



Forside

IS-304: 2023

Tittel: Automatisering av dataflyt fra nasjonale kartbaser til AI-algoritmer

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen og Geir Inge Hausvik
Veileder	Niels Frederik Garmann-Johnsen
Oppdragsgiver:	Tietoevry

Studenter:

Etternavn	Fornavn
Bredesen,	Ine Mari
Ekholt,	Tobias
Fugledal,	Niklas
Skjæveland,	Kasper Johan Idland
Slettene,	Sondre

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA <u> x </u>	NEI —
Kan besvarelsen brukes til undervisningsformål?	JA <u> x </u>	NEI —
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA <u> x </u>	NEI —

Forord

Denne rapporten dokumenterer prosjektet «Automatisering av dataflyt fra nasjonale kartbaser til AI-algoritmer». Rapporten tar for seg sentrale avgjørelser, bruk av prosjektstyringsrammeverk, og hvordan prosjektet er gjennomført med tilhørende verktøy og teknologier. Avgjørelser tatt i prosjektet er baserte på relevant litteratur, erfaring og aktørenes ønsker og meninger.

En spesiell takk går til Tietoevry og Kartverket for muligheten for samarbeid i dette prosjektet, støtte og tilgang på ressurser som har vært avgjørende for prosjektets suksess.

Vi ønsker å takke våre veiledere som har vært med å bidra til et vellykket prosjekt. En stor takk går til vår veileder fra Tietoevry Roar Engen, for hans råd innenfor prosjektstyring og gjennomføring av scrum metodikken. Vi vil også rette en takk til vår veileder hos kartverket Lars Fredrik Gyland for hans rådgivning og oppmuntring gjennom prosjektet. Niels Frederik Garmann-Johnsen fortjener også vår takk for veiledning i rapportskrivning.

Sander Jyhne vil vi også takke for hans veiledning innenfor teknologier og kunstig intelligens. Vi er også takknemlige for at Mathilde Ørstavik og Alexander Nossum har bidratt med verdifull kunnskap og assistanse, som har vært avgjørende under løsningen av tekniske utfordringer. En takk rettes også mot våre emneansvarlige Hallgeir Nilsen og Geir Inge Hausvik for deres innsiktsfulle forelesninger.

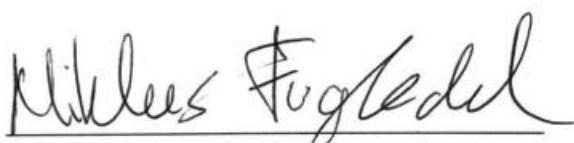
Til slutt ønsker vi å uttrykke vår takknemlighet overfor alle som har bidratt og vist engasjement til dette prosjektet. Muligheten til presentasjoner av prosjektet for Digin, Geoforum og FoU har bidratt til å holde oss fokuserte og dedikerte til prosjektet.



Ine Mari Bredeesen



Kasper Johan Idland Skjæveland



Niklas Fugledal



Sondre Slettene



Tobias Ekholt

Sammendrag

Denne rapporten dokumenterer prosessen for gjennomføringen av bachelorprosjektet "Automatisering av dataflyt fra nasjonale kartbaser mot AI-algoritmer" i samarbeid med TietoEvry og Kartverket. Prosjektet var en del av det større forskningsprosjektet KartAi. Målet med bachelorprosjektet var å utvikle et system for henting og generering av treningsdata som skulle anvendes i maskinlæring.

Ettersom at prosjektet var uforutsigbart tok gruppen i bruk prosjektstyringsmetodikken scrum. Vi benyttet scrum-aktivitetene *daily standup*, *sprint planning*, *sprint review* og *sprint retrospective*. Scrum artefaktene *sprint backlog* og *definition of done* ble også tatt i bruk. Kvalitet var høyt prioritert i prosjektet vi utførte derfor flere kvalitetssikrende tiltak, disse omhandlet kommunikasjon, dokumentasjon, følgeing av kodenstandard og utarbeiding av risikomatrise.

For at løsningen skulle få et brukervennlig design hadde vi en designprosess hvor vi laget skisser og en prototype som ble brukertestet. Under utviklingen av frontend-delen av løsningen benyttet vi HTML, CSS og JavaScript med Bootstrap og Leaflet. Vi utviklet også en backend-løsning med bruk av Python for uthenting og generering av treningsdata til bruk i KI-modeller. Prosjektet hadde en bratt læringskurve og vi møtte flere utfordringer underveis. I løpet av arbeidet har gruppen måtte håndtere tilgangsproblemer, kompetansegap og håndtering av nye teknologier og verktøy, men har til slutt klart å utvikle et produkt som overgikk gruppens, produkteier og aktørers forventninger.

En gjennomgang av den endelige løsningen kan bli sett her:

<https://youtu.be/ EqpHVukez4>

Github-repository kan bli funnet her:

<https://github.com/tobiasekholt/Bachelor>

Innholdsfortegnelse

1	Introduksjon	6
1.1	Sentrale begreper	7
1.2	Gruppen	8
1.3	Aktørene	8
1.4	Prosjektet	9
1.5	Mål og krav.....	9
2	Prosjektstyring	11
2.1	Metodikk.....	11
2.2	Kvalitet	13
3	Prosjektgjennomføring.....	15
3.1	Valg av verktøy og teknologier	15
3.2	Sammendrag av sprintene	20
3.3	Kartlegging.....	22
3.3.1	Prosjektstart og scope	22
3.3.2	Roadmap.....	22
3.3.3	Kapasitetsplan	23
3.3.4	Risikomatrise	24
3.4	Modellering.....	25
3.5	Designprosessen	27
3.5.1	Konseptutvikling	27
3.5.2	Skissering og prototyping	28
3.5.3	Designprinsipper.....	31
3.6	Evaluering/brukertesting	33
3.6.1	Forberedelse	33
3.6.2	Fremgangsmåte	33
3.6.3	Resultater fra brukertesting	34
3.7	Utviklingsprosessen	35
3.7.1	Utviklingen av webgrensesnittet	35
3.7.2	Innhenting og lagring av data	36
3.7.3	Generering og nedlastning av treningsdata	37
3.7.4	Automatisering av backend	38
3.7.5	Sammenkobling av backend og frontend	38

3.7.6	Valg av metode for programvarelevering.....	38
4	Resultatet	39
4.1	Endelig løsning.....	39
4.2	Veien videre	40
5	Refleksjon	41
5.1	Prosjektarbeid	41
5.1.1	Gjennomføring av scrum	41
5.1.2	Prosjektstyringsverktøy.....	42
5.1.3	Kvalitetssikring.....	42
5.1.4	Sosial loffing	43
5.2	Erfaringer med teknologier og utvikling	43
5.2.1	Valg av rammeverk.....	43
5.2.2	Kompetansegap	44
5.2.3	Bruk av ChatGPT som verktøy	45
5.2.4	Sentrale utfordringer	45
6	Konklusjon	47
7	Litteraturliste	48
8	Vedlegg	52

Figurliste

Figur 1. Illustrasjon av tidslinje til sprinter og hovedfokus.....	20
Figur 2. Skjerm bilde av endelig roadmap.....	23
Figur 3. Skjerm bilde av Sprint 6 i Kapasitetsplanen.....	24
Figur 4. Skjerm bilde av et moment i Risikomatriksen med tilhørende vurderinger og tiltak.....	25
Figur 5. En av de første modellene laget av dataflyten til løsningen.....	26
Figur 6. Endelig modell av dataflyten til løsningen.....	27
Figur 7. Diagram over brukerreisen til webbløsningen.....	28
Figur 8. Skisser av utarbeidet under planleggingen av designet til løsningen.....	29
Figur 9. Skjerm bilde av landingssiden i prototypen.....	30
Figur 10. Skjerm bilde av landingssiden i den endelige løsningen.....	30
Figur 11. Skjerm bilde av kort på landingssiden når man holder musepekeren over kortet for tegning på kart (venstre).....	31
Figur 12. Skjerm bilde av webbløsningen når systemet jobber med nedlastning av data.....	32
Figur 13. Skjerm bilde av tekst og infoknapper på prototypen.....	34
Figur 14. Skjerm bilde av bekreftelsesmelding i prototypen.....	34
Figur 15. Skjerm bilde av siden for å tegne utsnitt på kart i webbløsningen.....	39
Figur 16. Skjerm bilde av e-post med treningsdata mottatt etter bestilling.....	40
Figur 17. Illustrasjon av et utsnitt på kart i form av en bbox(svart) sett i forhold til et polygon(rosa).....	46

Liste over vedlegg

Vedlegg 1: Gruppekontrakt.....	52
Vedlegg 2: Risikomatrikse.....	54
Vedlegg 3: Dagbok.....	54
Vedlegg 4: Roadmap.....	54
Vedlegg 5: Prototype.....	54
Vedlegg 6: Skisser.....	55
Vedlegg 7: Data fra brukertest: Tidsbruk.....	56
Vedlegg 8: Data fra brukertest: Spørsmål.....	57
Vedlegg 9: Kravspesifikasjon.....	57
Vedlegg 10: Data fra brukertest: Oppgavesett.....	59
Vedlegg 11: Uttalelse fra oppdragsgiver.....	60
Vedlegg 12: Selvevaluering.....	61

1 INTRODUKSJON

I denne rapporten presenteres gjennomføringen av bachelorprosjektet «Automatisering av dataflyt fra nasjonale kartbaser til AI-algoritmer». Prosjektet er utført som en bacheloroppgave i IT og Informasjonssystemer ved Universitetet i Agder. I denne rapporten gjøres det rede for prosessen fra prosjektstart til endelig produkt. Denne prosessen inkluderer kartlegging, planlegging, estimering, utvikling, kvalitetssikring og anvendelse av metoder knyttet til dette.

Bachelorprosjektet var en del av forskningsprosjektet KartAi, som er et offentlig initiativ med formålet å øke kvaliteten i Norges offisielle eiendomsregister (matrikkelen) og Sentral Felles Kartbase (SFKB) ved bruk av kunstig intelligens i regi av Kartverket, Norkart, og Kristiansand kommune (Nossum et al., 2021). Målet for bachelorprosjektet var å utvikle et system for henting og generering av treningsdata som skulle anvendes i maskinlæring. Med formålet om å oppnå dette, gjennomgikk vi en designprosess og implementerte en backend-løsning som kunne hente og generere treningsdata. Videre utviklet vi en frontend-webløsning for brukere å angi koordinater og bestille ønsket treningsdata.

Innledningsvis vil vi definere sentrale begreper benyttet i rapporten, presentere involverte parter samt målene og kravene til prosjektet. Videre vil vi ta for oss prosjektstyringen og nøye beskrive prosessen for gjennomføringen av prosjektet. Avslutningsvis vil vi reflekteres rundt gjennomføringen av prosjektet hvor vi tar for oss ulike oppgaver i prosjektet, problemstillinger, prosjektstyring, systemer og dokumentasjon.

1.1 SENTRALE BEGREPER

Dette delkapittelet vil definere og forklare ulike tekniske begreper og teknologier som blir omtalt i rapporten. Formålet er å gi en bedre forståelse av de ulike uttrykkene og konseptene som brukes i sammenheng med prosjektet, for å sikre at leseren får en god forståelse av rapporten

Kunstig intelligens (KI)- systemer som utfører fysiske eller digitale handlinger, basert på tolkning og behandling av data, med hensikt å oppnå et gitt mål (European Commission, 2019). Maskinlæring er en form for kunstig intelligens, og dette er en viktig del av dette prosjektet, ettersom at prosjektet omhandler uthenting og bruk av treningsdata til maskinlæring.

API (Application Programming Interface) et grensesnitt som gir direkte tilgang til data og funksjonalitet i et datasystem, og gjør det enklere for et system eller en tjeneste å kommunisere med datasystemet (Rossen & Nätt, 2022).

NGIS-OpenAPI- NGIS-OpenAPI er et API som gir tilgang til uthenting av geografiske data gjennom nasjonalt geografisk informasjonssystemer (NGIS) som er en forvaltningsplattform av oppdaterte databaser hos Kartverket (Kartverket, 2023).

Sentral felles kartdatabase (SFKB)- er en samling av datasett med detaljerte kartdata i form av vektordata, som ble benyttet aktivt gjennom prosjektet (Granum, 2020).

Ortofoto- er flybilder med samme geometriske egenskaper som et kart. I tillegg kan ortofoto gi informasjon om landoverflaten som et kart ikke gir (eksempel farger, mønstre og tekstur) og blir derfor benyttet i økt grad til presentasjon av geografisk informasjon (Mæhlum, 2021).

GeoJSON- en form av JavaScript object notation (JSON) brukt til håndtering av stedfestet informasjon, som følger et standardisert format (Esri, u.å).

Bounding Box (bbox)- er en rektangulær boks som omringer et objekt, som i hovedsak brukes til gjenkjenning av objekter, der selve målet er å identifisere et objekt og dets posisjon, for eksempel ved bruk av koordinater (Subramanyam, 2021).

Geografisk informasjonssystem (GIS)- er en programvare for innsamling, organisering, lagring, analyse og visualisering av geografisk informasjon (Ørstavik & Mæhlum, 2022).

Web map service (WMS)- er en internasjonal standard for utveksling av elektroniske kart gjennom The Hypertext Transfer Protocol (HTTP) spørringer for et gitt område mot en server. Spørringen som sendes til serveren kan inneholde området i form av

koordinatpunkter, hvilket kartlag en ønsker å hente, og hvilket format output skal være i (Mæhlum, 2023).

European Petroleum Survey Group (EPSG)- er en organisasjon som har definert en rekke koordinatsystemer som blir brukt i GIS og geodata. Det finnes en rekke forskjellige koordinatsystemer som er numerisk kode som blir kalt for EPSG-kode. Disse blir brukt til å referere til et spesifikt koordinatsystem som gjør at den dataen som blir brukt er riktig referert når den blir brukt i applikasjoner for visualisering av kart (EPSG.io, u.å). I prosjektet vil begrepet EPSG bli brukt til å beskrive diverse koordinatsystemer som er blitt tatt i bruk.

1.2 GRUPPEN

Gruppen består av fem studenter som er inne i sitt siste semester av bachelor-studiet IT og informasjonssystemer ved Universitetet i Agder. Flere av gruppemedlemmene har jobbet med hverandre tidligere i studiet og har hatt flere gode erfaringer sammen. Under bachelorstudiet har gruppen opparbeidet seg felles og variert kompetanse innenfor flere fagområder. Dette har ført til at gruppen kjenner hverandres styrker og har en allerede godt utviklet gruppedynamikk. For prosjektet sin del har det betydd effektiv fordeling av arbeidsoppgaver, hyggelige og lærerike diskusjoner og et sluttprodukt vi er stolte av. For å sikre ekstra kvalitet og samhold i gruppen er det også utviklet en gruppekontrakt som finnes vedlagt i Vedlegg 1.

1.3 AKTØRENE

Tietoevry er bedriften gruppen gjennomførte bachelorprosjektet med. Tietoevry er et ledende internasjonalt konsulentselskap som jobber med utvikling av systemer og tjenester til kunder i 20 forskjellige land (Tietoevry, u.å). Hos bedriften fikk gruppen tildelt en veileder og fikk benytte seg av kontorlokalene. Bacheloroppgaven var å gjennomføre et oppdrag for en kunde- Kartverket. Kartverket er statens fagorgan for kart, geodata, og eiendomsinformasjon. Bedriften har ansvar for å bygge og vedlikeholde en nasjonal geografisk infrastruktur for sjø og land (Regjeringen, u.å.u.å). Hos Kartverket var produkteier Sander Jyhne og en veileder som støttet gruppen under prosjektet. Her fikk gruppen også benytte seg av kontorlokaler og grupperom.

I tillegg til de to hovedaktørene fikk gruppen også mye støtte og teknisk veiledning fra Norkart. Norkart er et norsk teknologiselskap som tilbyr løsninger innenfor kommunalteknikk, kart og eiendomsinformasjon til offentlig og privat sektor (Norkart, u.å). Gjennom bedriften fikk gruppen teknisk veiledning og tilgang til eksisterende kode knyttet til KartAi-prosjektet.

Under første oppstartsmøte hos Tietoevry ble vi introdusert til vår partner og veileder i Tietoevry: Roar Engen, produkteier fra Kartverket: Sander Jyhne, Lars Fredrik Gyland som veileder i Kartverket og Alexander Nossun som er vår teknologiske veileder fra Norkart.

1.4 PROSJEKTET

Bachelorprosjektet er en del av KartAi-prosjektet i regi av Kartverket, Norkart, og Kristiansand kommune. KartAi er et offentlig forskningsprosjekt som skal øke kvaliteten i Norges offisielle eiendomsregister (matrikkelen) og Sentral Felles Kartbase (SFKB) ved bruk av kunstig intelligens (Nossun et al., 2021).

Forskningsprosjektet har tre delmål også kalt arbeidspakker, vårt prosjekt er knyttet til Arbeidspakke 1. Denne arbeidspakken handler om å gjøre det mulig å identifisere bygninger ved hjelp av flere ulike datakilder. Dette gjøres ved å lage en «geodatasjø» med optimaliserte datakilder for bruk i kunstig intelligens og stor dataanalyse.

De andre to arbeidspakkene handler om saksbehandling, hvor Arbeidspakke 2 handler om å lage et bedre beslutningsgrunnlag for kommunal saksbehandling ved å indentifisere bygning under en viss størrelse i Kristiansand med å anvende og skape nye KI-algoritmer fra geodatasjøen omtalt i delmål 1. Arbeidspakke 3 handler om å sikre en mer effektiv saksbehandling gjennom automatisert, pålitelig og etisk kvalitetsheving av den nasjonale matrikkelen og kartbaser ved proaktiv involvering av innbyggere(Nossun et al., 2021).

1.5 MÅL OG KRAV

Målet med dette prosjektet er å automatisere og forenkle uthenting av treningsdata til bruk i maskinlæring. I dag er uthenting av treningsdata en kompleks og tidkrevende prosess som også er tilgangsavhengig. Ved å automatisere og effektivisere denne prosessen vil flere kunne hente treningsdata basert på bygningsdata og flyfoto til bruk i maskinlæring. Ved bruk av en selvbetjeningsportal vil prosessen være mer brukervennlig og omfanget av personer som kan utføre prosessen kan bli større på sikt.

Gruppen fikk ingen kravspesifikasjon fra produkteier. Kravene ble laget av gruppen selv og ble basert på informasjon gruppen fikk underveis. Ut ifra hvert krav ble det laget en brukerhistorie som ble grunnlaget for akseptansekriterier gruppen satte opp. Akseptansekriteriene er en liste over egenskaper som kreves for at et produkt blir akseptert (McKenna, 2016, s. 77) For at et krav skulle være oppnådd måtte alle akseptansekriteriene til kravet bli møtt.

Et eksempel på dette er kravet «Tegne på kart», som fikk brukerhistorien: «Som en bruker vil jeg kunne tegne et polygon/en firkant på et kart, for å spesifisere området jeg vil ha treningsdata for.». Ut ifra denne brukerhistorien ble det satt opp følgende akseptansekriterier:

- ✓ Det er en side som viser et kart
- ✓ Brukeren kan tegne en polygon eller firkant på kartet
- ✓ Koordinatene hentes ut fra det tegnede utsnittet, lagres i systemet og vises på skjermen

Da disse ble nådd, var dette kravet oppnådd. Ved å identifisere akseptansekriterier kan man fastsette når et produkt er fullført og akseptabelt for kunden (McKenna, 2016, s. 32-33). Dermed eliminerte vi forvirring og usikkerhet rundt når produktet kunne defineres som ferdig. Samtlige krav er tilgjengelige i Vedlegg 9.

