7,5	7,5	4	0		48	
100%	Gustav	Utvikler	6,5	6,5	6,5	0	5,5		7,5	7,5	7,5	4	0		51,5	
100%	Larisa	Utvikler (UX/UI)	6,5	6,5	6,5	0	5,5		7,5	7,5	7,5	4	0		51,5	
100%	Cassandra	Utvikler	6,5	6,5	6,5	0	5,5		7,5	7,5	7,5	4	0		51,5	
100%	Tørres	Prosjektleder	6,5	7,5	4	0	4		7,5	7,5	4	4	0		45	
Totalt antall timer per dag			33,5	34,5	30	0	20,5		37,5	37,5	34	20	0			
Totalt antall utviklingstimer tilgjengelig for denne sprinten (ekskludert planlegging)															150	
Planlegging															33,5	
Totalt (inkludert sprintplanlegging)															183,5	

Merk: Innlevering av bachelorrapporten fredag 16.05 kl. 12:00

* **MERK.** Hele dokumentet kan hentes [her](#).

(https://docs.google.com/spreadsheets/d/1o2IODmhBtp9dieZnmU7c-7HUVL_e3vwozn-32YPc_l/edit?usp=sharing)

Vedlegg C - Utdrag fra loggførte timer

Tid utviklingstimer estimert og loggført sprint 2						
	Estimert	Brukt	Overestimering		Underestimering	
Backend	122	67	55	45,08%		
Frontend	0	0				
Brukertesting	32	27	5	15,63%		
Prototyping	0	9			9	900,00%
Rapportskriving	15	27			12	80,00%
Administrasjon	5	4	1	20,00%		
Kompetanseheving	32,5	65			32,5	100,00%
Total	206,5	199	61		53,5	

Tid utviklingstimer estimert og loggført sprint 3						
	Estimert	Brukt	Overestimering		Underestimering	
Backend	80,5	57,5	23	28,57%	0	0,00%
Frontend	0	0	0		0	
Brukertesting	2	1	1	50,00%	0	0,00%
Prototyping	42	24	18	42,86%	0	0,00%
Rapportskriving	4	4	0	0,00%	0	0,00%
Administrasjon	61	57	4	6,56%	0	0,00%
Kompetanseheving	32,5	14,5	18	55,38%	0	0,00%
Total	222	158	64		0	

* **MERK.** Hele dokumentet kan hentes [her](https://docs.google.com/spreadsheets/d/1AoIJ9GU7W19O3wOoE68S3dY0aI_oVPuIONdJbCra1dE/edit?usp=sharing).

(https://docs.google.com/spreadsheets/d/1AoIJ9GU7W19O3wOoE68S3dY0aI_oVPuIONdJbCra1dE/edit?usp=sharing)

Vedlegg D - Utdrag fra risikomatrise

Risikomoment	Konsekvensgrad (1-4)	Sannsynlighetsgrad før tiltak (1-4)	Risikovurdering før tiltak	Forebyggende tiltak	Konsekvensgrad etter tiltak (1-4)	Sannsynlighetsgrad etter tiltak (1-4)	Restrisiko etter tiltak	Håndtering ved inntruffet risiko
Manglende kommunikasjon mellom gruppemedlemmer og/eller produkteier Manglende kommunikasjon kan føre til misforståelser mellom gruppemedlemmer/produkter som fører til feilbeslutninger i prosjektet, senket fremdrift, dårlig allokering av ressurser og at sluttresultatet ikke møter fastsatte kravspesifikasjoner og forventet kvalitet. Dårlig kommunikasjon i gruppen kan og skape konflikter og eller uvillighet til å hjelpe eller snakke med hverandre som kan negativt påvirke	4	3	12	Opprettelse av kanaler for kontinuerlig kommunikasjon mellom medlemmene og produkteier for å dele ressurser, publisere avtaler og stille spørsmål. Daily scrum blant gruppemedlemmer for å holde alle oppdatert på hverandres fremdrift. Planleggingsmøter for kommende sprint i fellesskap med produkteier for å tydeliggjøre forventninger til resultat og riktig prioritering av oppgaver for sprinten.	4	1	4	Identifisere hvilke metoder som fungerer dårlig og endre disse (er det dårlig kommunikasjon på fysiske møter, på discord eller i møter på teams?) Tilrettelegge for bedre kommunikasjonsmetoder Ha en ærlig og åpen diskusjon dersom noen opplever problemer i gruppen.

* **MERK.** Hele dokumentet kan hentes [her](#).

(https://docs.google.com/document/d/1iCB2mhPdNdMBg6V-0Rflk_SYjgdJZ0zfHEFlcs54Oqs/edit?usp=sharing)

Vedlegg E - Utdrag med reviderte roadmaps

Veikartet per 02.04 (sprint 6)	Presprint	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Eksamen
Kursstart												
Forståelse og kunnskap												
Webløsning												
Bruke SLD mot WMS i stedet for PostGIS												
Utvide med andre typer data og klasser												
Se på det å bruke FGB og COG												
Dokumentasjon & diverse												
Implementere NGIS + Diverse												
Frontend												
Brukertesting												
Prototyping												
Overlevering og Innlevering												
Forberede muntlig eksamen												
Rapportskriving												
Frontend												
Backend												
Modellering												
Analyse												
Få kartAi repo til å fungere												

* **MERK.** Hele dokumentet kan hentes [her](#).

(https://docs.google.com/spreadsheets/d/1Vw1Awo5GVhC2iidWtnwA5R1qcV9eKDQOR_nUTP1QoU0/edit?usp=sharing)

Vedlegg F - Utdrag fra sprint dagbok

Sprint 5 Dato: 12.03.2024

Victor Bakken

- Gjennomført sprintplanlegging og utforsket COG funksjonalitet.
- Delta i geomatikk kurs og begynne å skrive python kode for å henter ned spesifikke områder
- Ingen

Gustav Svartsund

- Gjennomført sprintplanlegging og sett på videoer om gdal og cog
- Delta på geomatikk kurs og begynne på rapportskrivningen
- Ingen

Larisa Saitova

- Gjennomført sprintplanlegging og kompetanseheving innen bootstrap
- Delta på geomatikk kurs og lage funksjon for å slette merket kartområde
- Ingen

Cassandra Vårdal

- Hva har du gjort: Gjennomført sprintplanlegging, kompetanseheving for FGB og begynt å skrive kode som henter ned FGB data og konverterer det til tif filer
- Delta på geomatikk kurs og fortsette med FGB kode
- Ingen hindring

Tørres Båshus

- **Siden sist:** Gjennomført sprintplanlegging og jobbet med html-sidene.
- **Plan for idag:** Delta på geomatikk kurs og fortsette med utvikling av html-sidene
- **Hindringer:** Ingen

Øvrige notater: Ingen

* *MERK. Hele dokumentet kan hentes [her](https://docs.google.com/document/d/16cyN6FUMrSegt0lr5cWVIGyLXIhb5x547Wn6qb5C520/edit?usp=sharing).*

(<https://docs.google.com/document/d/16cyN6FUMrSegt0lr5cWVIGyLXIhb5x547Wn6qb5C520/edit?usp=sharing>)

Vedlegg G - Utdrag fra sprintplaner

Sprint 7

Mål:

- NGIS
- Mail
- Metadata
- Satelittfoto
- Fikse detaljer på frontend
- Gjennomføre brukertester

Behov:

- GDPR - får i orden hvem som er ansvarlig og etablere kontaktpunkt
- Ivar Oveland om satelittfoto

210 utviklingstimer tilgjengelig.

Arbeidsoppgaver:

Backend (79 Timer)

NGIS - PostGIS

- Hente ned data fra NGIS (Se demo.py fra i fjor) - (6 timer)
- Konvertere data til tif eller png (Se FGLabel.py fra i år) - (2 time)

MAIL

- Legge til input mail (constraints.html) (2 timer)
- Sy sammen fastAPI (Legge til send mail funksjonalitet på slutten av generate photos). (2 time)
- Sjekke at det funker - bugfixing (4 timer)
- Lage mail til dette (1 time)

Metadata

- Legge til support for metadata lasting i kode (4 timer)
- Skrive kode som sender metadata i dataset.zip (4 timer)

**MERK. Hele dokumentet kan hentes [her](#).*

(<https://docs.google.com/document/d/1s-K-HZU9vzgYSi61NLb6uWvg-xHzu75B5fLZyXnXPC4/edit?usp=sharing>)