2 PROSJEKTSTYRING

Dette kapittelet presenterer metodikken som ble anvendt til prosjektstyring og hvordan gruppen sikret kvalitet under prosjektet. Prosjektstyringsmetodikken scrum blir definert, og det gjøres rede for de ulike rollene knyttet til metodikken. Aktiviteter og artefakter fra metodikken samt hvordan disse ble anvendt og gjennomført blir beskrevet. Videre presenteres ulike tiltak for å sikre kvalitet gjennom prosjektet, disse inkluderer dokumentasjon, kommunikasjon, risikomatrixe og kodenstandard.

2.1 METODIKK

Innenfor IT-bransjen er scrum en mye brukt prosjektstyringsmetodikk, spesielt egnet for agile prosjekter hvor man kan forvente endringer underveis. Metodikken inneholder flere aktiviteter og artefakter og går ut på å dele større oppgaver opp i mindre, mer håndterbare enheter. På denne måten kan man oppnå effektiv problemløsning gjennom at problemer og feil oppdages tidligere i prosessen (Karlsen, 2021, s. 277-278). Valget av scrum som prosjektstyringsmetodikk ble påvirket av flere faktorer. Dette inkluderte den antatte forekomsten av endringer underveis grunnet begrenset forståelse av prosjektets omfang ved prosjektstart. I tillegg er rammeverket kjent for å fremme samarbeid og effektiv fremdrift, noe som var viktig for å sikre suksess i prosjektet (Køster, 2023) Andre påvirkende faktorer var anbefaling fra veileder hos Tietoevry og tidligere erfaring med metodikken. Tidsrammen for sprintene ble satt til to uker grunnet usikkerhet i prosjektet og anbefaling fra veileder hos Tietoevry da dette var standard hos dem.

I scrum-rammeverket er det definert tre roller i et scrum-team; produkteier, scrum-master og utviklerne (Bibik, 2018, s. 16-17).

- **Produkteier-** I dette prosjektet har Sander Jyhne som produkteier ansvar for å samle inn og definere kravene til produktet, prioritere oppgaver og sikre at utviklerne har all nødvendig produktinformasjon (Bibik, 2018, s. 17). Videre er Jyhne også ansvarlig for å delta i *sprint planning* og *sprint review*, og gi tilbakemeldinger til teamet etter hver sprint. Gjennom å ta ansvar for disse oppgavene, vil Jyhne bidra til å sikre en smidig utviklingsprosess og en best mulig fungerende produktleveranse.
- **Scrum-master-** Har ansvar for at alle i teamet forstår og bruker scrum-rammeverket på riktig måte. I tillegg har scrum-master ansvaret for å organisere prosessen, som inkluderer å lede *daily standup*, *sprint review* og *sprint retrospective* (Bibik, 2018, s. 17). Ettersom at et av gruppemedlemmene hadde et ønske om å ta denne rollen, ble rollen fordelt deretter.
- **Utviklerne-** har i dette prosjektet ansvaret for utviklingen av selve tjenesten og sørge for å ha de ferdighetene som trengs for å kunne ferdigstille et produkt

(Køster, 2023). I prosjektet ble utviklerteamet delt inn i front- og backend-team basert på personlige egenskaper, erfaring og interesse.

Gruppen har gjennomført scrum-aktivitetene *daily standup*, *sprint*, *sprint planning*, *sprint retrospective* og *sprint review*. Dette er tidsbegrensede og gjentakende møter som definerer rammeverket og som sørger for kommunikasjon og transparens i prosjektet (Garcia et al., 2022).

- **Daily standup-** en daglig aktivitet hvor gruppen deler informasjon om arbeidet som ble utført dagen før, planlagt arbeid for dagen, samt eventuelle hindringer (Karlsen, 2021, s. 280). Gruppen gjennomførte *daily standup* hver morgen, og dokumenterte underveis i et felles Google Docs dokument (Vedlegg 3).
- **Sprint planning-** gjennomføres ved oppstarten av hver sprint, med formål å foreta forberedelser for sprinten, inkludert valg av oppgaver (Karlsen, 2021, s. 280). Under møtene deltok scrum-teamet og veileder fra Tietoevry. Produkteier bidro med ønsker om fokusområder og mål for sprinten, og veileder fra Tietoevry med tilbakemeldinger og råd for løsning av oppgavene.
- **Sprint review-** en aktivitet gjennomføres ved slutten av hver sprint, hvor utviklerne presenterer sin progresjon og fremgang (Karlsen, 2021, s. 280). På disse møtene presenterte gruppen hva som hadde blitt gjort i sprinten for produkteier. Disse møtene bidro til å sikre at gruppen holdt seg på riktig spor og rapporterte fremdriften til produkteier.
- **Sprint retrospective-** utføres etter *sprint review* og har som hensikt å forbedre sprint-prosessen (Karlsen, 2021, s. 281). Disse ble holdt internt i gruppen for refleksjon over endt sprint og identifikasjon av potensielle forbedringsområder til kommende sprint.

Teamet besluttet å ta i bruk scrum-artefaktene *sprint backlog* og *definition of done*. I *sprint backlog* ble oppgavene definert under *sprint planning* og lagt inn, og ble oppgavene for kommende sprint. Grunnet usikkerhet i prosjektet, og at kravspesifikasjon ble utviklet av gruppen underveis i prosjektet (Vedlegg 9), var det vanskelig å lage en *product backlog*, som er en overordnet oversikt over ideer som skal utvikles (Schwaber & Sutherland, 2020, s. 10-11). Derfor valgte gruppen heller å utvikle en grundig *sprint backlog* for hver sprint, ettersom at oppgavene for kommende sprint ble definert under *sprint planning*. I tillegg laget gruppen som tidligere nevnt en oversikt over krav og akseptanskriterier for hvert krav. Akseptanskriteriene bidro til å fastsette *definition of done* (Schwaber & Sutherland, 2020, s. 12), som sier noe om kriterier og oppgaver som må være gjennomført før produktet kan defineres som ferdig. Ettersom at de samlet definerer kravene til det ferdige produktet.

2.2 KVALITET

Kvalitet har hatt høy prioritet i dette prosjektet, i dette delkapittelet vil vi presentere hvordan kvalitet defineres, sikres og forbedres i prosjektet. Videre vil de ulike kvalitetssikrende tiltakene vi gjennomførte for sikring av kvalitet bli beskrevet.

Kvalitet er et abstrakt begrep som kan defineres på forskjellige måter avhengig av perspektiv. I Norsk Standard (NS-EN ISO 9000) defineres kvalitet ved "... i hvilken grad en samling av iboende egenskaper oppfyller behov eller forventning som er angitt, vanligvis underforstått eller obligatorisk." (Gundersen & Halbo, 2023) Nasjonal digital læringsarena (NDLA) ser på kvalitet fra kundens perspektiv og hvordan tjenesten oppfyller kundens krav kan defineres (Torgersen, 2018). Gruppen ønsker å levere kvalitet både i form av tjenesten som skal leveres, samtidig som kvaliteten i prosjektstyring også står sentralt. I de følgende seksjonene vil det pekes på hvilke tiltak og rammeverk som er tatt i bruk for at kvaliteten som oppnås er høyest mulig.

For å konkretisere kvalitet kan det brytes ned til intern, og ekstern kvalitet. Den interne kvaliteten kan bedømmes ut ifra hvordan prosjektet gjennomføres, hvilke rammeverk og metoder som tas i bruk, hvor gjenbrukbar og hvor testbar koden er (Stavrinoudis & Xenos, 2008). Det kan også være kvalitetskrav fra en oppdragsgiver som i dette prosjektet vil være Tietoevry. Ekstern kvalitet handler om hvordan brukeren eller kunden definerer kvaliteten. Dette kan måles på brukbarhet, funksjonalitet, effektivitet, fleksibilitet og enkelhet (Stavrinoudis & Xenos, 2008).

- **Dokumentasjon-** Tidlig i prosjektet besluttet gruppen å skrive en prosjektdagbok, som ligger vedlagt i vedlegg 3. Bakgrunnen bak dette valget var at dette var noe gruppen hadde gjort i praksis og hadde god erfaring med å skrive en dagbok. I dagboken skrev alle gruppemedlemmene inn sin *daily standup* for å dokumentere dette skriftlig. Scrum-master brukte også dagboken til å skrive korte refleksjoner eller notater om hva som hadde skjedd denne dagen, inkludert fremdriften i prosjektet, avholdte møter og eventuelle påminnelser. Dagboken har vært til stor hjelp for å holde kontroll på prosjektet, og har fungert som et kvalitetssikrende tiltak i prosjektstyringen.
- **Scrum-** Å ta i bruk scrum var også noe gruppen benyttet for å sikre kvalitet. En undersøkelse som er publisert i "Information and Software Technology" viste at team som følger scrum på et rigid nivå, og gjennomfører alle aktiviteter på en strukturert måte har større sjanse for å oppnå suksess (Kadenic et al., 2023). Gruppen har som nevnt brukt flere aktiviteter i scrum-rammeverket for intern kvalitet, samtidig som hyppige møter mot produkteier og aktører for ekstern kvalitet.
- **Kommunikasjon-** Fysisk kommunikasjon og samtaler er kritisk for å oppnå suksess i prosjekter (McKenna, 2016, s. 21-22). Gruppen anså kommunikasjon som en viktig del av kvalitetssikring. Kontinuerlig og tydelig kommunikasjon med oppdragsgiver, produkteier, veiledere og internt i gruppen bidro til at alle aktører hadde de samme forventningene til resultatet og var med på å sikre

kvaliteten. Et eksempel på dette er styringsgruppemøtene som ble holdt gjennom prosjektperioden. Det ble avholdt tre styringsgruppemøter, og her ble fremgangen presentert for alle veiledere og åpnet for tilbakemeldinger for videre arbeid.

- **Risikomatrise-** Et kvalitetssikringstiltak gruppen tok i bruk var å utarbeide en risikomatrise for å håndtere usikkerheter i prosjektet. Alle prosjekter innebærer usikkerhet, og det å håndtere denne usikkerheten vil kunne redusere risiko (Raknes, 2019). Risikomatrisen blir diskutert mer i detalj i kapittel «3.3.4 Risikomatrise».
- **Kodestandard-** Kodestandard er en rekke retningslinjer som er designet for å sørge for at koden som er produsert er nøyaktig og lesbar. En standard skal sørge for at både utvikleren av koden selv, samt lesere av koden skal kunne lese og forstå den og for å kunne opprettholde den. Implementering av kodestandard i et prosjekt vil sørge for bedre kvalitet, redusere feil og styrke samarbeidet mellom utviklerne (Codacy, 2023). Vårt prosjekt har vi tatt i bruk kodestandard som skal sikre koden vår og gjøre den så presis som mulig ved blant annet å tenke aktivt på navngiving, kommentering, *cohesion* og *coupling*.

3 PROSJEKTGJENNOMFØRING

Dette kapitlet vil gi en omfattende oversikt over gruppens gjennomføring av prosjektet. Det vil bli presentert en rekke ulike aspekter knyttet til prosjektgjennomføringen, inkludert valg av verktøy og teknologier, sprintene, modellering, designprosessen, brukertesting og utvikling. Kapitlet har som mål om å gi en innsikt i hvordan prosjektet ble gjennomført fra planlegging til levering.

3.1 VALG AV VERKTØY OG TEKNOLOGIER

Underkapitlet forklarer valg av verktøy og teknologier og hvordan disse er tatt i bruk knyttet opp mot prosjektet. Under prosjektet har det blitt tatt i bruk mange nye ukjente og kjente teknologier, i likhet med verktøy. Underkapitlet tar for seg disse og forklarer hvordan de er brukt og hvorfor de er tatt i bruk, samt hva de gjør for prosjektet.

Jira Software

Jira Software er et smidig prosjektstyringsverktøy aktivt benyttet i dette prosjektet (Atlassian, u.å). Beslutningen om å ta i bruk verktøyet ble tatt etter ønske fra Kartverket og Tietoenvry, ettersom de ønsket å beholde all dokumentasjon slik at det kunne benyttes videre. Verktøyet ble brukt til å holde kontroll på oppgaver i sprintene, hvor gruppen kunne fordele ansvar, estimere, kategorisere og prioritere hver oppgave. Verktøyet ble også brukt til å lage et *roadmap*, for tydelig oversikt over målene til sprintene og fremgangen i prosjektet.

Figma

Gruppen besluttet å bruke Figma som prototyping-verktøy. Figma er et kraftig og stadig mer populært sanntids designverktøy blant designere (Staiano, 2022, s. 4, 20). Beslutningen om å bruke dette verktøyet ble tatt på grunnlag av flere faktorer, inkludert at det tillater bruk av det meste av funksjonaliteten uten kostnad, noe som sparer ressurser og gir økt fleksibilitet i designprosessen. En annen faktor var at det er skybasert som ga gruppen fri tilgang til prototypen, uavhengig av sted og enhet (Staiano, 2022, s. 5). Erfaring med Figma hos et av gruppemedlemmene påvirket også beslutningen, da dette kunne spare tid ettersom at å lære et nytt verktøy kan være tidkrevende (Staiano, 2022, s. 10).

Miro

Miro er et digitalt verktøy som lar flere mennesker kunne jobbe, tegne og samarbeide samtidig. En plattform som Miro, fungerer utmerket for idemyldring eller felles planlegging (The New Company, 2020). I Miro kan man sette opp forskjellige figurer, notater og andre integrasjoner som gjør det enklere å visualisere. Med utallige funksjonaliteter og metoder gjør Miro det enklere for gruppen å samarbeide og komme med innspill (Wiik, u.å). Gruppen har brukt dette verktøyet under ulike *workshops*, både internt, men også i samarbeid med aktørene som Norkart og Kartverket. Her har gruppen jobbet med skisseringer av løsninger innenfor design og oppsett av modell-løsninger. Det ble også brukt som en lagringsplass for informasjon av forskjellige løsninger som har blitt produsert av gruppen og teknisk veileder fra Norkart. Grunnen til at gruppen valgte å ta i bruk Miro var at det er et bra verktøy for virtuelt samarbeid og deling (Perminova, u.å).

Visual Studio Code

Gruppen tok i bruk Visual Studio Code (VS Code) som koderedigeringsprogram. Valget ble tatt på grunnlag av at dette var redigeringsprogrammet gruppen hadde mest erfaring med fra før og programmets mange fordeler. VS Code er et lett men kraftig redigeringsprogram (Visual Studio Code, u.å). Det har kraftige verktøy for *debugging* og feilsøking som gir muligheten til å finne og fikse opp koden raskt. Programmet gir mulighet for å enkelt kunne koble opp prosjektet til GitHub. Dette bidro til enklere samarbeid mellom utviklerne og lettere for å spore endringer i koden. Programmet inneholder også et stort utvalg av «*plugins*», som gjør det mulig å utvide med mange ulike funksjoner (Heller, 2022). Samlet sett var VS Code et gunstig valg for utviklingen i prosjektet, på grunn av dets funksjonsrike og tilpasningsdyktige utviklingsmiljø som tilbød en effektiv plattform for kodeutvikling og prosjektarbeid.

PgAdmin

Gruppen tok i bruk pgAdmin som er et administrasjonsverktøy for PostgreSQL-database. De direkte fordelene med bruk av pgAdmin er at grensesnittet er brukervennlig, det tilbyr komplette funksjoner som gjør det enklere for å administrere PostgreSQL-database, inkludert muligheten til å kunne kjøre avanserte spørringer. Det er helt gratis å bruke og kildekoden er tilgjengelig for endring og tilpasning (Das, 2021). Generelt inntrykk av verktøyet var positivt helt fra starten, ettersom det bidro til oversiktlig og enkel lagring, overføring og visualisering av dataen som ble arbeidet med.

GitHub

Versjonskontrollsystemer er nyttige verktøy for administrasjon av kildekode, da de gjør det mulig å lagre hver versjon av programmet som man jobber med (Zolkifli et al., 2018, s. 1). I dette prosjektet ble versjonskontrollsystemet GitHub tatt i bruk. GitHub er en nettbasert plattform for versjonskontroll og samarbeid for programvareutviklere, som brukes til å lagre kildekode for et prosjekt og ha muligheten til å spore historikk i forskjellige endringer (Lutkevich, u.å). GitHub hjalp gruppen med å unngå å overskrive hverandres arbeid, samtidig som det ga kontroll på forskjeller mellom ulike versjoner og bidro til at arbeidet ble integrert på en smidig måte (Tsitoara, 2020). Valget var basert på at gruppen hadde tidligere erfaring med systemet, og på grunn av dets kostnadsfrie tilgang og evne til å støtte effektivt samarbeid rundt kildekode, sporing av endringer og håndtering av versjonskontroll (Spinellis, 2012).

ChatGPT

ChatGPT av OpenAI er en nylig lansert chatbot for samhandling med kunstig intelligens gjennom dialog (OpenAI, 2022). GPT-4, OpenAI's siste utvikling viser ytelse på linje med menneskelige standarder på flere faglige og akademiske områder (OpenAI, 2023a). På den tekniske fronten kan chatboten bidra til blant annet kodegenerering, forklaring av kode, debugging og løsningsforslag (OpenAI, 2022).

Gruppen ble tidlig oppmuntret av produkteier og veiledere til å ta i bruk Chatboten som hjelpemiddel for programmeringen. Bruk av chatboten kan ved riktig bruk forbedre produktivitet på en rekke måter, og har økt effektiviteten på kodeproduksjonen betydelig i dette prosjektet (Teubner et al., 2023, s. 96). Et eksempel på dette er da gruppen arbeidet med å utvikle et Python-script for uthenting av data fra en datakilde via API og lagre dataene i en annen datakilde. Basert på den begrensede kunnskapen på dette tidspunktet estimerte gruppen at oppgaven ville kreve en uke med arbeid, men med chatbotens assistanse var scriptet utviklet på en arbeidsdag.

Dette delkapittelet tar for seg de ulike teknologiene som ble valgt for utviklingen av løsningen og begrunnelsen bak disse valgene. Vi vil beskrive hver teknologi, og hva som gjorde dem velegnet for vårt prosjekt.

HTML og CSS

HyperText Markup Language (HTML) ble benyttet til strukturering av webgrensesnittet, mens Cascading Style Sheets (CSS) ble brukt til å organisere det grafiske innholdet. HTML fungerer som en grunnleggende byggestein for webgrensesnittet og definerer betydningen og strukturen til innholdet (MDN Web Docs, u.å). CSS er et språk som brukes til å definere hvordan innhold skal gjengis på skjermen (MDN Web Docs, u.å). Valget om å bruke disse språkene var en bevisst beslutning, ettersom at vi allerede hadde kjennskap til dem. Med tanke på den

krevene backend-delen som vi visste ville kreve mye oppmerksomhet og ressurser, kunne vi spare tid og ressurser ved å bruke noen kjente språk til frontend-delen.

JavaScript

JavaScript er omtalt som det programmeringsspråket som er mest brukt per dags dato og er et av språkene nettlesere støtter og forstår seg på best (Svekis et al., 2021, s. 2). I dette prosjektet er språket brukt for å implementere funksjonaliteten som skjer i webgrensesnittet. Valget om å bruke dette språket ble basert på at det finnes god støtte og dokumentasjon rundt språket, samt at det er standard for web-utvikling (MDN Web Docs, u.å).