Vedlegg H - Utdrag fra beslutningsoversikt

Generelt			
Tittel	Beslutning	Begrunnelse	Bakgrunn
Risikomatrise	Valgte å ha en risikomatrise for å håndtere risikoer	En risikomatrise er en oversiktlig måte å definere risikoer og hvordan de skal håndteres.	Utviklingsprosjekt med flytende scope og såpass mange "oppdragsgivere"
Kodestandard	Valgte å implementere en kodestandard i koden	Koden måtte gjøres mye mer oversiktlig, samt ha support for flere datakilder.	Koden i prosjektet som gruppen arbeider videre på er ikke dokumentert med få om noen dokumentarer i koden. Gruppen har brukt mye tid på å forstå applikasjonen og hvordan den fungerer.
Gruppekonsent	Gruppen valgte å ha en gruppekonsent	Det er viktig at gruppens medlemmer er enige seg imellom om hvordan man arbeider sammen, hvilke forpliktelser man, hvilke regler som styrer samarbeidet og eventuell konsekvenser for overtredelser av ett eller flere punkter i gruppekonsenten	
Kommunikasjonsplan	Gruppen valgte å ha en kommunikasjonsplan	For å gjøre kommunikasjon tydelig og saklig, men fremdeles ha rom for litt tull (som skaper god stemning innad gruppen)	
Beslutningsdokument	Gruppen valgte å ha et dokument over alle beslutninger som har blitt tatt	Det er mange beslutninger som blir tatt under prosjektets gang og det er lett å glemme hva man har besluttet om man ikke har det skrevet ned et sted. Et beslutningsdokument gir en oversikt over alle beslutninger som er blitt tatt og de blir lagret skriftlig så at de ikke blir glemt.	
Flytende scope	Det ble besluttet sammen med produkteier å ha et flytende scope	Oppdragsgiver forsto det var mye å sette seg inn i, og ville hjelpe gruppen med å bidra til et greit oppdragsforløp.	

**MERK. Hele dokumentet kan hentes [her](#).*

(https://docs.google.com/document/d/1kw_8nEuNnth_1v8cqRV_0bLIIV0kgJbv6Ko1d4WbDo/edit?usp=sharing)

Vedlegg I - Utdrag fra brukerhistorier

Brukeren: Alle som har interesse av å bruke kartdata til å trene opp en AI-modell og øvrige geomatikk

Behovet: Behovet er å trene opp en kunstig intelligens på kartdata

Fordelen: Med denne applikasjonen vil brukeren få kartdata fra ønsket område til å bli brukt for å trene opp en kunstig intelligens.

Kartdata som kan brukes til å trene opp kunstig intelligens er ikke lett tilgjengelig. Denne applikasjonen gjør dette enkelt for brukeren.

1. Distribudere applikasjonen på nett via Microsoft Azure
2. Legge til nye datakilder for WMS, COG, FGB og NGIS
3. Refaktorere eksisterende kode til noe som er mer forståelig og lettere å videreutvikle
4. Forbedre brukervennligheten for applikasjonen

Tittel:	Tilgang på nett
Behov:	Som bruker vil jeg kunne få tilgang på applikasjonen gjennom nettleseren min uten å måtte laste ned og kjøre prosjektet lokalt.
Kriterier:	• Applikasjonen er tilgjengelig gjennom en url • Applikasjonen fungerer på nett uten at en bruker må tukle med tekniske detaljer.
Fordel:	Tilgjengeliggjør applikasjonen for brukere på en måte som ikke krever at de må laste ned applikasjonen via github.

**MERK. Hele dokumentet kan hentes [her](#).*

(<https://docs.google.com/document/d/13aAiG-oBLfZCRk9xFelWxxBKCyhXbdM41C7J5vICLv4/edit?usp=sharing>)

Vedlegg J - Brukertest mal

Brukertest

Brukerhistorie:

Som bruker av denne applikasjonen så vil du markere et område på et kart ved å tegne en polygon eller en firkant som skal brukes som treningsdata for Ai.

Utfør følgende oppgave:

1. *Forstår du forsiden?*
2. *Velg metode? (du skal tegne)*
3. *Marker området (polygon eller firkant) som du vil bruke som treningsdata*
4. *Fyll ut verdiene*
5. *Last ned data til din lokale pc*

Spørsmål

1. *Var applikasjonen forståelig?*
2. *Hva var bra med applikasjonen?*
3. *Hva var mindre bra med applikasjonen?*
4. *Hva savner du/kunne ønsket var på applikasjonen?*

Vedlegg K - Utdrag fra brukertest 1, våren 2024

Brukerhistorie:

Som bruker av denne applikasjonen så vil du markere et område på et kart ved å tegne en polygon eller en firkant som skal brukes som treningsdata for Ai.

Utfør følgende oppgave:

1. Forstår du forsiden?

Måtte forklare hva applikasjonen går ut på. Bruker forstod ikke fra forsiden hva denne gjør

2. Velg metode? (du skal tegne)

Valgte metode raskt. Forstod umiddelbart hvilken han skulle velge.

3. Marker området (polygon eller firkant) som du vil bruke som treningsdata

Tegnet et polygon lett. Bruker fant ikke med en gang ut av hvordan gå videre, komme seg til neste side.

4. Fyll ut verdiene

Brukeren hadde ingen idé om verdiene som skulle fylles ut.

5. Send filen til din egen e-post

Fylte ut feltet med en gang og trykke send knappen.

Spørsmål...

**MERK. Hele dokumentet kan hentes [her](https://docs.google.com/document/d/1Uc_NYZ7yavyn6_SvEk_-oL9JkrGh5fuCfYTf6fN3E-4/edit?usp=sharing).*

(https://docs.google.com/document/d/1Uc_NYZ7yavyn6_SvEk_-oL9JkrGh5fuCfYTf6fN3E-4/edit?usp=sharing)

Vedlegg L - Utdrag fra brukertest runde 2, våren 2024

Brukerhistorie:

Som bruker av denne applikasjonen så vil du markere et område på et kart ved å tegne en polygon eller en firkant som skal brukes som treningsdata for Ai.

Utfør følgende oppgave:

1. Forstår du forsiden?

Brukeren trykket på start-knappen før vi fikk si noe. Brukeren kom fort igang

2. Velg metode? (du skal tegne)

Brukeren er usikker på hvordan man går frem. Hun vet ikke hva hun skal velge. Brukeren forstod ikke umiddelbart at hun ikke behøver å laste ned fil.

3. Marker området (polygon eller firkant) som du vil bruke som treningsdata

Fant raskt knappen for å tegne en firkant og tegnet raskt en firkant på kartet. Brukeren hadde problemer med å finne next-knappen da man må scrolle ned for å få den synlig.

4. Fyll ut verdiene

Brukere forstår ikke siden med settings. Hadde litt problemer med å forstå hva som skal fylles ut og vet ikke helt hva verdiene står for. Til å begynne med så leser ikke brukeren info'en som ligger under spørsmålene.

Brukeren forstår ikke treningsdata eller validering

Brukeren fant fort ut av hvordan valg av farge fungerer

Ikke umiddelbart klart for brukeren at man kan velge om man vil ha bygninger, veier og broer.

Ikke klart for brukeren hva som menes under Image settings.

Brukeren uttrykker at de ikke forstår hva verdiene står for

Brukeren får ikke med seg at det er en minimum grense og maksimum grense for valgene på settings

Brukeren forstår ikke tile sizes

5. Last ned data til din lokale pc

**MERK. Hele dokumentet kan hentes [her](https://docs.google.com/document/d/1YXJCWtJZw2g33Hj5eTWv65bxTDlaQm-HB28ImZJxDXQ/edit?usp=sharing).*

(<https://docs.google.com/document/d/1YXJCWtJZw2g33Hj5eTWv65bxTDlaQm-HB28ImZJxDXQ/edit?usp=sharing>)

Konsepter

API

API står for Application Programming Interface og er en kode som brukes for å utveksle data mellom forskjellige systemer eller apper. En API-integrasjon hjelper systemer å kommunisere med hverandre – uten at mennesker er involvert. Når API-et til to systemer er koblet sammen, åpner det opp for en strømlinjet deling av data og automatiserte prosesser. (Hentet fra Visma.no)