Bootstrap

I tillegg til gruppens egendefinerte CSS-stilark, tok gruppen i bruk CSS rammeverket Bootstrap. Bootstrap er et rammeverk som tilbyr variabler, komponenter og verktøy for effektiv frontend-utvikling (Bootstrap, u.å). Beslutningen om å ta i bruk Bootstrap fremfor andre frontend-rammeverk var mye basert på at gruppen hadde tidligere arbeidet med rammeverket, valget blir nærmere beskrevet i kapittel «5.2.1 Valg av rammeverk».

Leaflet

For webgrensesnittets kart og kartfunksjoner ble JavaScript-biblioteket Leaflet tatt i bruk. Leaflet er et bibliotek med åpen kildekode for interaktive kart som inneholder en rekke kartfunksjoner (Leaflet, u.å). Dette valget ble tatt etter oppfordring fra veileder hos Kartverket og Norkart.

Python

I backend-delen av prosjektet, som håndterte dataflyt, datakilde-tilkoblinger og funksjoner, tok gruppen i bruk Python som programmeringsspråk. Det tilbyr et stort utvalg av biblioteker som er spesielt rettet inn mot kunstig intelligens som vår oppgave handler om, for eksempel TensorFlow (Innes et al., 2018). Avgjørelsen om å ta dette i bruk ble tatt basert på samtale med produkteier som selv har mest erfaring med Python. Det var også en fordel ettersom at eksisterende kode som skulle benyttes i prosjektet var skrevet i dette språket.

PostgreSQL

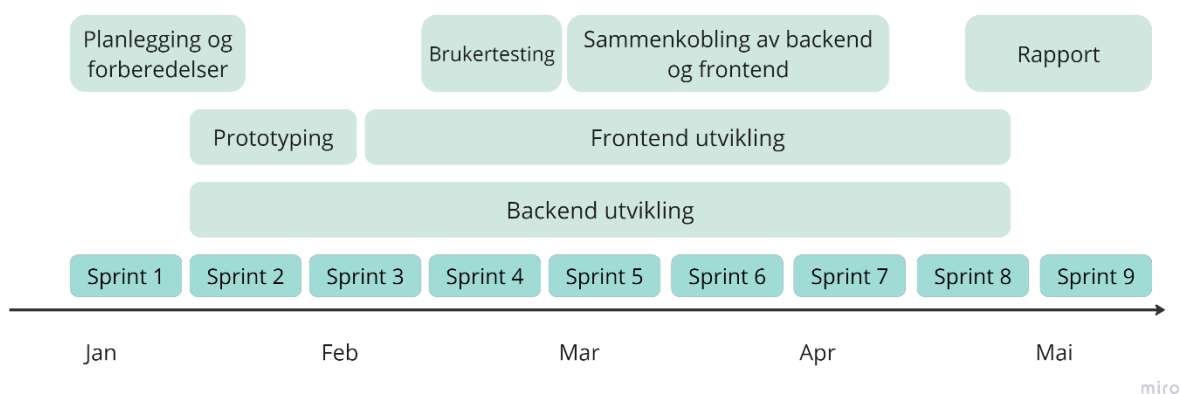
PostgreSQL er en relasjonsdatabase med åpen kildekode (Microsoft, u.å). Til prosjektet gir det betydelige fordeler innenfor geospasiale funksjoner. Gruppen har tatt i bruk dette på grunn av flere anbefalinger fra teknisk veileder hos Norkart. I tillegg, er datatypene til PostgreSQL innenfor geometri som inneholder punkter, linjer, sirkler og polygoner som er aktuelt for prosjektet (Postgresql, u.å).

FastAPI

Et kode-rammeverk har sin hensikt i å kunne assistere under utviklingen av nye tjenester og for å kunne hjelpe til med å garantere en gitt standard i et prosjekt. Et rammeverk for koding hjelper utviklere med funksjonalitet, slik at fokuset kan være på å gjøre et prosjekt unikt (Ranjan, 2021). For prosjektet valgte gruppen å ta i bruk rammeverket, FastAPI for backend som tilkobling til webgrensesnittet. Beslutningen om å bruke FastAPI ovenfor Flask ble gjort etter samtaler med produkteier og etter grundige undersøkelser rundt rammeverket.

3.2 SAMMENDRAG AV SPRINTENE

Dette delkapittelet vil gi en oversiktlig oppsummering av de ulike sprintene gjennom prosjektet. Formålet er å gi en oversikt over tidsperspektivet til prosjektgjennomføringen. Vi vil kort forklare hovedfokuset og fremgangen for hver sprint. En oversikt over tidslinjen for sprintene er presentert i figur 1.



Figur 1. Illustrasjon av tidslinje til sprinter og hovedfokus.

Sprint 1 hadde fokus på kartlegging, planlegging og tilegning av kunnskap og forståelse om prosjektet og nødvendige teknologier. I begynnelsen var det mye å sette seg inn i, men gruppen fikk informasjon fra involverte aktører både gjennom møter og dokumenter. For den første sprinten ble målet satt til å lese dokumenter og etablere en viss forståelse av prosjektet og relevante teknologier og utrykk. I Kapittel «3.2 Kartlegging» vil vi utforske og presentere mer om denne sprinten og dens utførelse. I tillegg ble flere av valgene av teknologier som presentert i «3.1 Valg av verktøy og teknologier» tatt under denne sprinten.

I sprint 2 ble designprosessen innledet med konseptutvikling og utarbeiding av skisser, etterfulgt av prototyping, designprosessen vil bli utforsket mer detaljert i kapittel «3.5 Designprosessen». Samtidig ble arbeidet med backend-løsningen påbegynt, med fokus på innhenting og lagring av data, som vil bli videre omtalt i delkapittel «3.7.2 Innhenting og lagring av data». I denne sprinten startet også arbeidet med modellering, som vil bli nærmere omtalt i delkapittel «3.4 Modellering». Videre ble det satt av en betydelig mengde tid til planlegging og gjennomføring av en presentasjon under en faglunsj for med GeoForum og Digin. I tillegg begynte vi også med rapportskrivning. I denne sprinten ble også det første styringsgruppemøte holdt.

I sprint 3 fortsatte flere aktiviteter som ble påbegynt i sprint 2, inkludert modellering, prototyping og innhenting og lagring av data. I løpet av sprinten ble prototypen ferdigstilt, og dermed kunne arbeidet med frontend utviklingen starte, med det første fokuset på å implementere et kart og tegnefunksjoner. Denne prosessen blir

nærmere omtalt i delkapittel «3.7.1 Utviklingen av webgrensesnittet». Videre ble også WMS-servere med ortofoto til trening av modeller undersøkt, dette beskrives nærmere i delkapittel «3.7.3 Generering av treningsdata». I denne sprinten holdt vi også en presentasjon, denne gangen for Forsking og utviklingsavdeling hos Kartverket.

I sprint 4 ble brukertesting av webgrensesnittet planlagt og gjennomført, videre beskrevet i delkapittel «3.6 Evaluering/brukertesting». Etter en del fremgang i frontend utviklingen, ble det påbegynt undersøkelser rundt hvordan backend og frontend skulle kobles sammen. Her ble flere rammeverk testet og vurdert som beskrevet i delkapittel «5.2.1 Valg av rammeverk». Gjennom et møte med teamet bak KartAi og en del testing fikk gruppen også til å generere treningsdata med bruk av KartAi-prosjektet.

Sprint 5 hadde som mål å koble sammen frontend og backend, noe som var påbegynt i Sprint 4. Av den grunn ble det allokert ekstra ressurser til frontend-utvikling, slik at dette ble ferdigstilt. Rammeverk ble også undersøkt i denne sprinten, hvor det ble avgjort å bruke FastAPI som rammeverk til backend-delen, begrunnet i delkapittel «5.2.1 Valg av rammeverk». Gruppen jobbet også mot å få automatisert hele backend-prosessen, slik at alle scriptene skulle kjøre i riktig rekkefølge, beskrevet i delkapittel «3.7.3 Automatisering av backend». Det ble også gjennomført et nytt styringsgruppemøte hvor det ble gått gjennom tidligere aksjonspunkter og hvordan disse ble tatt til følge.

I Sprint 6 fortsatte implementeringen av FastAPI rammeverket for sammenkobling av frontend og backend. Det ble også besluttet at Docker skulle brukes til levering av prosjektet. Videre ble resultatene fra brukertesting analysert, og det ble gjort ulike justeringer og forbedringer i frontend koden.

Sprint 7 ble noe forkortet grunnet påskeferie, men det forekom likevel ulike gjennombrudd. Vi fikk til vellykket kjøring av hele prosjektet gjennom Docker container og fikk koblet sammen front-og backend. I tillegg fikk vi programmet til å laste ned data uten å trene KI-modellene, som beskrives i kapittel «3.7.3 Generering og nedlastning av treningsdata». Videre ble det besluttet å ta i bruk Norkart sin WMS server, et valg som blir videre begrunnet i delkapittel «3.7.3 Generering og nedlastning av treningsdata». Parallelt ble ulike input-felt i webløsningen implementert. Det ble også gjennomført et nytt styringsgruppemøte hvor foreløpig resultat ble presentert for aktørene.

Sprint 8 hadde fokus på ferdigstilling av prosjektet, hvor det blant annet ble jobbet med å gjøre designet til webløsningen mer responsivt. I tillegg ble all kode gjennomgått, med fokus på å sørge for at retningslinjene satt i starten ble fulgt. I slutten av sprinten ble det satt kodestopp, og fokuset ble satt til rapportskriving. I sprint 9 var også fokuset å skrive bachelorrapporten og levere denne.

3.3 KARTLEGGING

I dette kapittelet beskrives oppstarten av prosjektet og planleggingen som ble gjort. Det inkluderer etablering av struktur, opprettelse av dokumenter og planlegging av prosjektgjennomføringen. Det presenteres også en *roadmap* som viser de overordnede oppgavene i prosjektet og en kapasitetsplan som gir en oversikt over tilgjengelige utviklingstimer i hver sprint. Det inkluderer også en risikomatrix som tar for seg ulike risikomoment gruppen støtte på i løpet av prosjektet.

3.3.1 Prosjektstart og scope

Oppstarten i prosjektet handlet om planlegging og strukturering av prosjektet. Prosjektet startet med et møte hvor gruppen og aktørene knyttet til prosjektet deltok. Her ble vi introdusert til alle veiledere, hva kunden ønsket fra oss, fikk utdelt nødvendig informasjon og veiledning om god praksis for prosjektstyring. Videre i planleggingsfasen ble det gjennomført ulike aktiviteter som opprettelse av kapasitetsplan og risikomatrix. I tillegg ble det opprettet formelle dokumenter som gruppekontrakt, retningslinjer for rapportskrivning, retningslinjer for kode, *roadmap* over prosjektet, og en prosjektbeskrivelse.

Når prosjektet var presentert, ble det nødvendig å definere omfanget (scopet) av prosjektet. Oppgaven bestod av to deler, hvor scopet til del 1 opprinnelig ble satt til å inneholde uthenting av SFKB data via NGIS-OpenAPI. Gruppen besluttet at scopet kunne utvides, avhengig av tidsbruken på opprinnelig scope. Det ville da eventuelt utvides til å inkludere bruk av bygningsdata i KartAi-prosjektet og oppkobling mot WMS-server for generering av treningsdata. På denne måten ville gruppen unngå å ta for seg et for stort scope, og heller beholde muligheten til å utvide det på et senere tidspunkt. Scopet til del 1 ble tidlig utvidet, da opprinnelig scope ble utført raskere enn estimert. Del 2 var knyttet til frontend med tilhørende design, og utvikling av en selvbetjeningsportal hvor en bruker kunne definere et område, og få treningsdata fra dette området.

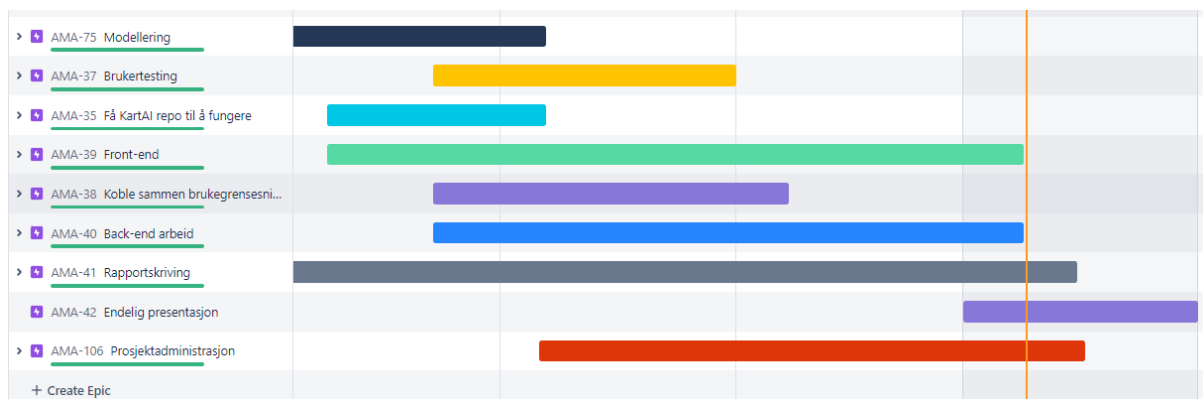
3.3.2 Roadmap

I begynnelsen var det utfordrende å planlegge prosjektet og hva som skulle bli gjort, på grunn av begrenset forståelse for oppgaven. For å løse dette anbefalte veileder hos Tietoevry at gruppen laget et *roadmap* for å få en bedre oversikt over hva som skulle gjøres. Gruppen laget et *roadmap* med *epics*, som er store, abstrakte overordnede oppgaver som ofte pågår over flere sprinter og blir brutt ned til mer konkrete mindre oppgaver i *sprint backlog* (Salimi, u.å)

På grunn av den begrensede kunnskapen i begynnelsen, besluttet gruppen at de første sprintene skulle ha fokus på tilegnelse av forståelse og kunnskap blant gruppemedlemmene. Av denne grunn hadde de første *epics* fokus på dette. Det ble

også utarbeidet *epics* relatert til NGIS-OpenAPI, ettersom at det var tydelig at dette stod sentralt i backend-delen av prosjektet. Tilknyttet frontend og design delen ble det satt opp oppgaver relatert til konseptutvikling og visualisering.

De ulike *epics* ble satt i tidsperspektiv i forhold til hvor gruppen ønsket å være i prosjektet i forhold til fremdriften. *Epics* varierte i forhold til hvor konkret de var. Et eksempel på en relativt konkret *epic* var «Brukertestning» med en tidsramme på tre sprinter og knyttet til få oppgaver. En mindre konkret *epic* var «Backend arbeid», som inneholdt et stort antall oppgaver og bredte seg over nesten hele prosjektperioden. Det endelige *roadmap* vises i Figur 2 og fullstendig i vedlegg 4.



Figur 2. Skjerm bilde av endelig roadmap.

3.3.3 Kapasitetsplan

For å få en oversikt over hvor mange utviklingstimer som var tilgjengelig til enhver tid, og etter anbefaling fra veileder hos Tietoenvry, utarbeidet gruppen en kapasitetsplan. Planen ble basert på en mal fra Tietoenvry og inneholdt en oversikt over alle dager og antall timer gruppen skulle jobbe i hver sprint. Planen bidro til at gruppen og produkteier hadde en felles forventning til gruppens kapasitet. I planen ble vanlige arbeidstider og timer til *sprint planning*, kurs og eventuelle planlagte fravær/høytider separert. Som illustrert i Figur 3 hadde gruppen i utgangspunktet en kapasitet på totalt 250 timer per sprint.

Teamets tilgjengelighet i mars/april (Timer)																
Allokerings	Navn	Rolle	Man	Tir	Ons	Tors	Fre	Lør	Søn	Man	Tir	Ons	Tors	Fre	Lør	Totale antall timer per person
			20	21	22	23	24	25	26	27	28	29	30	31	1	
100%	Kasper	Utvikler	6,5	6,5	0	6,5	5,5			6,5	6,5	0	6,5	5,5		50
100%	Tobias	Utvikler	6,5	6,5	0	6,5	5,5			6,5	6,5	0	6,5	5,5		50
100%	Niklas	Utvikler	6,5	6,5	0	6,5	5,5			6,5	6,5	0	6,5	5,5		50
100%	Ine	UX & Utvikler	6,5	6,5	0	6,5	5,5			6,5	6,5	0	6,5	4,5		49
100%	Sondre	Prosjektleder	6,5	6,5	0	6,5	5,5			6,5	6,5	0	6,5	5,5		50
50%	Totale antall timer per dag		32,5	32,5	0	32,5	27,5			32,5	32,5	0	32,5	26,5		

Totale antall utviklingstimer tilgjengelig for denne sprinten (eks. planning)		216,5
Prosjektledelse (eks. planning)		32,5
Totalt (ink. sprintplanning)		249

Forklaringer	
	Helg / Helligdager
	Ferie / Avspaseringer
	Sprintplanlegging
	Sykdom
	Kurs / kompetanseheving

Figur 3. Skjerm bilde av Sprint 6 i Kapasitetsplanen.

3.3.4 Risikomatrixe

En annen sentral aktivitet under kartleggingsfasen var utarbeidingen av en risikomatrixe. Dette ble gjort etter anbefaling fra veileder hos Tietoevry som også gav gruppen en mal. En risikomatrixe har som mål å øke sjansen for å nå prosjektets mål, samtidig minimere risiko for at prosjektet feiler. Risikostyring er en kontinuerlig prosess og skal følges opp konstant for å implementere eventuelle endringer (Eriksen, 2022).

Risikomatrixen tok for seg flere momenter gruppen så for seg kunne oppstå og hindre fremgang i prosjektet. Hver risiko ble bedømt etter sannsynlighet, og konsekvens, og gitt en score mellom 1-4 til både sannsynlighet og konsekvens. Disse ble multiplisert og gav en risikoscore på 1-16. Fargekoder ble brukt til å kategorisere risikoscoren, hvor rød var høy risiko, gul var middels risiko, og grønn er lav risiko. Hver risiko ble tildelt til et gruppemedlem som var ansvarlig for oppdatering av risikomomentet, og sørget for at tiltak ble iverksatt om problemer oppstod. Figur 4 illustrerer et moment fra risikomatrixen med tilhørende vurderinger og tiltak, den endelige risikomatrixen kan sees i vedlegg 2.

Risikomoment	Konsekvensgrad(1-4)	Sannsynlighet sgrad(1-4)	Samlet risikovurdering(1-16)	Forebyggende tiltak	Skadereduserende tiltak	Ansvarlig
<p>Prosjektet har usikkerheter knyttet til tilgangsbehov Nåværende forståelse av prosjektet er ikke nødvendigvis stor nok til å vite hvilke tilganger som er nødvendige for gjennomførelse. Dersom behov for tilgang på f.eks systemer eller databaser oppstår sent i prosessen vil det kunne føre til at prosjektet ikke blir ferdig innen fristen eller at kvaliteten på resultatet blir dårligere enn forventet.</p>	4	2	8	God planlegging og videreformidle raskt ved usikkerhet om behov for tilganger.	Spørre om mulighet for tilgang så raskt som mulig.	Kasper

Figur 4. Skjermbilde av et moment i Risikomatriksen med tilhørende vurderinger og tiltak.