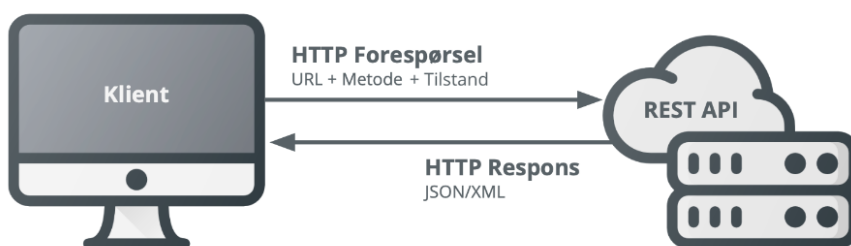
Eksempler:

- Betaling med Vipps og Klarna i nettbutikk
- Innlogging via Google eller Facebook.
- Oversikt over flyvninger og priser (for eksempel Skyscanner)
- Oppdater lagerstatus fordi nettbutikk er integrert med logistikksystemet og varelageret
- Slack integrert med Teams eller Google drive.

RESTAPI

REST – Representational State Transfer – er en programvarearkitektur som sikrer løse koblinger mellom to tjenester i en integrasjon. I dag anvendes arkitekturen først og fremst for å utvikle såkalte REST APIer. Et REST API er en skalerbar webtjeneste hvor en klient kan lese og manipulere ressurser på en tjener, som regel med HTTP som kommunikasjonsprotokoll. REST ble først introdusert for to tiår siden i en doktorgradsavhandling av Roy Fielding, som også er en av arkitektene bak HTTP. (Hentet fra NTNU.no)

Følgende figur illustrer den grunnleggende virkemåten til et REST API:



Figur 1: Den grunnleggende virkemåten til et REST API

**MERK. Hele dokumentet kan hentes [her](https://docs.google.com/document/d/1h5_p21oEkIpp850ORd-MXXUGQW14IRlgTNk6JqhRcfA/edit?usp=sharing).*

(https://docs.google.com/document/d/1h5_p21oEkIpp850ORd-MXXUGQW14IRlgTNk6JqhRcfA/edit?usp=sharing)

Vedlegg N - Low fidelity skisse

**MERK. Skissen kan hentes [her](#).*

(<https://drive.google.com/file/d/1il2zNXA6JRXg5mHzZupdW9MMfCe1eu2Y/view?usp=sharing>)

Vedlegg O - Figma prototype

**MERK. Prototypen kan hentes [her](#).*

(<https://www.figma.com/file/6njccU7H24fUKv3BgoEmgW/Prototype-NGIS-Training-Machine?type=design&node-id=0-1&mode=design>)

Vedlegg P - Instruksjon og forside

Step by step instruction

1. Choose data sources and how to draw on map.
 - You start with choosing you data sources for labels and orthography. You can also instead upload a file with metadata instead.
 - Then you choose how you want to choose area on a map
 2. Designate area to be used for training.
 - You can draw a square or a polygon on a map,
 - or you can write in the coordinates,
 - or you can upload a geoJSON-file of the area you want to use.
 3. On constraints, you fill in the parameters and settings for you training data.
 - Data Settings: Here you enter how much of the data will be used for training, how much will be used for validation and the minimum percentage of buildings. If you are unsure about which values to put in, then set them at 60% for Training, 40% for Validation and 10% for Building percentage.
 - WMS Layers: If you have chosen WMS as label source, you can choose if you want buildings, roads or bridges in the training data and you can choose which colour they will have. This is not possible for NGIS or FGB as label source.
 - Image settings: Here you can define the size of the tiles in pixels of the training data. It must be a value between 100-1104. For image resolution you can choose a value between 0.1-0.5. 0.1 is 10cm.
 - Dataset settings: Here you can choose if you want the data sent to your email and give you dataset a name. If you do not enter your e-mail, you can download the dataset directly to your computer locally.
 - When you have filled all the parameters and required fields, you can order you data by clicking on the order button and then confirm.
 - If you have not entered an email, you will after having clicked order be given the option to download the data.
-

KartAi Training Data Generator

This application generates map data in the form of tiles to train an AI model

If you wish to know more, click on "Learn more"

[Learn more](#)