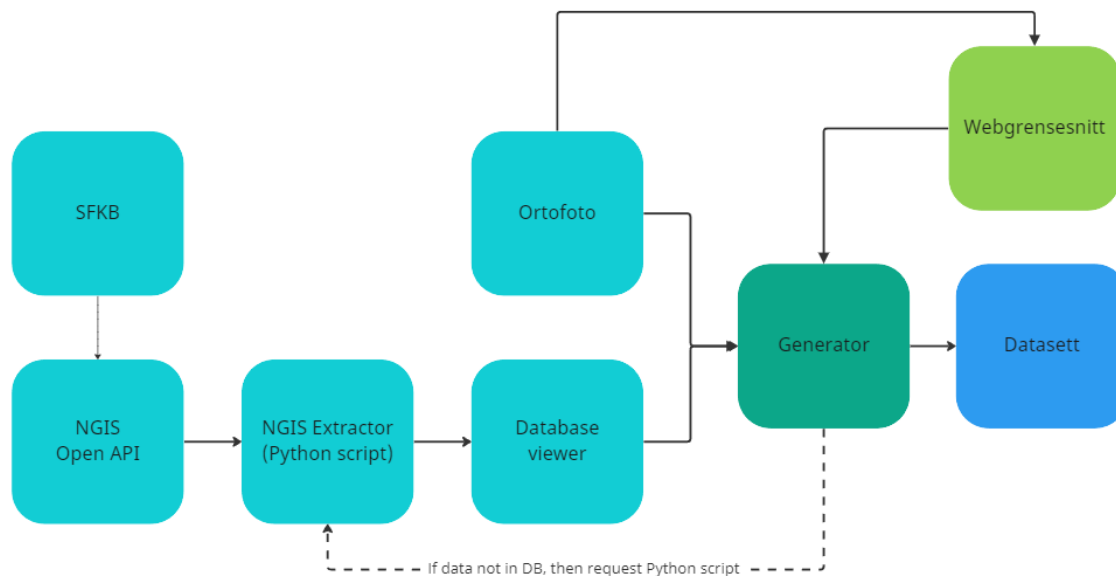
3.4 MODELLERING

Dette kapitlet beskriver prosessen med å lage modeller av systemets komplekse dataflyt. Det gjøres rede for bakgrunnen til opprettelsen av de ulike modellene og hvordan ulike aktørers behov og krav ble tatt i betraktning. Videre beskrives det hvordan gruppen arbeidet seg frem til en endelig modell som visualiserer dataflyten i systemet.

Tidlig i prosjektet fikk gruppen en oppgave om å tegne en modell av dataflyten i løsningen. I første omgang ble dette sett på som en relativt kortvarig oppgave som kunne løses i løpet av et par dager med litt veiledning. Etter å ha tegnet opp flere varianter av hvordan gruppen tenkte dataflyten i løsningen skulle være, fikk vi tilbakemelding om at vi ikke hadde forstått alt helt riktig. Produkteier var i utgangspunktet fornøyd, men det var flere aktører i prosjektet som også måtte godkjenne modellen. Her ble det oppdaget at de forskjellige veilederne hadde litt ulike ønsker om hva innholdet skulle være. Produkteier i Kartverket var mest opptatt av at komponentene var riktig, og dataflyten mellom systemer skulle stemme. I motsetning til veileder i Kartverket som ønsket en modell som viste dataflyten, men den skulle også være enkel å forklare videre til personer som ikke hadde den tekniske forståelsen. Veileder i Tietoenvry ønsket derimot en modell som var riktig i forhold til UML-modellering og at komponentene i modellen skal ha riktig ikon, og gå i riktig retning.

For å tilfredstille de forskjellige kravene ble det utarbeidet over 20 forskjellige modeller siden oppstarten av prosjektet. De første modellene ble utformet etter hvordan gruppen kunne se for seg at oppgaven skulle løses. Disse var ikke basert på

en fullstendig forståelse og kunnskap av prosjektet med tanke på dataflyt og datakilder. Et eksempel på en av de første modellene er illustrert i Figur 5.

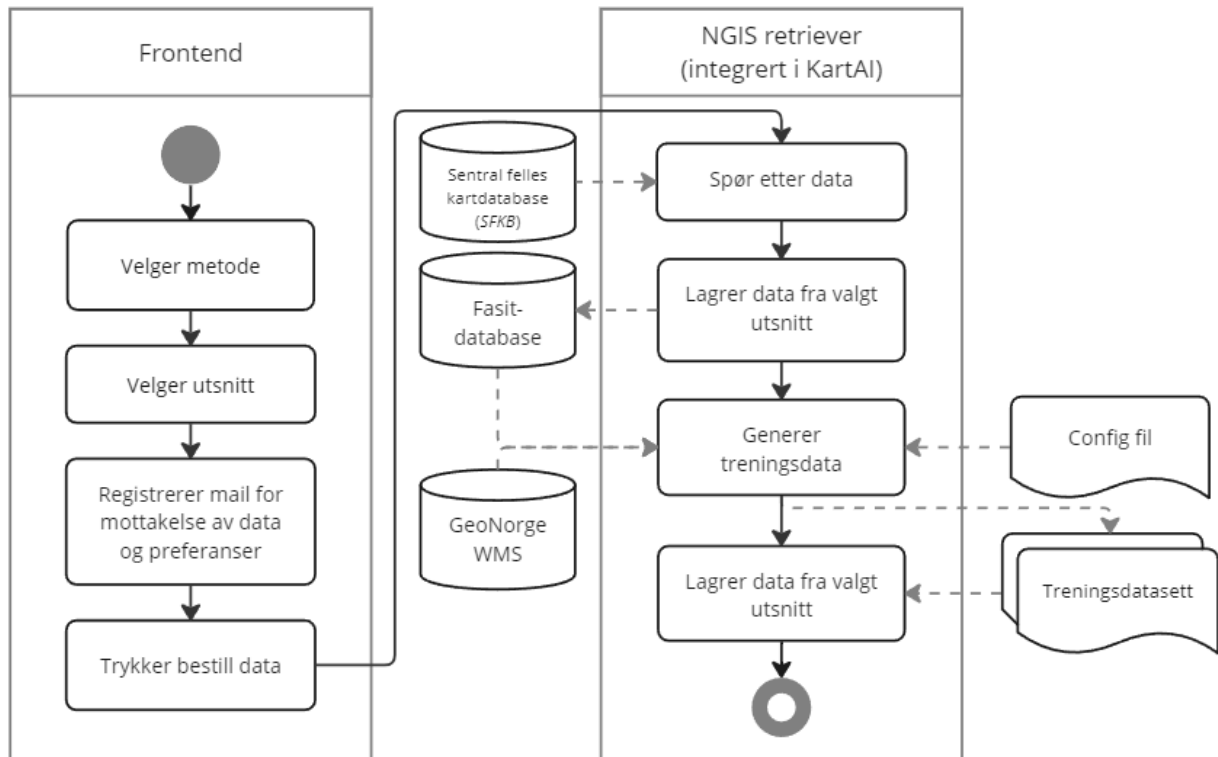


Figur 5. En av de første modellene laget av dataflyten til løsningen.

Senere i prosjektet når gruppen fikk mer innsikt i hvordan dataflyten i løsningen skulle være, ble det laget nye modeller, men det oppstod nye utfordringer. Under en *workshop* med utvikleren av KI-algortimene i KartAi-prosjektet ble en modell presentert. Hensikten med workshopen var å forstå hvordan koden fungerte, og hvordan løsningen kunne gjennomføres. Dette resulterte i flere konflikter i forhold til modellene som var laget. Disse konfliktene gikk ut på hvordan dataflyten mellom datakildene og hvordan KartAi kunne ta imot data. I utgangspunktet trodde vi at data kunne sendes fra Python-scriptet til KI-algortimene, men det viste seg at KartAi henter data direkte fra datakildene.

Ettersom aktørene alle hadde sine krav og forventninger ble modellering en kontinuerlig iterativ prosess, til vi kom fram til et endelig utkast. Den endelige modellen (Figur 6) er et hybrid-diagram mellom aktivitetsdiagram og sekvensdiagram og er delt i to forskjellige deler, frontend og backend. Det er flere deler i denne modellen som må presiseres.

- NGIS retriever er et Python-script laget av gruppen som er integrert i KartAi. Scriptet sin oppgave er å ta i bruk API, koblinger til database, og dataflyt.
- KartAi inneholder KI-modellene, det er her treningsdata genereres og KI-modellene trenes.
- Frontend-delen er webgrensesnittet hvor brukeren definerer hvilket område data ønskes fra i tillegg til enkelte regelsett (se kapittel «4.1 Endelig løsning»).



Figur 6. Endelig modell av dataflyten til løsningen.

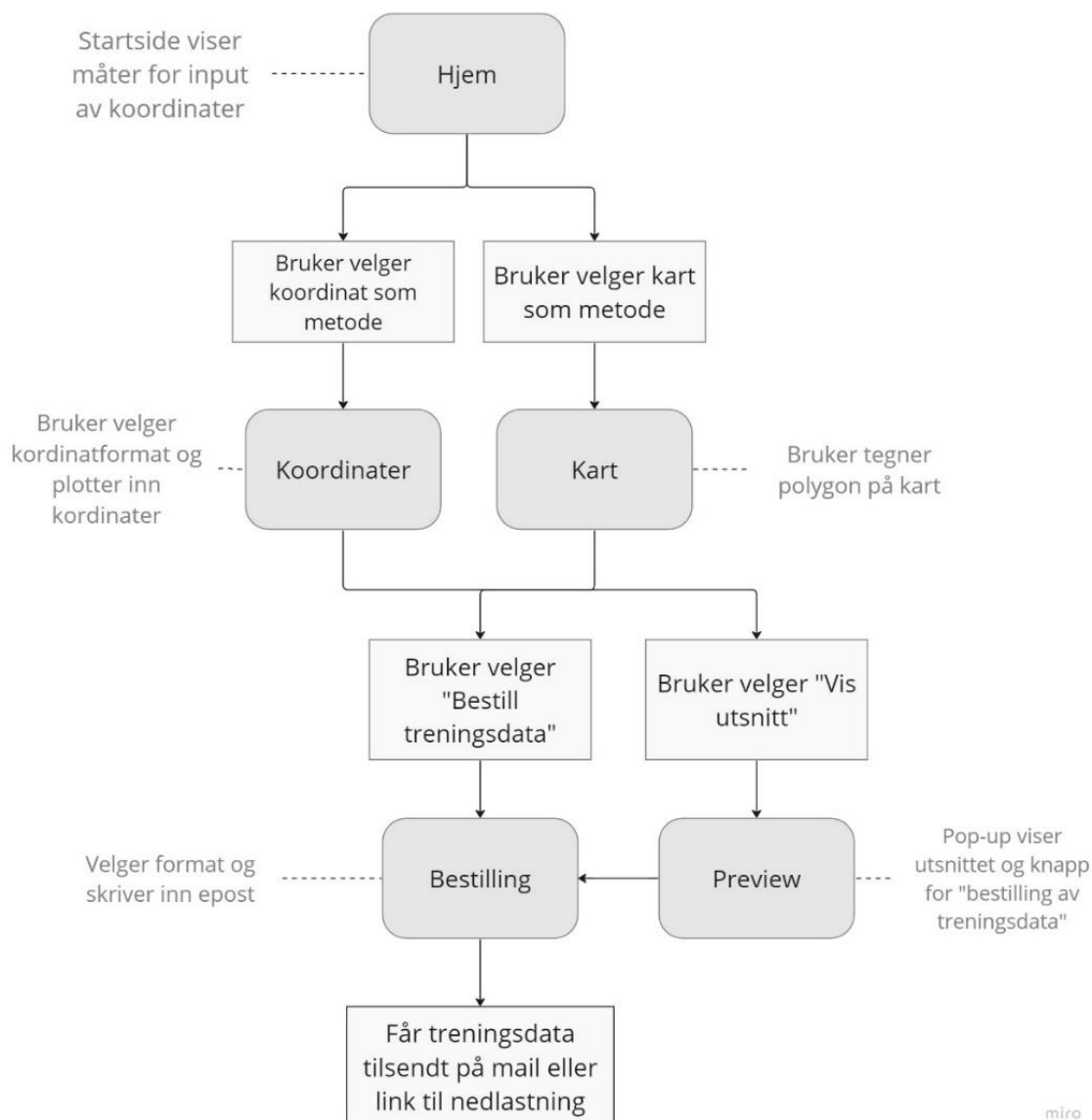
Det definerte området og regelsettet sendes til NGIS retriever som henter ut data basert på område som er valgt. Dataen som hentes ut blir skrevet til en fasit-database. Så initieres KartAi delen av systemet, som henter data fra to forskjellige datakilder, fasit databasen som inneholder fasitdata, og en GeoNorge WMS med ortofoto som modellen skal trene på. KartAi genererer deretter et datasett fra disse datakildene og klargjøre data for maskinlæring. Dette datasettet komprimeres til en ZIP-fil og sendes til brukeren via e-post.

3.5 DESIGNPROSESSEN

Kapittelet tar for seg utvikling og forståelse av hva webløsningen skulle inneholde og hvordan den skal bli seende ut. Det blir gjort en gjennomgang av designprinsipper for løsningen og presentert diverse visualiseringer av prototypen og hva den inneholder.

3.5.1 Konseptutvikling

I begynnelsen av prosjektet var det avgjørende å utarbeide en klar forståelse av hva webløsningen skulle inneholde. For å oppnå dette ble det i samarbeid med produkteier laget et diagram over brukerreisen (Figur 7) som viste trinnene og sidene webløsningen skulle inneholde. Diagrammet gjorde det mulig å kartlegge hvilket innhold som skulle inkluderes og hvordan navigasjonen skulle være strukturert.

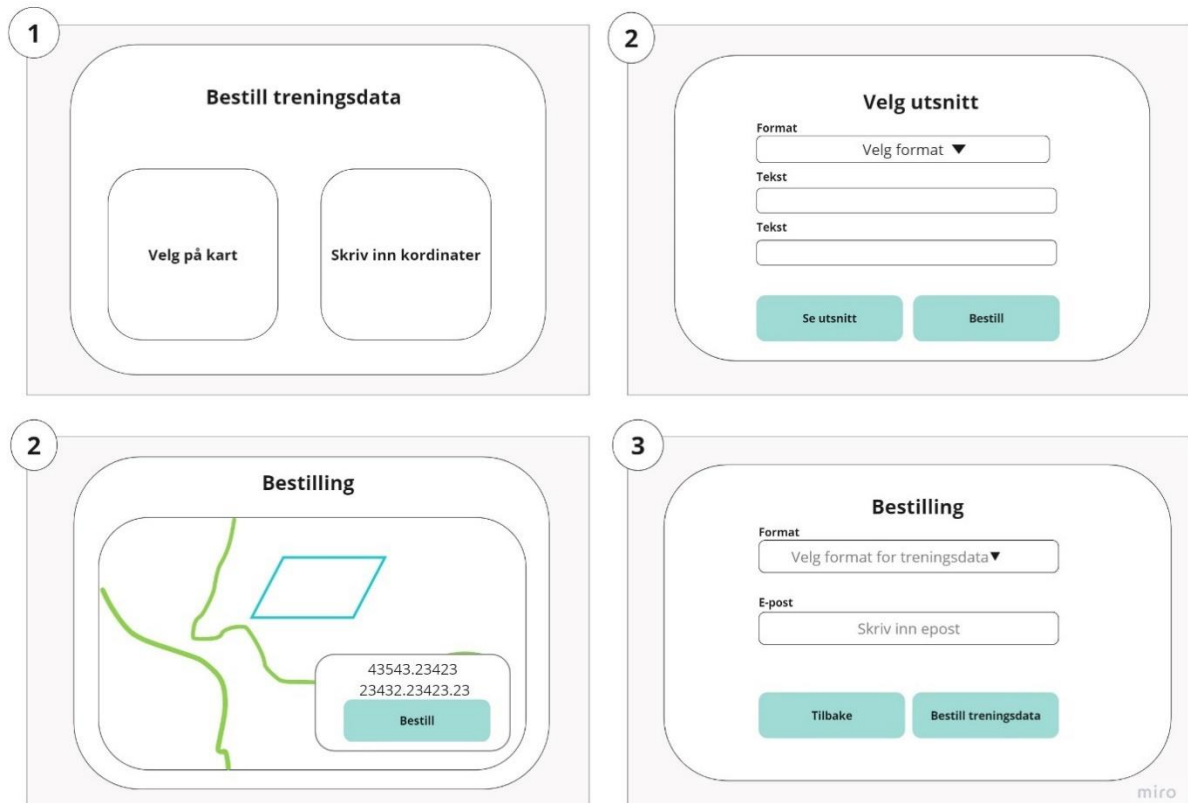


Figur 7. Diagram over brukerreisen til webløsningen.

3.5.2 Skissering og prototyping

Basert på stegene i brukerreisediagrammet ble det utarbeidet skisser av den planlagte webløsningen, disse vises i Figur 8 og vedlegg 6. Skissering har til hensikt å uttrykke ideer og skape ny innsikt og kunnskap rundt ideene (Sandnes, 2018, s. 249-250). Hovedformålet til skissene var å sikre en felles forståelse og enighet mellom gruppen og produkteier om hvordan løsningen skulle fungere og se ut. Dette gjorde det mulig for begge parter å diskutere hvilke elementer designet skulle inneholde, som bidro til å etablere en felles forståelse og enighet mellom gruppen og arbeidsgiver om

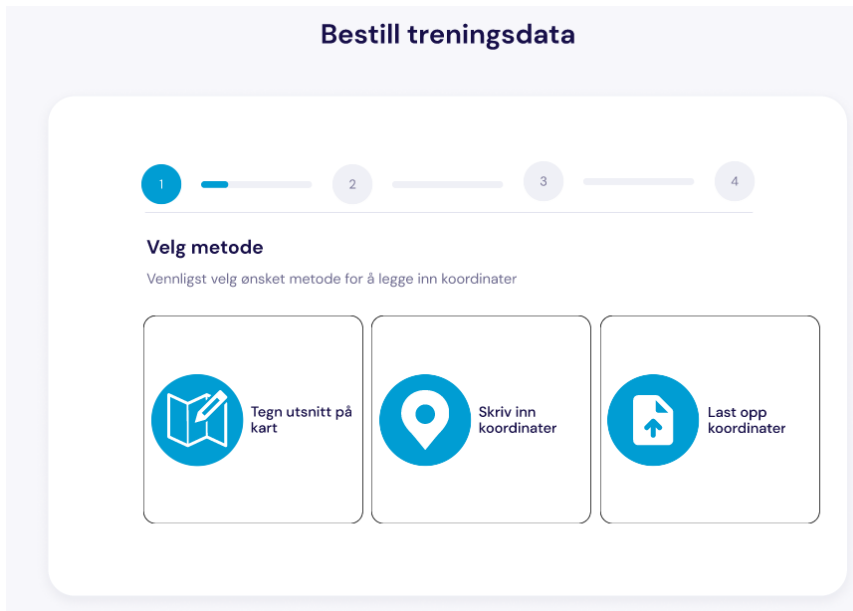
løsningsens design. Skissene la grunnlaget til det neste steget i designprosessen som var utvikling av prototypen.



Figur 8. Skisser av utarbeidet under planleggingen av designet til løsningen.

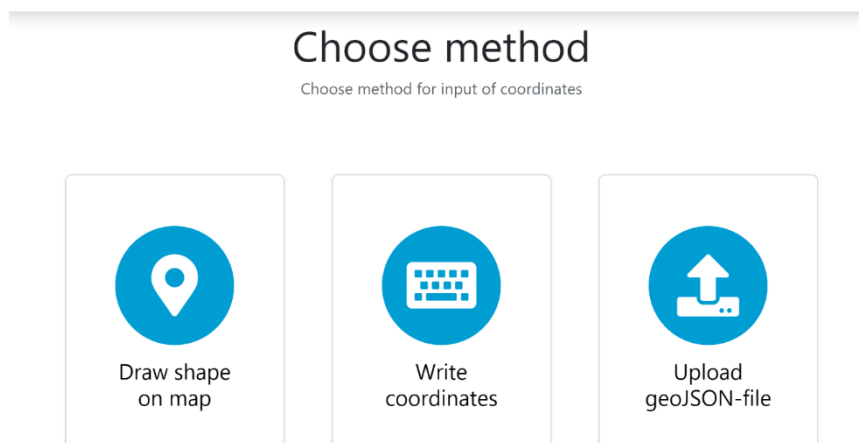
For å sikre at produktets funksjonalitet og design oppfylte produkt- og brukerbehovene, ble det besluttet at en prototype skulle lages. Ved å lage en prototype og brukerteste den kunne vi sikre at produktet var brukervennlig og oppfylte kravene før utviklingen av webløsningen startet, og dermed spare tid og ressurser (Sandnes, 2018, s. 267).

Prototypen ble laget i Figma med fokus på å lage et enkelt og brukervennlig webgrensesnitt, med utgangspunkt i skissene (Vedlegg 5). Prototypen ga en klar oversikt over webgrensesnittets planlagte utseende og funksjoner og ble dermed et viktig grunnlag for utviklingen. Webgrensesnittets utseende ble relativt likt som prototyping viser dermed prototypens innflytelse på designprosessen, et eksempel på dette er landingssiden til prototypen (Figur 9) og landingssiden til den endelige løsningen (Figur 10).



Figur 9. Skjerm bilde av landingssiden i prototypen.

Kart-AI Trainingdata

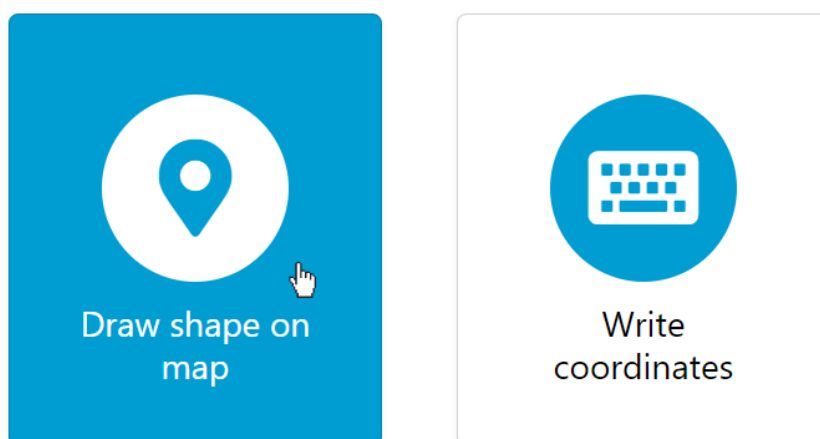


Figur 10. Skjerm bilde av landingssiden i den endelige løsningen.

3.5.3 Designprinsipper

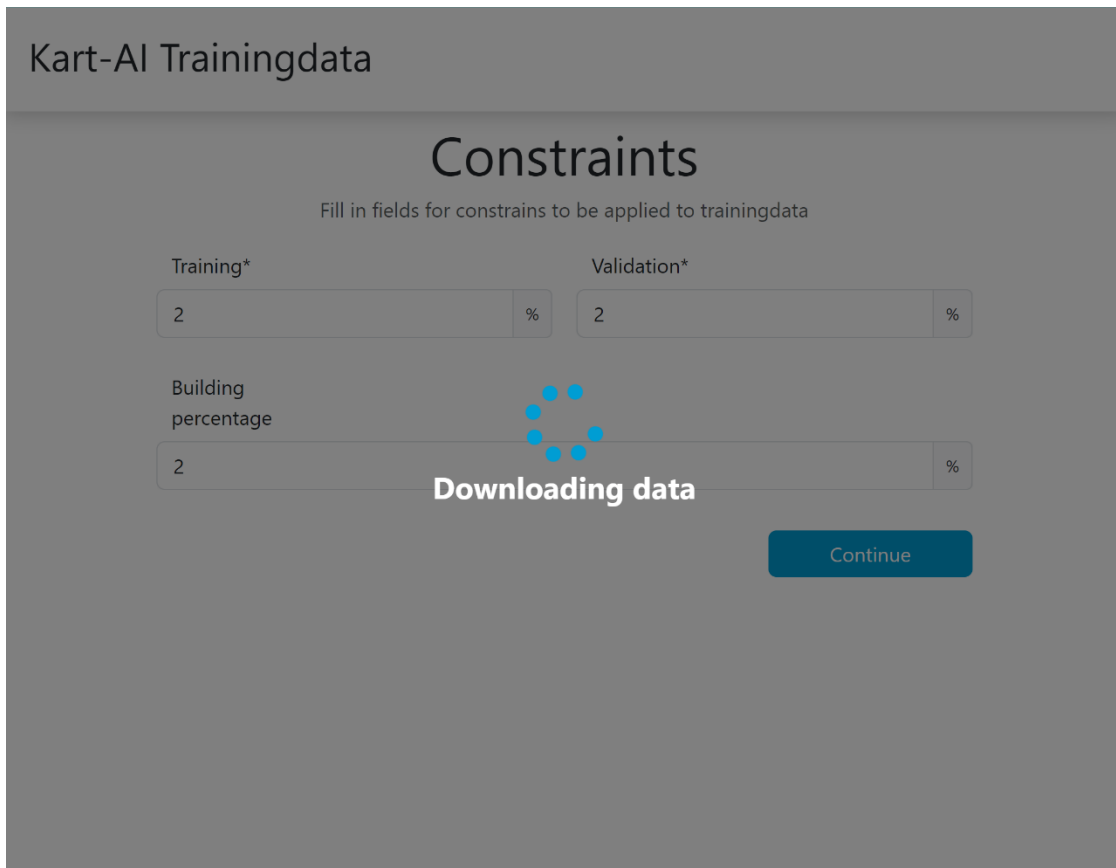
Designprinsipper kan være til nytte for å veilede designeren gjennom designprosessen og var aktivt benyttet til dette i prosjektet (Benyon, 2014, s. 86). Alle avgjørelser som er tatt angående design og brukervennlighet er bygget på Benyons tolv designprinsipper. Videre vil vi ta for oss noen eksempler på anvendelse av noen av designprinsippene i designprosessen.

- *Visibility* er et designprinsipp som fokuserer på at brukere enkelt skal kunne se og forstå tilgjengelige funksjoner og systemfunksjonalitet, har vært et sentralt aspekt i utformingen av nettstedet (Benyon, 2014, s. 87). Et eksempel på dette er at vi har brukt visuelle signaler som farger og kontraster til å skille ut viktige elementer og grensesnittet er utformet på en enkel måte for å unngå forstyrrende elementer.
- *Affordance prinsippet* handler om å designe elementer slik at det er klart hva de er til (Benyon, 2014, s. 87) For å oppnå dette har vi tydeliggjort interaktive elementer som knapper og klikkbare kort på startsiden, med *hover*-tilstander som gir et tydelig visuelt signal om mulighet for interaksjon. Et eksempel er fargeendring på kort på startsiden når brukeren holder musepekeren over kortene, for å tydeliggjøre at disse er klikkbare, illustrert i Figur 11. Et annet eksempel på *affordance* er at knappen for å gå videre er grå til brukeren legger inn ønskede koordinater, slik at brukeren forstår at det ikke er mulig å gå videre før koordinatene er lagt inn. Dette går også innenfor *constraints* prinsippet som omhandler det å gi begrensninger til brukeren for å forhindre at brukeren gjør feil og ved å begrense handlinger som ikke er tillatt. *Constraints* vil hjelpe brukeren til å gjøre riktig valg i systemet og fungere som en veileder. Dette er noe som er implementert ved å sørge for at brukeren ikke kan gjøre handlinger eller gå videre ved mindre kravene er oppfylt til løsningen (Benyon, 2014, s. 87).



Figur 11. Skjerm bilde av kort på landingssiden når man holder musepekeren over kortet for tegning på kart (venstre).

- Et annet prinsipp avendt under designet av løsningen er *Feedback*, Som handler om å fortelle brukeren om hva som skjer (Benyon, 2014, s. 87). Et eksempel på denne er bruken av et animert *loading*-ikon som forteller brukeren at systemet er opptatt med noe, vist i Figur 12. På denne måten forsterkes brukeren følelse av kontroll og man unngår at brukeren forlater webgrensesnittet, eller blir forvirret over hva som skjer (Benyon, 2014, s. 87).



Figur 12. Skjerm bilde av webløsningen når systemet jobber med nedlastning av data.

- Valget av det animerte *loading*-ikonet ble basert på *Familiarity* prinsippet som også er mye benyttet under utviklingen av designet, ettersom dette animerte ikonet vanlig i mange andre systemer som brukeren sannsynligvis er kjent med. Dette går ut på å bruke språk og symboler som brukeren er kjent med (Benyon, 2014, s. 86)

3.6 EVALUERING/BRUKERTESTING

Evaluering er en fase i designprosessen hvor hver del av grensesnittet gjennomgår testing og evaluering av brukervennlighet (Gong et al., 2018, s. 72). Gruppen foretok en evaluering, som dette kapitlet vil ta for seg, dette inkluderer forberedelsen, gjennomføringen og resultatene.

3.6.1 Forberedelse

For å sikre at en løsning er tilstrekkelig brukervennlig og støtter de riktige oppgavene, er det lurt å prøve ut systemet på brukere (Sandnes, 2018, s. 290). Med dette som hensikt besluttet gruppen å gjennomføre brukerbaserte evalueringer (brukertester). Brukerbasert evaluering ble valgt fremfor ekspertevaluering ettersom at det ville bidra til resultater basert på målgruppens egne erfaringer og kunnskap. Den valgte metoden for brukerbasert evaluering var kooperativ evaluering, på grunnlag av den begrensede størrelsen på målgruppen, da denne metoden har som mål å samle inn mest mulig data fra hver enkelt test og dermed optimalisere resultatene (Benyon, 2014, s. 221). På denne måten kunne gruppen samle mer informasjon fra færre testsubjekter, noe som var hensiktsmessig ettersom antallet testsubjekter var begrenset.

Brukertester bør ha et klart mål og baseres på aktuelle scenarier (Sandnes, 2018, s. 294). Som nevnt tidligere ble webgrensesnittet laget for å støtte tre ulike metoder for å føre inn koordinater for ønsket treningsdata. For å evaluere brukervennligheten av disse metodene, ble brukertestene basert på tre scenarier som tok for seg hver sin metode. Brukertestene inneholdt de samme oppgavene, men var ulike i hvilken metode brukeren skulle velge for innføring av koordinater. Målet med evalueringen i dette prosjektet var å avdekke feil og forbedringspotensial som designeren selv kan overse på grunn av ens subjektive tilnærming til designet (Stull, 2018, s. 312). Fire testdeltakere ble valgt basert på deres kunnskap om prosjektet og aktiviteten som produktet var ment å støtte. Dette antallet ble bestemt ut fra antall personer som hadde tilstrekkelig kunnskap til å gi verdifulle bidrag til evalueringen.

3.6.2 Fremgangsmåte

Testene ble utført enkeltvis slik at gruppen fikk bedre oversikt og mulighet til å prioritere hver enkelt testperson og deres utførelse (Sandnes, 2018, s. 296). Testene tok sted i private grupperom for å minimalisere distraksjoner og forstyrrelser. Under hver test deltok tre gruppemedlemmer, med hver sin rolle. Et gruppemedlem hadde rollen som ordstyrer og hadde ansvar for å introdusere oppgaven og følge med på testpersonen under testen (Sandnes, 2018, s. 297). De andre to gruppemedlemmene hadde rollen som observatør. Rollen som observatør innebærer å følge med på hva testpersonen gjør, og ta notater (Sandnes, 2018, s. 29). Testpersonen utførte testen på en datamaskin koblet opp mot en større skjerm, slik at både ordstyrer og observatører kunne følge med på testpersonens utførelse.

Før testen fikk testpersonen utdelt et oppgaveark som inneholdt målet med testen, ulike begrensninger i prototypen som kunne påvirke brukeropplevelsen, og oppgavesettet som skulle utføres, disse er vedlagt i vedlegg 10. Etter testen samlet ordstyrer inn subjektive betraktninger fra testpersonen, gjennom åpne spørsmål. Dette ble gjort for å samle inn eventuell tilleggsinformasjon da den viktigste informasjonen kommer fra selve brukertesten (Sandnes, 2018, s. 299).

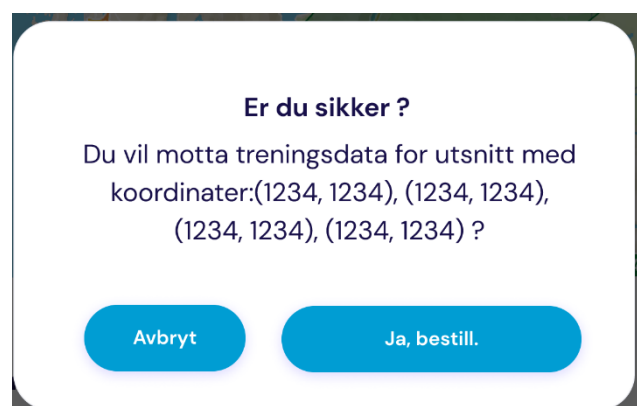
3.6.3 Resultater fra brukertesting

Resultatene fra tidsbruken under brukertestene (Vedlegg 7) viste avvik i forventet tidsbruk og den faktiske tiden brukerne brukte. De fleste avvikene skyldtes begrensninger i Figma eller prototypen, dermed var dette avvik som ikke ville forekomme i den endelige løsningen. Vi identifiserte også noen avvik grunnet misforståelser i oppgaveformuleringen, for eksempel når det gjaldt å ekspandere utsnittet. Dersom det skulle vært flere brukertester, ville vi vært mer tydelige i oppgaveformuleringen. Avvikene i tidsbruk avdekket ingen problemer i forhold til designet, og ble derfor ikke tatt til betraktning.

Resultatene fra brukertestene (Vedlegg 8) indikerte at webgrensesnittet var intuitivt og lett å bruke for personer som var kjent med geo-uttrykk og begreper. På en annen side ble det observert at personer uten tilstrekkelig kjennskap til dette fagområdet kunne oppleve webgrensesnittet som utfordrende å tolke, hvilket var en forventet observasjon. Testpersonene ga tilbakemeldinger om behovet for mer forklarende tekst og større infoknapper og skrift (Figur 13). Det ble også foreslått å endre bekreftelsesmeldingen (Figur 14) fra å inneholde inntastede koordinater til utsnittet på kart. Produkteier uttrykte et ønske om å inkludere en nedtrekksmeny med flere valg for prosentandel og muligheten til å tilpasse datasettet. Generelt sett var testpersonene fornøyde med webgrensesnittet og fant ikke oppgavene spesielt utfordrende.



Figur 13. Skjerm bilde av tekst og infoknapper på prototypen.



Figur 14. Skjerm bilde av bekreftelsesmelding i prototypen.

Funnene ble nøye vurdert i forhold til brukerbehov, produkteiers krav og tidsrammen til prosjektet. Det ble besluttet å forstørre skriftstørrelsen i den endelige løsningen etter tilbakemelding fra testpersonene. Når det gjelder infoknappene, besluttet gruppen å ikke utvikle dette i den endelige løsningen ettersom at produkteier også var brukeren av tjenesten og selv hadde etterspurt hvert felt. Andre elementer fra prototypen som ble besluttet å ikke ta med under utviklingen er fullskjermvisning av kart, forhåndsvisning av tegnet form ved bestilling, *step-indicator* og *tooltips*. På grunn av den bratte læringskurven til prosjektet og den begrensede tidsrammen, ble kun de viktigste elementene implementert. Det fremkom også i brukertesting at disse skapte mer usikkerhet i forhold til verdien det ga. Selv om vi av andre grunner besluttet å fjerne elementene som brukertestene viste problemer, var brukertesting likevel av stor verdi da den bekreftet den brukervennlige utformingen av designet.

3.7 UTVIKLINGSPROSESSEN

Kapittelet tar for seg utviklingen av webgrensesnittet for prosjektet og hvilke programmeringsspråk og verktøy som er brukt. Utviklingsprosessen blir detaljert beskrevet ved å forklare hva som er målene til prosjektet og tjenesten. Kapittelet tar også for seg utfordringer som uthenting av riktig data fra API-et og hvordan dette problemet blir løst. Til slutt, blir brukertestene diskutert for å kunne ta hensyn til å forbedre brukervennlighet for webgrensesnittet.

3.7.1 *Utviklingen av webgrensesnittet*

Som nevnt i delkapittel «3.1.2 Teknologier» besluttet gruppen å utvikle webløsningen i HTML, CSS og JavaScript med Bootstrap og Leaflet. I begynnelsen var fokuset rettet mot å forstå Leaflet og implementeringen av kart og kartfunksjoner i webgrensesnittet, ettersom at dette var nytt for gruppen. Det gikk dermed litt tid til eksperimentering med Leaflet og tilegning av forståelse av biblioteket og dets funksjoner.

Etter å ha opparbeidet en tilstrekkelig forståelse av biblioteket, var gruppen klare til å begynne med utviklingen. Den første prioriteten var å utvikle et nettsted som viste et kart ved hjelp av Leaflet. Når vi hadde fått til å vise kartet, gikk fokuset videre til å lage funksjonalitet for tegning på kartet. Her ble Leaflet Draw benyttet, som er en *plugin* av Leaflet for tegning på kart. Etter å ha implementert funksjonalitet for tegning av en form på kartet, ble fokuset rettet mot å utforme en forside og etablere en god struktur for webgrensesnittet. Vi fant Bootstrap rammeverket nyttig i denne prosessen, da det muliggjorde effektiv opprettelse av et responsivt og brukervennlig design.

Videre var fokuset å bruke JavaScript til å hente og vise koordinater til den tegnede figuren. Når dette var implementert begynte vi med neste metode for å legge inn koordinater, dette var siden for inntastning av koordinater. Dette var en mindre krevende og oppgave, da gruppen hadde noe erfaring med input-felt fra før. Under

prosjektet kom produkteier med et ønske om en tilleggs-metode for å legge inn koordinater, som var opplastning av en GeoJSON-fil. Opplastning av filer og uthenting av informasjon fra filer var et nytt område for gruppen, men etter hvert klarte gruppen å implementere dette også. Deretter ble skjermene for innstilling av begrensninger, innføring av e-post og bekreftelse utviklet. Under hele utviklingen var fokuset på å lage et nettsted som tok utgangspunkt i prototypen, men samtidig tok funnene fra brukertestene i betraktning.

3.7.2 Innhenting og lagring av data

Fokuset i begynnelsen når det gjelder backend, var utviklingen av et Python-script som via NGIS-OpenAPI hentet ut data fra SFKB og lagre dataen i en PostgreSQL-database. I prosjektet fikk denne delen navnet NGIS Retriever og baserte seg på Norkarts eksisterende kode for tilkobling og behandling av NGIS-OpenAPI. Ettersom det allerede eksisterte kode for tilkobling mot API-et, ble jobben vår å omforme denne koden slik at den hentet riktig data, midlertidig lagret dataen lokalt og lastet den opp i en PostgreSQL-database.