[Get started!](#)

Vedlegg Q - Kodedokumentasjon

```
# Open and visualize the COG file using rasterio
with rasterio.open(cog_url) as src:
    # Loop through each smaller bounding box
    for index, bbox in enumerate(bboxes): # Enumerate the bounding boxes
        window = from_bounds(*bbox, transform=src.transform) # Create a window from the bounding box
        rgb = np.dstack([src.read(i+1, window=window) for i in range(3)]) # Read the RGB bands for the bounding box

        if rgb.size > 0:
            # Save the figure as an RGB image for each bounding box
            figure_path = os.path.join(images_directory_path, f"rgb_cog_image_{index}.png")
            plt.imshow(figure_path, rgb)
            # Generate tif file
            data_path = os.path.join(images_directory_path, f"rgb_cog_image_data_{index}.tif")
            imageio.imwrite(data_path, (rgb * 255).astype(np.uint8))
            # Print the path to the saved image
```

Vedlegg R - Selvevaluering

Victor Bakken

Under prosjektets startfase så var jeg ikke helt sikker på om dette var den korrekte utfordringen for meg og mine ferdigheter. Emnet GIS var helt nytt for alle, i tillegg var ikke python var ikke et programmeringsspråk jeg var kjent med.

Min rolle i prosjektet har primært vært backend, med noen få quick fixes på frontend om jeg oppdaget noe som måtte endres. Jeg har stått for ortofoto-kildene, god dokumentasjon, og eksperimentert i både Azure og mail sending. Etter refleksjon på tampen av prosjektets gang, så sitter jeg igjen med en positiv følelse. Det jeg personlig, og gruppen min har fått til, velger jeg å se på som et resultat av gode agile rutiner, og solid arbeid. Jeg er meget fornøyd med det vi har fått til, og det vet jeg at oppdragsgiver også er. Vi har som gruppe stått veldig selvstendig gjennom hele prosjektprosessen, selv med frie tøyler, så har vi klart å jobbe strukturert, smidig, og målrettet. Jeg tror jeg snakker for alle at prosjektet har vært ekstremt lærerikt, og vi har utført oppgavene foran oss med et smil. Avsluttende så vil jeg takke for et gøyalt og strålende halvår på kontorene til Kartverket og Tietoenvry.

Tørres Andreas Båshus

I starten var jeg forventningsfull, men også litt skeptisk overfor prosjektet da vi begav oss inn på ukjent territorium. Geomatikk, kunstig intelligens og mange av de teknologiene som vi skulle anvende oss av var nye for meg. Kompetansegapet var stort for alle. Heldigvis har vi alle klart å tilegne oss de kunnskapene som var nødvendige for å fullføre prosjektet.

I dette prosjektet hadde jeg rollene som prosjektleder, scrum-master, UX/UI-designer og frontend-utvikler. Jeg hadde mange ulike hatter og multitasket gjennom hele bachelorprosjektet. Jeg stod for alt det administrative, var kontaktpunkt utad og innad, var scrum-master, holdt i alle trådene, gjennomførte brukertester, laget prototype, kodet på frontend, satte meg inn i GDPR, skrev policier, skrev innhold, tok initiativ til at vi skrev tester og mye annet.

Gruppen har fungert over all forventning. Jeg er stolt at vi løste prosjektet så selvstendig og smidig som vi gjorde, og at det var gruppen som var initiativtaker til alle de ulike oppgavene som ble utført i prosjektet.

Det har vært et spennende og lærerikt halvår hos Kartverket og Tietoenvry.

Larisa Saitova

Etter at vi fikk tildelt oppgaven fra Kartverket var jeg ganske bekymret for at vi ikke ville greie å gjennomføre dette prosjektet med god nok kvalitet. Ingen i gruppen hadde erfaring med geomatikk og vi hadde et svært kompetansegap å overkomme. Derfor satte jeg stor pris på muligheten til å kunne delta i omfattende geomatikk kurs som vekket sterk interesse for fagfeltet. Etter innføring i prosjektet var jeg bekymret for at mitt bidrag ville ha mindre betydning for prosjektet siden videreutviklingen i stor grad refererte til backend, mens jeg hadde