For å få koden til å hente riktig data ble det gjort endringer til den opprinnelige spørringen. En av disse endringene var å få spørringen til å hente ut spesifikke features, som er ulike typer data i form av bygningsdetaljer, for eksempel mønelinje, takkant og bygningsomriss. Det viste seg senere at API-et kun støttet uthenting av enten én bestemt *feature* eller alle. Dette gjør at man får for mange eller for få *features* i forhold til bestillingen. På grunn av denne begrensingen hos API-et, besluttet gruppen å ta i bruk parameteren «Bygning» som inneholder de nødvendige *features* for anvendelse av KartAi-prosjektet.

Å lagre dataen lokalt var en beslutning tatt av gruppen, ettersom det i vår mening var den mest effektive metoden å frakte dataen til databasen. På denne måten kunne Python-scriptet enkelt loope over alle *features* som ble hentet fra API-et og laste dette opp til databasen. Dette ble utført med bruk av `psycopg2` som er et av Python sine mest populære database adapter for PostgreSQL-databaser (Gregorio & Varrazzo, 2021). Bygningsdataen som ble hentet av API-et var basert på et sett med koordinater. Disse koordinatene ble bestemt i webgrensesnittet av brukeren og lagret i KartAi-delen av koden, i et polygon-format. API-et hadde kun funksjonalitet for `bbox` og koordinatene måtte derfor konverteres. Dette ble gjort gjennom henting av maksimum og minimum av x og y verdiene som da dannet en boks av det definerte området. Koordinatene ble også lagret i en annen kart-projeksjon og måtte derfor transformeres til riktig projeksjon.

For å kunne lagre dataen eksternt, fikk vi tildelt en tilkobling til KartAi sin egen database. Dette gav oss muligheten til å opprette vår egen database innenfor KartAi, men også få et innblikk i hvordan dataen måtte lagres for å kunne brukes i bachelorprosjektet. Databasen ble så lagt opp med to kolonner, en for JSON-data og en for geometri-data. *Features* data ble lagret i JSON-kolonnen og geometrien samt posisjonen til hver *feature* ble lagret i geometrikolonnen. Dermed kunne vi visualisere

dataen i QGIS, en applikasjon for GIS-systemer, som brukes til å behandle romlige data og produsere kart visualiseringer (QGIS, u.å). Videre ble databasen utvidet med to ytterligere kolonner, en for å holde styr på radene og en hvor data for kryssningspunkt mellom *features* ble lagret, hvor sistnevnte var et ønske fra produkteier.

3.7.3 Generering og nedlastning av treningsdata

For å generere treningsdata måtte gruppen ta i bruk kode fra KartAi-prosjektet. Dette ble en krevende prosess grunnet størrelsen og kompleksiteten til koden (se delkapittel «5.2.2 Kompetansegap»). Etter å ha hentet koden fra GitHub, måtte det lastes ned flere pakker som var nødvendige for at koden skulle kjøre. For å kartlegge hvilke pakker som var nødvendige, måtte vi lese feilmeldingen som oppstod når vi forsøkte å kjøre koden. Denne informerte om én nødvendig pakke av gangen, noe som førte til at gruppen måtte gjennomgå en betydelig mengde feilmeldinger og nedlastninger før koden ble kjørt på riktig måte. Videre ble det opprettet en konfigurasjonsfil, med hensikt å informere KartAi koden om hvilke datakilder som skulle benyttes. For at KartAi koden skulle vite hvilke koordinater den skulle trene på, omdefinerte gruppen en fil i KartAi-koden med hensikt og hente koordinater fra webgrensesnittet og videreformidle disse.

Tidlig i prosjektet dukket det opp en utfordring, med at man ikke kunne laste ned treningsdata via KartAi-prosjektet uten å trene KI-modellene. Dette var basert på et «*Lazy Loading*» design mønster, som vil si at ressurser ikke ble igangsatt før de faktisk trengtes med hensikt å forbedre ytelse og spare systemressurser (Wickham, 2018, s. 47-84). Dette ville skapt lang ventetid for brukeren, noe både produkteier og gruppen ikke ønsket. Problemet ble oppdaget tidlig i prosjektet, men gruppen tenkte at det kunne løses senere og prioriterte derfor andre utfordringer. Når vi hadde kommet i en av de siste sprintene, kom produkteier med ønske om at gruppen løste problemet. Han tenkte dette kunne løses enkelt ettersom det hadde vært mulig i en eldre versjon av KartAi, men etter dialog med utviklerne ble det avdekket at det ville kreve store endringer og ta lang tid. En utvikler hos KartAi klarte imidlertid å løse dette noe som ble et stort gjennombrudd i bachelorprosjektet ettersom produkteier kunne få det som ønsket.

Videre ble det testet to forskjellige WMS-servere, en under Norkarts domene og Norge i bilder under kartverkets domene. Begge tilbød et ortofoto kartlag som kunne brukes til å trene på, men de viste seg imidlertid å ha store forskjeller knyttet til funksjonalitet. Vi opplevde at Norge i bilder var gratis i bruk, men hadde begrensninger som maks antall kall per sekund. Dette skapte noen problemer for treningsdataen som ble generert. Det var også begrenset hvilke størrelser «*tilesize*» flyfotoene hadde. Grunnet disse utfordringene besluttet gruppen å benytte seg av Norkart sin WMS-tjeneste.

3.7.4 Automatisering av backend

Når koden var ferdig utviklet og funksjonaliteten til de forskjellige scriptene fungerte, ble fokuset rettet mot å oppnå automatisk kjøring av skriptene i riktig rekkefølge. Dette var nødvendig for å unngå feil ved henting av treningsdata. For å oppnå dette ble det opprettet et Python-script som tok i bruk en innebygd modul i Python som gjorde det mulig å konfigurere kjøresekvensen til scriptene. Scriptet ble koblet til en bestillingsknapp i webgrensesnittet, som initierte scriptet når den ble klikket.

Rekkefølgen ble satt opp slik at scriptet for overføring av data fra SFKB til databasen startet først, ettersom at KartAi-koden var avhengig av denne dataen. Deretter kjørte scriptet for generering av treningsdata. Til slutt ble et script for sletting av data fra databasen startet, dette var for å unngå å fylle opp databasen med unødvendige mengder data.

3.7.5 Sammenkobling av backend og frontend

Sammenkoblingen av backend og frontend-koden ble gjort gjennom FastAPI. Rammeverket ble brukt til å sende koordinatene brukeren hadde definert (i GeoJSON format), til backend-delen med en HTTP-spørring til et FastAPI *endpoint* i. Deretter ble dataen behandlet og distribuert til der den var nødvendig.

For å koble sammen frontend og backend-koden så ble FastAPI(definert i delkapittel «3.1.2 Teknologier») rammeverket brukt . Både delen for generering av treningsdata og NGIS delen for henting og lagring av data av backend-koden er avhengig av både koordinater og «*constraints*» for å fungere. Disse blir gitt av brukeren i webgrensesnittet og det var derfor nødvendig med en måte og overføre brukerens input til backend-delen. Med FastAPI sitt rammeverk sendes brukerens input, i dette tilfellet i form av JSON, med en HTTP-spørring til et FastAPI «*endpoint*» i backend-delen. Dataen blir deretter behandlet og distribuert til der den behøves.

3.7.6 Valg av metode for programvarelevering

Som nevnt tidligere krevde KartAi prosjektet installasjon av flere eksterne biblioteker for det kunne kjøres, hvilke biblioteker varierte avhengig av hvilken datamaskin man kjørte det på. Ved å bruke Docker ville alle bruke samme virtuelle maskin og prosjektet ville da fungere uavhengig av maskin.

For å få prosjektet vårt til å fungere i Docker lagde vi en Dockerfil i rotmappen til prosjektet vårt. Denne Dockerfilen spesifiserer hva et Docker-image skal inneholde. Dette kan blant annet være hvilke pakker/biblioteker som skal installeres, hvilke filer som skal kjøres i en container, og hvilke porter som skal brukes. Når Docker-imaget er bygd kan man starte en Docker-container som bruker imaget, for så å ta i bruk programvaren.

4 RESULTATET

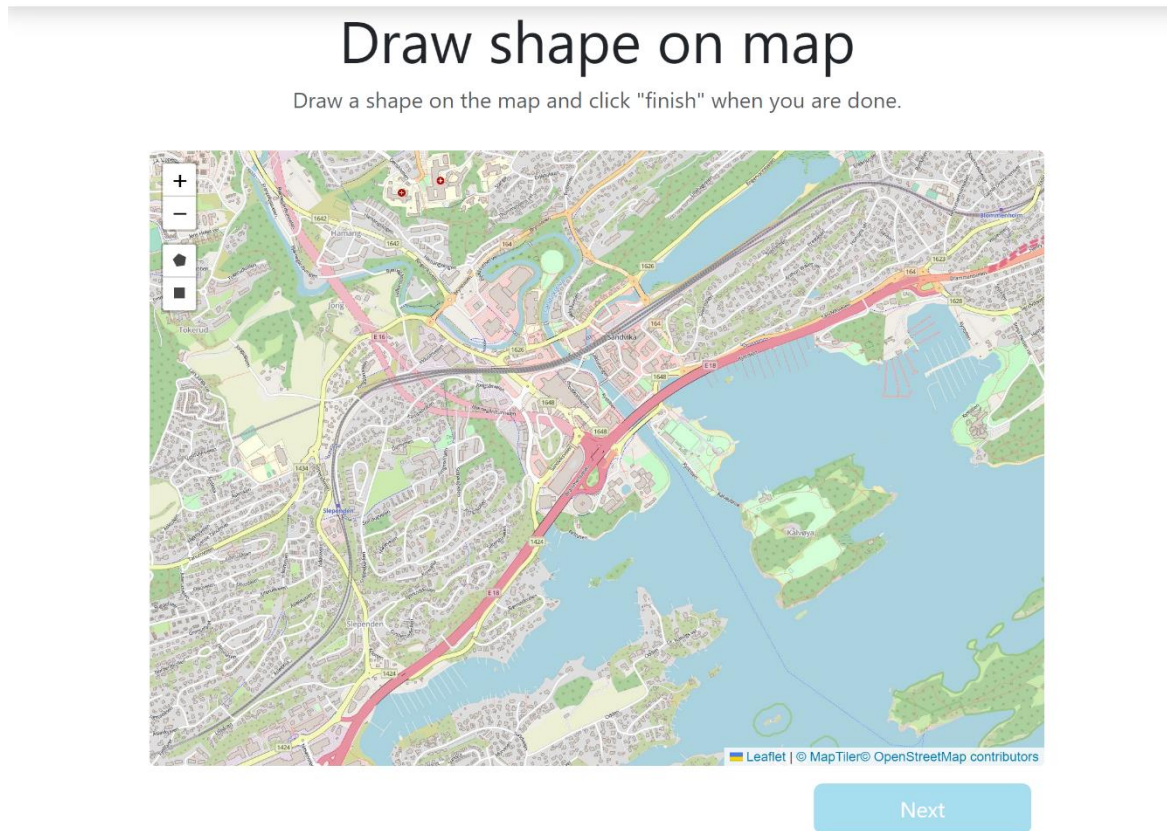
I dette kapitlet vil vi presentere resultatet som er den endelige løsningen. Vi vil beskrive hvordan systemet brukes og hvordan det fungerer. Avslutningsvis vil tjenestens og prosjektets vei videre presenteres.

4.1 ENDELIG LØSNING

Den endelige løsningen består av et webgrensesnitt hvor en bruker kan bestille treningsdata og en backend løsning som ut ifra koordinatene angitt i bestillingen genererer treningsdata for bruk i maskinlæring.

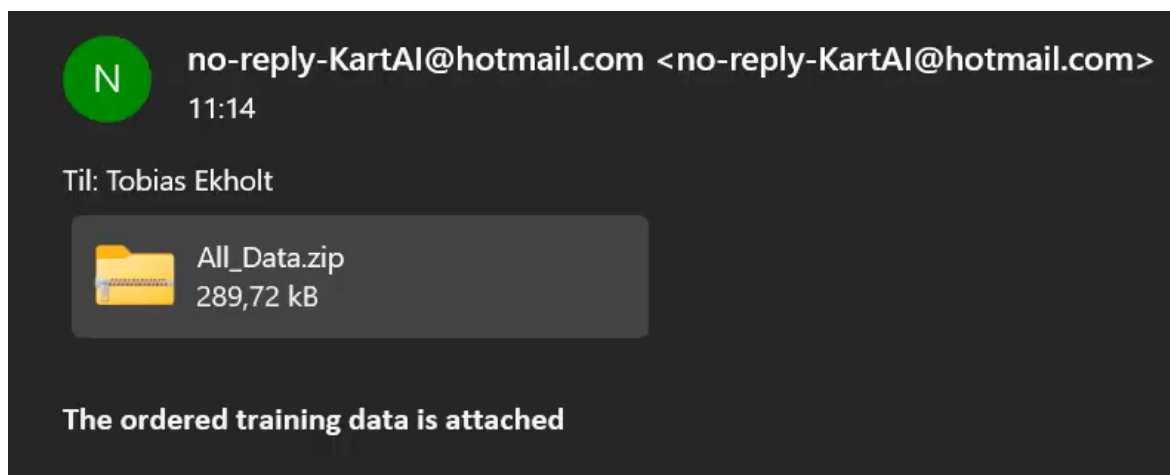
- For bestilling av treningsdata starter brukeren på en landingsside, hvor man kan velge ønsket metode for spesifisering av ønsket utsnitt, tidligere fremvist i Figur 10. Hver metode presenteres med et illustrerende ikon og tekst, og navigerer ved trykk brukeren videre til siden for utføring av valgt metode.
- En av metodene er tegning på kart, siden til dette har et kart med tegnefunksjoner, hvor brukeren kan tegne en polygon eller firkant for å spesifisere ønsket utsnitt for treningsdataen, avbildet i Figur 15. Systemet vil da ta inn koordinatene, og dette blir grunnlaget for hva datakildene skal hente.

Kart-AI Trainingdata



Figur 15. Skjerm bilde av siden for å tegne utsnitt på kart i webløsningen.

- Det neste steget er valg av regler som skal gjelde for treningsdataen og bekreftelse av generering. Når brukeren har bekreftet vises en skjerm som forteller brukeren at treningsdata genereres. Systemet henter da bygningsdata med features fra SFKB via NGIS-OpenAPI, og et ortofoto fra samme område via en WMS-server. Denne dataen vil kjøres igjennom et script i KartAi koden hvor treningsdata blir generert.
- Når dataen er generert navigeres brukeren videre til en side for bestilling, når bruker trykker på knappen for bestilling vil treningsdata bli sendt til brukeren på e-post som vist i figur 16. Deretter blir brukeren sendt til en skjerm som informerer brukeren om at data er sendt.



Figur 16. Skjerm bilde av e-post med treningsdata mottatt etter bestilling.

4.2 VEIEN VIDERE

For levering av løsningen, hadde gruppen to mulige fremgangsmåter. Vi kunne enten hoste tjenesten på en server hos Kartverket, eller kjøre tjenesten i en Docker-container. Det ble sammen med produkteier besluttet å få programvaren til å kjøre i Docker, da det ville være mindre tidkrevende.

Ettersom at løsningen er en del en KartAi prosjektet, skal det vi har utviklet benyttes videre. Av den grunn er det flere essensielle funksjoner som ikke er implementert i løsningen, for eksempel autentisering. Løsningen er derimot funksjonell for produkteiers bruk med alle nødvendige funksjoner for uthenting og generering av treningsdata. Samtidig fungerer den som et grunnlag for videreutvikling av en større og mer omfattende løsning.

5 REFLEKSJON

Dette kapittelet vil inneholde gruppens refleksjoner rundt prosjektet. Det vil reflekteres rundt ulike oppgaver i prosjektet, problemstillinger, prosjektstyring, systemer og dokumentasjon. Refleksjoner rundt hvordan bruk av scrum-rammeverket har fungert i reelt prosjekt og hva vi har lært. Det blir også diskutert rundt estimeringer av forskjellige oppgaver og hvordan gruppen forholdt seg til dette gjennom prosjektet. Til slutt, presenteres noen kvalitetssikrende tiltak som har blitt tatt i bruk som tydelig kommunikasjon og risikovurderinger.

5.1 PROSJEKTARBEID

I dette kapittelet vil vi presentere våre refleksjoner knyttet til prosjektstyring, rammeverk, aktiviteter, artefakter, og fremdrift

5.1.1 Gjennomføring av scrum

Bruken av scrum-rammeverket i et reelt prosjekt har vært lærerikt og spennende. Gruppen sitter igjen med gode erfaringer med bruk av sprinter, *sprint planning* og *sprint review*. Ikke alle scrum-aktivitetene ga like mye nytte, for eksempel *sprint retrospective*. Dette kunne vært på grunn av at vi alltid jobbet tett mot hverandre, og produkteier. Hvis oppstod noen problemer eller uenigheter, tok vi det som regel opp internt med en gang. Dermed ble problemer løst fortløpende, og vi forbedret oss kontinuerlig og hadde ikke mye å ta opp under *sprint retrospective*.

Det å jobbe i to ukers sprinter fungerte bra, ettersom det oppfordret til hyppig kommunikasjon med aktører, i form av *sprint planning* og *sprint review* annenhver uke. Dermed fikk vi tilbakemeldinger underveis i prosessen som var med på å forbedre det endelige produktet. I tillegg bidro dette til økt motivasjon, ettersom at aktørene ofte ble imponerte over fremgangen til gruppe

Estimering var en utfordring ettersom vi ikke hadde mye forståelse om hvordan oppgavene skulle løses. Gruppen prøvde å sette opp oppgaver i *sprint backlog* med estimering i forhold til kapasitetsplanen. Vi fikk anbefaling fra vår veileder hos Tietoevry om å sette opp oppgaver som kunne løses på én dag. Da vi begynte med dette, ble det litt lettere å estimere oppgavene. Det var imidlertid noen oppgaver som tok lengre enn én dag, og dermed ikke kunne brytes ned. Planning poker ble vurdert til estimeringen, men vi besluttet at det ville koste oss for mer tid enn det var verdt. Måten vi løste det var at rett etter *sprint planning* så fordelte vi oppgaver til alle medlemmer basert på interesser og kunnskap. Deretter diskuterte og estimerte gruppen i plenum hvor lang tid som skulle brukes på oppgaven, med hensyn til at alle hadde 45 timer til rådighet i hver sprint.

I starten av prosjektet hadde gruppen som nevnt lite erfaring rundt oppgavene som skulle løses. Dette gjorde det utfordrende å gi et godt estimat på hvor lang tid en oppgave ville ta. Gruppen var veldig selvbevisst på dette, og derfor ble vi enige om å doble antallet timer vi trodde en oppgave ville ta. Dette førte til at de fleste oppgavene ble noe overestimert. Gjennom prosjektet fikk gruppen mer innsikt, og estimering av oppgaver ble en enklere øvelse når erfaringen og kunnskapen økte.