mer erfaring med frontend utvikling og UX design. Etter et dypere dykk inn i applikasjonen fant gruppen derimot et ekstremt behov for å bygge om frontend delen. Og jeg var ivrig til å påta meg dette ansvaret. I prosjektet hadde jeg ansvar for frontend utvikling av applikasjonen, og var i tillegg hovedansvarlig for UX/UI design. Dette involverte blant annet gjennomføring av all testing og analyse av frontend. I tillegg til skissering, prototyping og koding. Jeg er veldig stolt av hva gruppen har fått til å lage og hvor selvstendig vi har vært gjennom hele prosessen. Jeg vil veldig gjerne takke Tieto Evry og Kartverket, som helt fra starten har vist stort engasjement for vårt prosjekt, veiledet arbeidet og utstyrt oss med relevante ressurser. Deres kontinuerlige motivasjon og ros har motivert oss til å yte vårt beste hele veien.

Gustav Svartsund

Da jeg først fikk presentert prosjektet ordentlig hos Tietoevry med Kartverket tilstede, ble jeg veldig usikker på om dette var gjennomførbart. Det var mye ny teknologi jeg aldri har vært borte i, som Python og FastAPI, samtidig som Geomatikk var et helt nytt felt. Etter god planlegging og viktige møter med de interesserte i prosjektet gikk det opp for meg at dette kommer til å bli et lærerikt og spennende semester. Et av målene mine etter fullført utdanning var å bli en bedre utvikler, jeg ble derfor tildelt rollen som backend-utvikler og rapportansvarlig. Denne rollen passet perfekt, da jeg fikk en utfordring i det å videreutvikle en applikasjon. Erfaringen jeg fikk etter å ha måtte sette meg inn i eksisterende kode, geomatikk og utvikling samtidig som man skulle skrive ny kode var utfordrende, men nyttig. I tillegg til dette hjalp jeg litt til på frontend med små fixes og diverse. Etter disse månedene med dette prosjektet har jeg lært utrolig mye om utvikling, geomatikk, og det om man videreutvikler et prosjekt. Jeg er veldig stolt over det vi har fått til som gruppe i løpet av dette semesteret, da jeg føler vi gikk over all forventning. Samarbeidet var utrolig bra, og vi utnyttet ferdighetene til hele gruppen slik at sluttproduktet ble så bra som mulig. Dette semesteret har gått utrolig fort, og jeg vil gjerne takke Kartverket og Tietoevry for et godt samarbeid.

Cassandra Vårdal

Når vi først fikk presentert dette prosjektet var jeg litt redd for at vi hadde tatt oss vann over hodet og at vi kom til å bruke de første tre månedene kun på å lære oss domenekunnskap. Vi har derimot fått til utrolig mye og lært enda mer. I dette prosjektet har jeg hatt rollen som teknisk ansvarlig. Denne rollen har innebært å sette meg inn i hvilke teknologier som prosjektet er avhengig av og å formidle disse teknologiene til resten av gruppen. Jeg har prøvd å holde et overordnet blikk på prosjektet og bryte ned de store arbeidsoppgavene til mer gjennomførbare deler og å holde motivasjonen oppe i møte med nye teknologier og utfordringer. Prosjektet har vært et møte med utrolig mange nye teknologier som vi har måttet lære. Jeg føler at jeg endelig har fått brukt alt jeg har lært tidligere i utdanningen min i praksis på en utrolig god måte og det er kjempegøy at vi nådde alle målene vi satte oss i starten av semesteret. Jeg har vært utrolig heldig med gruppen og vi har jobbet ekstremt godt sammen, vi har hatt god og klar kommunikasjon gjennom hele prosjektperioden og jeg er utrolig stolt over sluttproduktet vi nå leverer fra oss.

Til slutt vil jeg takke Kartverket og Tietoevry som har vært utrolig gode støttespillere gjennom hele prosjektperioden og har bidratt med ressurser og veiledning når det trengs. Dette prosjektet kunne ikke blitt gjennomført uten den tette oppfølgingen og veiledningen vi har fått derfra.