5.1.2 Prosjektstyringsverktøy

Som nevnt tidligere skulle gruppen ta i bruk Kartverkets Jira som prosjektstyringsverktøy. Dette ble det i starten av prosjektet søkt om tilgang til, men gruppen var informert at det kunne ta litt tid. For å jobbe seg rundt dette ble det opprettet et eget Jira miljø midlertidig slik at prosjektstyring kunne starte i sprint 1. Gruppen var da også klar over at på et tidspunkt var det nødvendig med en migrering av Jira miljøet, slik at all dokumentasjon ville være lagret i Kartverket sitt miljø. Da tilgangen omsider kom på plass i sprint 2, måtte alle oppgaver i *sprint backlog* med tilhørende prioritering og estimerer flyttes over manuelt. Selv om denne jobben ikke var altfor tidkrevende så ble det fortsatt en utfordring tidlig i prosjektet. Det ble håndtert på en effektiv måte fordi gruppen hele tiden var klar over at dette kom til å skje på et tidspunkt. Ellers hadde gruppen gode erfaringer med verktøyet.

5.1.3 Kvalitetssikring

Kontinuerlig og tydelig kommunikasjon både med produkteier, innad i gruppen og med eksterne aktører har som nevnt tidligere vært en betydelig faktor for kvalitetssikringen i bachelorprosjektet. Det har bidratt til effektiv problemløsning og at prosjektet ble styrt på en måte som ivaretok alle interessenters ønsker og behov.

Et annet effektivt kvalitetssikrende tiltak var risikomatriksen, denne bidro til at gruppen kunne forutse og planlegge håndteringen av potensielle hindringer som kunne påvirke fremdriften i prosjektet. På denne måten når problemer oppstod hadde gruppen allerede klart skadereduserende tiltak, noe som bidro til å sikre kvalitet i prosjektet. Et eksempel på dette er risikomomentet om mangel på tilganger, noe gruppen skrev opp etter tidligere erfaring med tilgangsproblemer i prosjekter med flere aktører, systemer og interne nettverk. Dette var et problem som oppstod tidlig i prosjektet og ved flere anledninger, derfor var det veldig nyttig at gruppen allerede hadde planlagt skadereduserende tiltak for dette. Gruppen tok dermed raskt kontakt med ansvarlige for tilgang og fikk løst problemet så raskt som mulig.

Prosjektdagboken benyttet under prosjektet, viste seg også og være en verdifull ressurs, som også bidro til kvalitetssikring. Det å kunne ha en oversikt over prosjektets fremdrift og planen fremover, bidro til opprettholdelse av struktur, dermed kvalitet i prosjektet.

5.1.4 Sosial loffing

Fenomenet sosial loffing kommer fra teorien om at når en gruppe jobber sammen så er man mindre effektive i forhold til når man jobber individuelt (Svartdal, 2020). Dette er noe gruppen har erfart i prosjektet. Som nevnt tidligere har gruppen jobbet på to forskjellige steder, både hos Kartverket og hos Tietoevry. Hos Tietoevry har gruppen jobbet relativt individuelt, i motsetning til Kartverket hvor det har blitt jobbet i felleskap på et grupperom.

Vi erfarte både fordeler og ulemper med begge arbeidsmåtene. Ulempene vi erfarte ved å jobbe tett sammen var at effektiviteten og konsentrasjonen til tider kunne være lavere enn når vi jobbet individuelt, noe som kunne føre til mindre fremgang enn ønsket. Å jobbe individuelt førte ofte til mer effektivt arbeid, men det var også mindre rom for felles problemløsning og samarbeid.

For å håndtere ulempene til hver metode tilpasset vi arbeidsoppgavene etter arbeidsmiljøet, for eksempel tok vi for oss oppgaver som enkelt kunne utføres individuelt på dager hvor vi jobbet hos Tietoevry. Dette var oppgaver som lesning, skrivning og modellering, ettersom disse oppgavene ikke alltid krevde tett samarbeid. På dager hvor vi jobbet på grupperom hos Kartverket tok vi for oss oppgaver som var mer utviklingsrettet, ettersom at miljøet ga oss mer rom for samarbeid og felles problemløsning.

5.2 ERFARINGER MED TEKNOLOGIER OG UTVIKLING

Dette kapittelet tar for seg de beslutningene som ble tatt angående valg av frontend- og backend-rammeverk, samt de erfaringene gruppen gjorde seg i forbindelse med bruk av NGIS-OpenAPI og en WMS-server. Prosessen med å velge rammeverk involverte flere samtaler med produkteieren og utviklerne fra Tietoevry, noe som resulterte i grundige undersøkelser. Videre presenteres erfaringer og utfordringer knyttet til NGIS-OpenAPI, hvor positive opplevelser og utfordringer blir forklart i lys av KartAI-koden som gruppen jobbet med. Til slutt, beskrives CHAT-GPT som et verktøy for programmering, og hvordan gruppen tok i bruk dette verktøyet.

5.2.1 Valg av rammeverk

Første steg for frontend-utviklingen var valg av rammeverk og programmeringsspråk, hvor React med TypeScript først ble undersøkt, men bruk av denne tilnærmingen ville imidlertid bydd på visse utfordringer. For eksempel ville det da vært hensiktsmessig og ta i bruk biblioteket React Leaflet fremfor Leaflet JS. Gruppen mottok tidlig anbefalinger fra visse aktører om å unngå bruk av React Leaflet, ettersom at det ikke var åpen kildekode og potensielt kunne skape konflikter med prosjektet. Basert på dette i tillegg til den potensielle tidskostnaden ved å lære seg et nytt rammeverk besluttet gruppen og ikke ta det i bruk.

Videre ble rammeverket Tailwind CSS undersøkt og testet, ettersom det er et av de mest populære CSS rammeverkene (Gooding, 2020). Men etter grundig vurdering besluttet gruppen at det ville kreve for mye tid å sette seg inn i. Gruppen konkluderte med at den optimale tilnærmingen når det gjaldt rammeverk var å ta i bruk CSS-rammeverket Bootstrap. Ettersom at flere i gruppen hadde tidligere erfaring med dette ville det muligjøre raskere fremgang og produktivitet.

Under valg av backend-rammeverk for sammenkobling med frontend, ble det gjennomført undersøkelser rundt mulige Python-rammeverk som Flask og Django. Her ble det produsert noen testprosjekter innenfor alle rammeverkene for å kunne teste funksjonalitet og kunne argumentere for fordeler og ulemper mellom rammeverkene. Etter gjennomføring av testprosjektene virket Flask som et bedre alternativ for prosjektet på daværende tidspunkt, på grunn av kompleksiteten i prosjektet og enkelheten til Flask. Testingen av Django ga store muligheter, men basert på tidsressurser ville dette rammeverket ta for mye tid og kreve en dypere forståelse.

Gruppen reflekterte rundt rammeverk med produkteier og FastAPI ble tatt opp som en mulighet. Interessen for FastAPI økte ettersom det var relativt nytt og skilte seg ut i forhold til de andre rammeverkene og produkteier hadde god erfaring med det. Robustheten og det enkle omfanget av rammeverket gjorde også det enklere å koble sammen backend-koden med frontend-koden (Turing, u.å). I ettertid kan det sies at det ble brukt en betydelig mengde tid på undersøkelser rundt rammeverk som kunne ha vært gjort annerledes. Imidlertid ga dette oss verdifull dokumentasjon og muligheten til å teste flere rammeverk for å vurdere fordeler og ulemper for hvert enkelt, i forhold til spesifikasjonene for vårt prosjekt (Barguzar, 2022).

5.2.2 Kompetansegap

Ved prosjektstart var mangel på domenekunnskap og erfaring den største utfordringen, da vi ikke hadde erfaring med geografiske data og relaterte uttrykk. Vi opplevde det som utfordrende og danne et helhetlig bilde av oppgaven ut ifra all informasjonen vi fikk. Av denne grunn ble det lagt inn et punkt om dette i risikomatriksen (Vedlegg 2). Men etter en god mengde lesing, testing og research fikk vi satt oss inn i de forskjellige systemene, uttrykkene og hvordan arbeid med geografisk data ble utført.

Videre støtte vi på enkelte utfordringer med å forstå og ta i bruk KartAi koden. Dette var grunnet dets størrelse og kompleksitet og at vi ikke hadde erfaring med verken Python eller GIS. Dette var produkteier og aktører klar over og det ble derfor arrangert flere *workshops* sammen med utviklerne av koden til KartAi prosjektet. Dette var til stor hjelp, og gjorde at gruppen etter hvert klarte å ta i bruk prosjektet.

Utviklingen av løsningen var en lærerik prosess, med en bratt læringskurve. Gruppen gikk fra å ha begrenset erfaring med de ulike teknologiene og verktøyene beskrevet i rapporten, til å ha anvendt dem alle i utviklingen av det ferdige produktet.

Selv om det var en krevende prosess, ga den gruppen verdifulle erfaringer og kompetanse, som kan benyttes i fremtidige prosjekter.

5.2.3 Bruk av ChatGPT som verktøy

Som nevnt i delkapittel «3.1.2 Teknologier» besluttet gruppen å ta i bruk verktøyet ChatGPT som hjelpemiddel for programmering. Til tross for beslutningen om å bruke verktøyet var gruppen også bevisste på de potensielle problemene og hensyn som måtte tas i betraktning.

Ved økt popularitet rundt chatboten, oppstod også ulike problemstillinger rundt bruken av ChatGPT i utdanning. Universitetet i Agder og andre universiteter omtaler besvarelser skrevet ved hjelp av chatboten som juks (Hvitmyhr & Vik, 2023). Av denne grunn besluttet gruppen tidlig at chatboten kun skulle brukes som hjelpemiddel til programmering og at det under ingen omstendigheter skulle brukes til rapportskrivning eller hjelpemiddel for rapportutforming.

En faktor gruppen var oppmerksom på under bruken av verktøyet var kvaliteten på responsen man mottok, og at vi derfor ikke kunne stole blindt på denne. OpenAI informerte om at chatboten kunne generere troverdige, men feilaktige eller ukorrekte svar (OpenAI, 2022). I forhold til kode kunne den generere kode som inneholdt feil eller mangler, av den grunne burde man forholde seg kritisk til responsen (OpenAI, 2023b, s. 11). Vi sjekket derfor om ting den sa stemte med informasjon fra pålitelige kilder. Et eksempel på dette er når vi brukte det til å foreslå en bra mappe- og filstruktur til oppsettet av FastAPI, vi sjekket da om den forslåtte strukturen var en bra måte å gjøre det på ifølge andre mer pålitelige kilder. En ytterligere faktor gruppen tok i betraktning er sikkerhet, da det er en risiko for at generert kode kan være kompromittert eller ha sårbarheter (OpenAI, 2023b, s. 42). Dermed kan feilaktig bruk av chatboten bidra til å introdusere sikkerhetssårbarheter til egen kode (OpenAI, 2023b, s. 10).

5.2.4 Sentrale utfordringer

Som nevnt tidligere oppstod det gjennom prosjektet ulike problemer med tilganger, ettersom det var behov for blant annet tilgang til databaser, interne Jira systemer, kart servere, og teams servere. Den største utfordringer knyttet til tilgang, var at vi på et tidspunkt ikke fikk tilgang til en database i Azure-miljøet til Kristiansand kommune. På dette tidspunktet, ble det gjennomført mye testing og feilsøking av kode for å hente data, og dette førte til at gruppen ikke kunne teste eller feilsøke halvparten av arbeidstiden. Heldigvis var problemet løst innen en sprint, og gruppen kunne fortsette arbeidet med databasen som normalt.

Bruken NGIS-OpenAPI var en positiv opplevelse, men medførte visse komplikasjoner. Generelt opplevde vi API-et som enkelt i bruk og godt dokumentert i Swagger dokumentasjonen, men vi oppdaget noen mangler blant annet en oversikt

over hvilke EPSG som er tilgjengelig. Det var også et problem at API-et kun tok imot bbox og ikke polygoner. For vår oppgave innebærer dette at en god del mer bygningsdata blir hentet ut enn det som er nødvendig. Figur 17 illustrerer dette, hvor det rosa området som er tegnet er polygonen, mens det sorte rektangelet rundt viser bbox-området bygningsdata hentes fra. Det er heller ikke nevnt spesifikt i dokumentasjonen at polygon ikke er et gyldig format og gruppen ble derfor nødt til å oppsøke om polygon er mulig eller ikke fra utviklerne til API-et. Dette påvirket ikke funksjonaliteten til koden, men kunne bidra til å øke tiden for uthenting av data i større områder.

Videre kunne det vært hjelpsomt dersom man kunne hentet ut mer enn én spesifikk *feature* per spørring. Dette var noe API-et ikke støttet etter vår kunnskap. I vårt prosjekt hadde det å kunne filtrere ut unødvendige *features* gjort treningsdataen mer oversiktlig.



Figur 17. Illustrasjon av et utsnitt på kart i form av en *bbox*(svart) sett i forhold til et *polygon*(rosa).

6 KONKLUSJON

I denne rapporten har vi presenter gjennomføringen av bachelorprosjektet "Automatisering av dataflyt fra nasjonale kartbaser til AI-algoritmer". Hovedformålet til bachelorprosjektet var å utvikle en løsning for uthenting og generering av treningsdata for bruk i maskinlæring. Rapporten gir en grundig redegjørelse for prosessen fra prosjektstart til endelig produkt, inkludert kartlegging, planlegging, estimering, utvikling, kvalitetssikring og anvendelse av relevante metoder. Gjennom prosjektet har vi styrket vår kompetanse innenfor området kunstig intelligens og maskinlæring samt forbedret vår evne til å løse tekniske og metodiske utfordringer. Resultatet er et produkt som står til de fastsatte målene og kravene etablert fra kommunikasjonen med produkteier og aktører.

Vi anser prosjektet som vellykket basert på det endelige resultatet og tilbakemelding fra kunden/produkteier. Gjennom god planlegging, kommunikasjon og strukturert arbeid føler vi at vi har oppnådd et resultat som overgår forventningene vi hadde i starten av prosjektet. Vi håper at dette prosjektet vil bidra til fremtidige forbedringer og innovasjoner innenfor kunstig intelligens og bidra til økt kvalitet og effektivitet i behandlingen av geografiske data.

7 LITTERATURLISTE

- Atlassian. *Welcome to Jira Software*. Atlassian. Hentet 12.05.2023 fra <https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software>
- Barguzar, A. (2022, 21.04.2022). *Python Flask versus FastAPI: Which Should You Choose?* NetGuru. <https://www.netguru.com/blog/python-flask-versus-fastapi>
- Benyon, D. (2014). *Designing User Experience - A guide to HCI, UX and interaction design*. Pearson.
- Bibik, I. (2018). *How to Kill the Scrum Monster*. Apress. <http://link.springer.com/10.1007/978-1-4842-3691-8>
- Bootstrap. *Build fast, responsive sites with Bootstrap*. Bootstrap. Hentet 12.05.2023 fra <https://getbootstrap.com/>
- Codacy. (2023, 22.03.2023). *How to implement coding standards in your organization*. Codacy. <https://www.linkedin.com/pulse/how-implement-coding-standards-your-organization-codacy/>
- Commision, E. (2019). *A definition of AI: Main capabilities and scientific disciplines*. EUROPEAN COMMISSION. <https://digital-strategy.ec.europa.eu/en/library/definition-artificial-intelligence-main-capabilities-and-scientific-disciplines>
- Das, A. (2021). *An overview of PGAdmin – PostgreSQL Management Tool*. <https://www.sqlshack.com/an-overview-of-pgadmin-postgresql-management-tool/>
- EPSG.io. *EPSG.io: Find Coordinate Systems Worldwide*. Hentet 21.03.2023 fra <https://epsg.io/about>
- Eriksen, I. C. (2022). *Risikostyring i prosjekter*. Intersoft. Hentet 27.03.2023 fra <https://www.intersoft.no/risikostyring-i-prosjekter/>
- Esri. GeoJSON. esri. <https://doc.arcgis.com/en/arcgis-online/reference/geojson.htm>
- Garcia, L., Oliveira Jr, E. & Morandini, M. (2022). *Tailoring the Scrum framework for software development: Literature mapping and feature-based support*. Information and Software Technology.
- Gong, C., Qui, Y. & Zhao, B. (2018). *Establishment of Design Strategies and Design Models of Human Computer Interaction Interface Based on User Experience* (Bd. 10918). Springer International Publishing <https://doi.org/10.1007/978-3-319-91797-9>
- Gooding, S. (2020). *State of CSS 2020 Survey Results: Tailwind CSS Wins Most Adopted Technology, Utility-First CSS on the Rise*. WPTavern. <https://wptavern.com/state-of-css-2020-survey-results-tailwind-css-wins-most-adopted-technology-utility-first-css-on-the-rise>
- Granum, N. (2020). FKB. I *Store Norske Leksikon*. <https://snl.no/FKB>
- Gregorio, F. D. & Varrazzo, D. (2021). *Psycopg – PostgreSQL database adapter for Python*. Psycopg. <https://www.psycopg.org/docs/#>
- Gundersen, D. & Halbo, L. (2023). Kvalitet. I *Store norske leksikon*. <https://snl.no/kvalitet>
- Heller, M. (2022). *What is Visual Studio Code? Microsoft's extensible code editor*. <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>

- Hvitmyhr, B. L. & Vik, I. (2023, 19.01.2023). Ulike regler for studentens bruk av ChatGPT: Kan endre eksamen. VG. <https://www.vg.no/nyheter/innenriks/i/4oLeVa/regler-for-bruk-av-chatgpt-paa-eksamen-og-innlevering>
- Innes, M., Karpinski, S., Shah, V., Barber, D., Saito Stenetorp, P., Besard, T., Bradbury, J., Churavy, V., Danisch, S. & Edelman, A. (2018). *On machine learning and programming languages*. Association for Computing Machinery (ACM). <https://discovery.ucl.ac.uk/id/eprint/10051391/1/37.pdf>
- Kadenic, M. D., Koumaditis, K. & Junker-Jensen, L. (2023). Mastering scrum with a focus on team maturity and key components of scrum. *Information and Software Technology*, 153. <https://doi.org/10.1016/j.infsof.2022.107079>
- Karlsen, J. T. (2021). *Prosjektledelse : fra initiering til gevinstrealisering* (5. utgave. utg.). Universitetsforlaget.
- Kartverket. (2023, 17.01.2023). *Teknisk håndbok for NGIS-plattformen*. Kartverket. <https://www.kartverket.no/geodataarbeid/ngis/teknisk-handbok#heading-15474>
- Køster, C. (2023). *Scrum - et enkelt rammeverk for komplekst arbeid*. <https://www.smidigbloggen.no/scrum-et-enkelt-rammeverk-for-komplekst-arbeid>
- Leaflet. (21.09.2021). *an open-source JavaScript library for mobile-friendly interactive maps*. Leaflet. <https://leafletjs.com/>
- Lutkevich, B. (02.2023). *GitHub*. TechTarget. <https://www.techtarget.com/searchitoperations/definition/GitHub>
- Mæhlum, L. (2021). Ortofoto. I *Store norske leksikon*. SNL. <https://snl.no/ortofoto>
- Mæhlum, L. (2023). WMS - Web Map Service. I *Store Norske Leksikon*. Hentet 09.03.2023 fra <http://snl.no/WMS - Web Map Service>
- McKenna, D. (2016). *The Art of Scrum*. Apress. <https://doi.org/10.1007/978-1-4842-2277-5>
- MDN Web Docs. (16.04.2023). *CSS: Cascading Style Sheets*. MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/CSS>
- MDN Web Docs. (10.05.2023). *HTML: HyperText Markup Language*. <https://developer.mozilla.org/en-US/docs/Web/HTML>
- MDN Web Docs. (10.05.2023). *The web and web standards*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards
- Microsoft. *Hva er PostgreSQL?* <https://azure.microsoft.com/nb-no/resources/cloud-computing-dictionary/what-is-postgresql/>
- Norkart. *Om oss*. Hentet 18.04.2023 fra <https://www.norkart.no/om-oss/>
- Nossum, A. S., Nuland, T. & Gyland, L. F. (2021). *Om KartAI*. Hentet 14.03.2023 fra <https://kartai.no/om-oss/>
- OpenAI. (2022, 30.11.2022). *Introducing ChatGPT*. OpenAI. <https://openai.com/blog/chatgpt>
- OpenAI. (2023a). *GPT-4*. <https://openai.com/research/gpt-4>
- OpenAI. (2023b). *GPT-4 Technical Report*. OpenAI. <https://arxiv.org/pdf/2303.08774.pdf>
- Ørstavik, E. & Mæhlum, L. (2022). Geografisk informasjonssystem (GIS). I *Store Norske Leksikon*. SNL. [https://snl.no/geografisk informasjonssystem - GIS](https://snl.no/geografisk_informasjonssystem - GIS)
- Perminova, M. (24.03.2023). *What is Miro?* <https://help.miro.com/hc/en-us/articles/360017730533-What-is-Miro>