Vedlegg S - Uttalelse fra arbeidsgiver



UTTALELSE FRA OPPDRAGSGIVER

Tietoenvry skaper meningsfull teknologi som bidrar til å gjøre verden bedre. Vi er et ledende teknologiselskap med en solid nordisk arv og global kapasitet. Basert på våre kjerneverdier åpenhet, troverdighet og mangfold, jobber vi sammen med våre kunder for å utvikle en digital fremtid hvor virksomheter, samfunn og mennesker kan vokse. Våre 24 000 eksperter globalt er spesialisert innen sky, data og programvare, og betjener tusenvis av kunder innen næringsliv og offentlig sektor i mer enn 90 land. Tietoenvrys årlige omsetning er på rundt 3 MRD Euro og selskapet er børsnotert på NASDAQ i Helsinki og Stockholm, i tillegg til Oslo Børs.

Kartverket skal levere landsdekkende geografisk informasjon og tjenester for private og offentlige brukere. Kartverket har ca 800 ansatte og er organisert i fire divisjoner: geodesi, land, sjø og eiendom. Vi leder og koordinerer arbeidet med den nasjonale geografiske infrastrukturen i Norge. Et viktig målområde er at Norge skal være ledende i bruk av geografisk informasjon.

Det bekreftes med dette at Larisa Khasanbekovna Saitova, Cassandra Hartvedt Vårdal, Victor Bakken, Gustav Svartsund og Tørres Andreas Båshus, fra UiA gjennomførte et bachelorprosjekt hos Tietoenvry Create Kristiansand i samarbeid med Kartverket i Agder og Norkart. Studentgruppa leverte en webbasert løsning som kunne trekke ut og klargjøre geodata (flyfoto, ortofoto mm.) fra Kartverkets systemer og gjøre dette klart for trening av kunstig intelligens som benyttes videre i KartAI prosjektet.

Teknologi

Den webbaserte løsningen som gruppen utviklet var basert på en backend i Python hvor det ble benyttet WMS med SLD, FastAPI, Azure og Geospatial Database. Frontend ble prototypet ved hjelp av bl.a. Figma og utviklet med HTML, CSS, JavaScript, Leaflet, Bootstrap for å nevne noen av de viktigste teknologiene. Gruppen har også fått hostet løsningen via Docker i Azure og integrert med SMTP slik at denne er tilgjengelig for et stort publikum. Det er også benyttet verktøy som Github, Github Projects, Visual Studio Code, Wave, Lucidchart for å få til en smidig prosjektgjennomføring.

Omfang

Hver av studentene jobbet fulle dager 4 dager i uken med prosjektet i en periode fra begynnelsen av januar til begynnelsen av juni, totalt omfang på minst 480 timer per student.

Resultat

Studentene har jobbet med teknologi og domene som i veldig stor grad var ukjent for dem ved starten av semesteret. De har jobbet målbevisst og godt og endte opp med å levere mer enn forventet i opprinnelig scope. Studentgruppa har også måtte forholde seg til et nettverk av veiledere i Tietoevry, Kartverket og Norkart, noe de har taklet bra. Alt i alt er alle parter meget godt fornøyd med samarbeidet og resultatene som studentgruppa har levert. Det kan også nevnes at studentgruppa har stilt villig opp og holdt flere presentasjoner hos Kartverket og i Tietoevry.

Vi vil gjerne takke studentgruppen for en strålende innsats og ønske alle sammen lykke til videre med eksamen, videre studier og arbeidslivet.

Med vennlig hilsen

Tietoevry Norway AS



Roar Engen

Manager

Tietoevry Create Kristiansand

Kartverket



Lars Fredrik Gyland

Fylkeskartsjef/avdelingsdirektør

Kartverket i Agder

Vedlegg T - MoSCoW analyse

MoSCoW Prioritering			
Must have	Should have	Could have	Won't have
Søkefunksjon	Filopplastingsfelt for metadata	Tutorial video	Andre språk språk
Forklaring av funksjoner	Ny landingsside	Progresjonsindikator	Text to speech
Forklaring av forventet input	Bedre kontrast	Norsk og engelsk språk	
Opplasting av gyldige filer	Tilbakeknapp	Flere koordinatsystemer	
Tilbakemeldinger fra systemet	Slette markering på kart		
Oversikt over frigitte områder	Større skrift		
Instrukser for bruk av systemet			
Kontainere for datakilder			
checkboxbokser og tekstfelt for nye funksjoner			