- Postgresql. *What is PostgreSQL?* postgresql. Hentet 12.05.2023 fra <https://www.postgresql.org/about/>
- QGIS. *QGIS - The Leading Open Source Desktop GIS*. Hentet 21.03.2023 fra <https://www.qgis.org/en/site/about/index.html>
- Raknes, S. P. (2019). *Usikkerhetsstyring – muligheter og trusler*. Metier. Hentet 27.03.2023 fra <https://www.prosjektbloggen.no/usikkerhetsstyring-muligheter-og-trusler>
- Ranjan, R. (2021, 07.10.2021). *What is a Framework in Programming & Why You Should Use One*. Net Solutions. <https://www.netsolutions.com/insights/what-is-a-framework-in-programming/#:~:text=The%20purpose%20of%20a%20framework,programmings%20time%2C%20and%20simplifies%20testing>
- Regjeringen. (u.å). *Kartverket*. Hentet 20.04.2023 fra <https://www.regjeringen.no/no/dep/kdd/org/etater-og-virksomheter-under-kommunal--og-distriktsdepartementet/underliggende-etater/kartverket/id85701/>
- Rossen, E. & Nätt, T. H. (2022). API. I *Store Norske Leksikon*. Hentet 20.03.2023 fra <https://snl.no/API>
- Salimi, S. *Epic Stories / Epic*. Agile Academy. <https://www.agile-academy.com/en/agile-dictionary/epic-stories-epics/>
- Sandnes, F. E. (2018). *Universell utforming av IKT-systemer brukergrensesnitt for alle*. Universitetsforlaget.
- Schwaber, K. & Sutherland, J. (2020). *The Scrum Guide*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Spinellis, D. (2012). Git. *IEEE*, 29(3), 100-101. <https://doi.org/10.1109/MS.2012.61>
- Staiano, F. (2022). *Designing and prototyping interfaces with Figma*. Packt Publishing. https://books.google.no/books?hl=en&lr=&id=GOBeEAAAQBAJ&oi=fnd&pg=PP1&dq=prototyping+tools+for+user+interface+design+figma&ots=ekfqoMaT7M&sig=IZw8T1uaDLxFgNRTsT8u_Xn9v48&redir_esc=y#v=onepage&q&f=false
- Stavrinoudis, D. & Xenos, M. (2008). *Comparing internal and external software quality measurements*. *Frontiers in Artificial Intelligence and Applications*,
- Stull, E. (2018). *UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, Writers, Designers, and Developers*. Apress. <https://doi.org/10.1007/978-1-4842-3811-0>
- Subramanyam, V. S. (2021). *Basics of Bounding Boxes*. Analytics Vidhya. <https://medium.com/analytics-vidhya/basics-of-bounding-boxes-94e583b5e16c>
- Svardal, F. (2020). Sosial loffing. I *Store Norske Leksikon*. SNL. https://snl.no/sosial_loffing
- Svekis, L. L., Putten, M. v. & Percival, R. (2021). *JavaScript from Beginner to Professional: Learn JavaScript quickly by building fun, interactive, and dynamic web apps, games, and pages* (1, Red.). Packt Publishing.
- Teubner, T., Flath, C. M., Weinhardt, C., Aalst, W. v. d. & Hinz, O. (2023). Welcome to the Era of ChatGPT et al. *Business & Information Systems Engineering*, (65), 95–101. <https://doi.org/https://doi.org/10.1007/s12599-023-00795-x>
- The New Company. (2020). *Miro – virtuelle workshops og brainstorming i en hverdag med hjemmekontor*. The New Company.

- <https://www.thenewcompany.no/post/miro-virtuelle-workshops-og-brainstorming-i-en-hverdag-med-hjemmekontor>
- Tietoevry. *Om oss*. Hentet 18.04.2023 fra <https://www.tietoevry.com/no/om-oss/om-tietoevry/>
- Torgersen, O. (2018). *Kvalitet*. NKI Forlaget. Hentet 10.03.2023 fra <https://ndla.no/article/1667>
- Tsitoara, M. (2020). Version Control Systems. I *Beginning Git and GitHub* (s. 3-18). Apress. https://doi.org/10.1007/978-1-4842-5313-7_1
- Turing. *Python FastAPI vs Flask: A Detailed Comparison*. Turing. Hentet 28. April 2023 fra <https://www.turing.com/kb/fastapi-vs-flask-a-detailed-comparison>
- Visual Studio Code. *Getting Started*. Visual Studio Code. Hentet 12.05.2023 fra <https://code.visualstudio.com/docs>
- Wickham, M. (2018). Lazy Loading Images. I *Practical Android* (s. 47-84). Apress. https://doi.org/10.1007/978-1-4842-3333-7_3
- Wiik, E. *Slik kan du bruke Miro til digitale workshops*. Bouvet. Hentet 12.05.2023 fra <https://www.bouvet.no/bouvet-deler/slik-kan-du-bruke-miro-til-digitale-workshops>
- Zolkifli, N. N., Ngah, A. & Deraman, A. (2018). Version Control System: A Review. *Procedia Computer Science*, 135, 408-415. <https://doi.org/10.1016/j.procs.2018.08.191>

8 VEDLEGG

VEDLEGG 1: GRUPPEKONTRAKT

GRUPPEKONTRAKT

Gruppens deltakere:

Ine Mari Bredesen
Kasper Johan Idland Skjæveland
Niklas Fugledal
Sondre Slettene
Tobias Ekholt

Etter første prosjektmøte 10.01.2023 kom gruppen frem til følgende retningslinjer for det videre arbeid:

Mål: Målet for prosjektet, er å levere et vellykket og godt prosjekt som fører til toppkarakter på bacheloroppgaven.

Gruppens regler:

- Alle avgjørelser skal gjøres i fellesskap.
- Gruppen utarbeider et eget dokument med retningslinjer for rapportskrivning som skal følges
- Ved uenighet skal avstemming benyttes. Flertallet bestemmer
- Alt produsert materiale skal ha backup, og korrekturleses av et annet gruppe medlem.
- Informasjon til gruppedeltakere utenom møter skal sendes på messenger.
- Møteplikt på alle møter, unntak ved gyldig fravær med godkjenning av de øvrige gruppedeltakere.
- Hvis man sitter fast på en oppgave, eller ikke kan levere til oppsatt tid, skal de øvrige gruppe medlemmene informeres så tidlig som mulig slik at hjelp kan oppsøkes.
- Kun produsere originalt innhold, plagiat er uakseptabelt.
- Alt innhold må være i takt med akademiske regler, og følge reglene i APA 7.
- Alle gruppe medlemmer er pliktige til å delta (fysisk eller digitalt) på alle daily standup mellom 8-16 uavhengig av fravær. Kan bruke skjønn ved spesielle forhold.

- Hvert enkelt gruppemedlem plikter å fullføre tildelte oppgaver til oppsatt tid. Dersom dette ikke lar seg gjøre skal de øvrige i gruppen informeres så tidlig som mulig.
- Brudd på denne kontrakten vil medføre en muntlig og skriftlig til vedrørende fra resten av gruppen og veileder/emneansvarlig vil bli informert.
- Gjentatte overtredelser kan medføre at vedrørende gruppemedlem ekskluderes fra gruppen.
- Regelbrudd som ikke er avklart med resten av gruppen gis i første omgang en muntlig advarsel, og blir dokumentert i møtereferat. Andre gangs regelbrudd blir gitt skriftlig advarsel. Ved tredje gangs regelbrudd vil gruppen søke å ekskludere vedkommende fra gruppen.

- Ansvarsfordeling som følger:

Sondre Slettene, Scrum master, altnuligmann

Tobias, Utvikler, GitHub ansvarlig

Ine Mari Bredeesen, Utvikler & UX-designer

Niklas Fugledal, Utvikler & Produkteier kontaktperson

Kasper Johan Idland Skjæveland, Utvikler,

- Prosjektleder har ansvar for:

Å kalle inn til møter

Delegering av oppgaver



 Ine Mari Bredeesen



 Kasper Johan Idland Skjæveland



 Niklas Fugledal



 Sondre Slettene



 Tobias Ekholt

2

Sted og dato: Kristiansand, 20.01.23

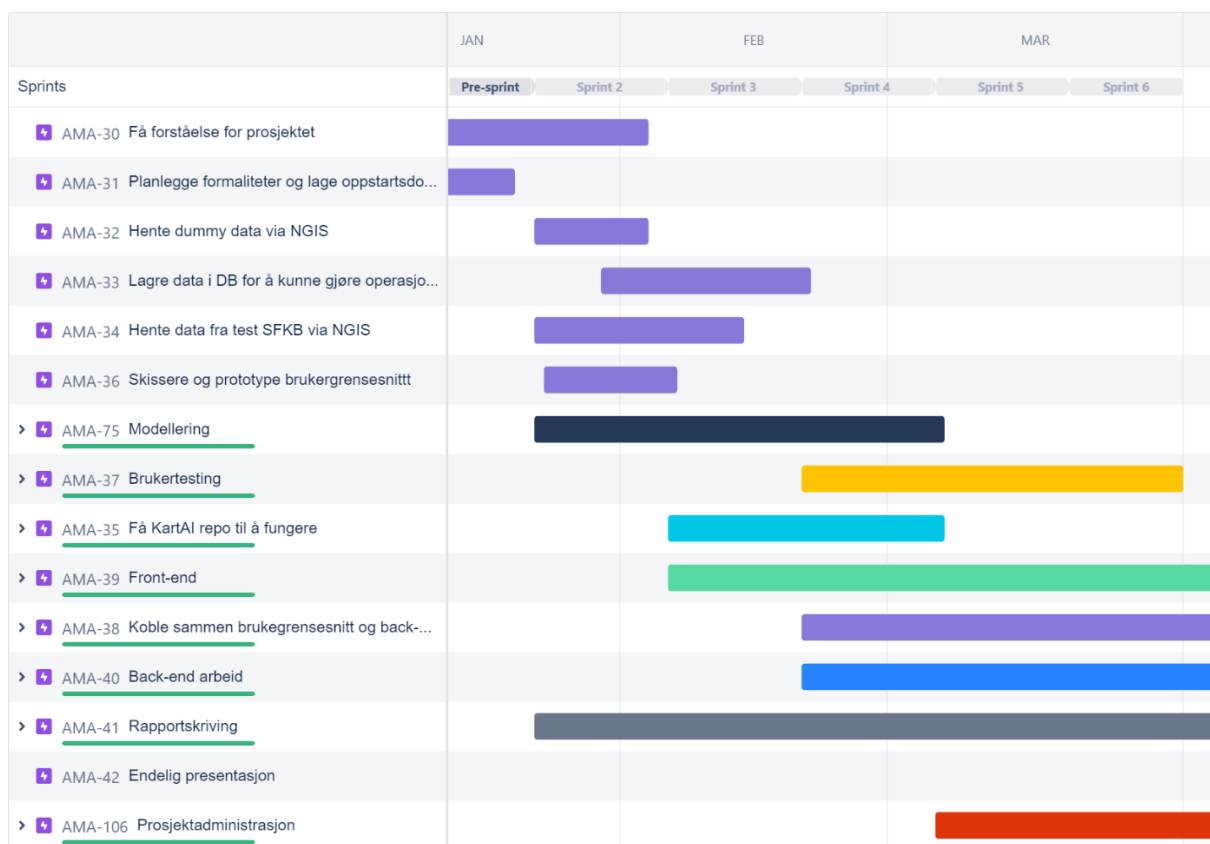
VEDLEGG 2: RISIKOMATRISE

[Link til risikomatrise her](#)

VEDLEGG 3: DAGBOK

[Link til dagbok her](#)

VEDLEGG 4: ROADMAP



VEDLEGG 5: PROTOTYPE

[Link til prototype her.](#)

VEDLEGG 6: SKISSER

1

Bestill treningsdata

Velg på kart Skriv inn kordinater

2

Velg utsnitt

Format
Velg format ▼

Tekst

Tekst

Se utsnitt Bestill

2

Bestilling

43543.23423
23432.23423.23

Bestill

3

Bestilling

Format
Velg format for treningsdata ▼

E-post
Skriv inn epost

Tilbake Bestill treningsdata

VEDLEGG 7: DATA FRA BRUKERTEST: TIDSBRUK

Tidsbruk

Test ID	Oppgavesett	Oppgave	Estimert tid (under)	Tidsbruk (under)	Avvik
01	Last opp fil	Velg metode	5.00	7.00	2.00
		Fil	15.00	16.00	1.00
		Ekspander utsnitt	10.00	28.00	18.00
		Fyll inn skjema	50.00	16.00	-34.00
		Bestill data	5.00	9.00	4.00
02	Tegn polygon	Velg metode	3	3	0
		Tegn polygon	30	60	30
		Ekspander utsnitt	20	10	-10
		Fyll inn skjema	50	20	-30
03	Skriv inn koordinater	Velg metode	5	18	13
		Skriv inn koordinater	45	76	31
		Ekspander utsnitt	10	10	0
		Fyll inn skjema	50	39	-11
		Bestill data	5	13	8
04	Tegn polygon	Velg metode	5	5	0
		Tegn polygon	50	53	3
		Ekspander utsnitt	10	82	72
		Fyll inn skjema	50	60	10
		Bestill data	5	4	-1

VEDLEGG 8: DATA FRA BRUKERTEST: SPØRSMÅL

Spørsmål	Svar 01	Svar 02	Svar 03	Svar 04	Nøkkelpunkter
Opplevde du oppgavene som vanskelige eller lette?	Veldig enkle	Alt gikk fint	Helt enkelt	Oppgavene var vanskelig å tolke for folk som ikke kjenner til geo verden med tilhørende ord og kontekst	Forståelig brukergrensesnitt, dersom man kjenner til geo-uttrykk og begreper.
Var det noe du ble usikker på eller noe som ikke var åpenbart?	Nei, eventuelt zoom i stedet for utsnitt	Litt usikker på constraints og hva dette var	Brukte litt tid på info og liten skrift på hjelpetekst	Ekspander utsnitt ble tolket som rediger polygon	Språket burde være mer tilpasset bruker og definisjoner må være tydelige. Infoknapper og skrift burde være større.
Er det noe du hadde ønsket skulle vært eller fungert annerledes?	Ikke som han kommer på	Ønsket en nedtrekksmeny med pare prosent ikke "10% bygning"	Nei, veldig fornøyd	Mer forklarende tekst i infobokser, kunne gjerne endret siste bekreftelsen med bilde i stedet for koordinater om man tegner et polygon	Vurdere å vise bruker bilde av utsnitt ved bekreftelse på utsnitt istedet for koordinater
Har du noen andre kommentarer?	Nei	Ønsket å kunne velge features fra dataset	nei	Nei	Ønskelig med input-felt: features

VEDLEGG 9: KRAVSPESIFIKASJON

Akseptansekriterier

Tittel	Valg av metode
Verdierklæring	Som en bruker vil jeg ha mulighet til å velge hvilken metode jeg vil bruke til å legge inn koordinater så jeg kan bruke metoden som passer mitt tilfelle best.
Akseptansekriterier	<ul style="list-style-type: none"> - Det er en landingsside som viser metodene for å legge inn koordinater - Hver metode linkes videre til gjennomføringen siden for å legge inn koordinater med valgt metode - Hver metode har et navn og et illustrerende ikon
Tittel	Tegning på kart
Verdierklæring	Som en bruker vil jeg kunne tegne et polygon/en firkant på et kart, for å spesifisere området jeg vil ha treningsdata for.

AkseptansekrITERIER	<ul style="list-style-type: none"> - Det er en side som viser et kart - Brukeren kan tegne en polygon eller firkant på kartet - Koordinatene hentes ut fra det tegnede utsnittet, lagres i systemet og vises på skjermen
Tittel	Laste opp fil
Verdierklæring	Som en bruker vil jeg kunne laste opp en GeoJSON-fil, for å spesifisere området jeg ønsker treningsdata fra.
AkseptansekrITERIER	<ul style="list-style-type: none"> - Det er en side hvor brukeren kan laste opp en GeoJSON-fil med koordinater - Koordinatene blir lagret i systemet - Skjermen har et kart som etter filen er lastet opp endres til å vise en polygon fra koordinatene i filen
Tittel	Skrive inn koordinater
Verdierklæring	Som en bruker vil jeg kunne skrive inn koordinater, for å spesifisere området jeg ønsker treningsdata fra.
AkseptansekrITERIER	<ul style="list-style-type: none"> - Det er en side for innskrivning av koordinater - Det er knapp for å bekrefte at man er ferdig med å skrive inn koordinatene - Kunne definere EPSG av koordinatene som blir lagt inn
Tittel	Constraints
Verdierklæring	Som bruker vil jeg spesifisere hvor mye av fraksjonen treningsdata som er trening og validering, slik at maskinlæringsmodellen skal oppnå et resultat som er pålitelig
AkseptansekrITERIER	<ul style="list-style-type: none"> - Det er en side med input-felt for spesifisering av <i>Constraints</i> - Siden har et inputfelt for hver <i>Constraint</i> - Dataen som er lagt inn i hvert input felt blir lagret for bruk videre i prosessen
Tittel	Prosentandel bygning
Verdierklæring	Som bruker vil jeg kunne spesifisere minimum mengde av det valgte treningsområdet som skal være bygning, slik at KI-modellen min faktisk skal få noe data å trene på
AkseptansekrITERIER	<ul style="list-style-type: none"> - En input felt hvor brukeren kan spesifisere bygningsprosent - Data fra input feltet lagres etter det er skrevet inn for videre bruk i prosessen
Tittel	Oppdateringer om generering av treningsdata
Verdierklæring	Som en bruker vil jeg få vite når treningsdataen genereres, slik at jeg er oppdatert på hva som skjer
AkseptansekrITERIER	<ul style="list-style-type: none"> - Visuelle elementer og tekst forteller brukeren at systemet jobber med generering av treningsdata
Tittel	Bestilling

Verdierklæring	Som en bruker vil jeg kunne legge inn mailen jeg vil ha treningsdata på, slik at jeg kan motta dataen på mail.
Akseptansekriterier	<ul style="list-style-type: none"> - En side med input-felt for mail-adresse - Vise brukeren hvilke filer som vil bli sendt
Tittel	Bekreftelse på at data er sendt
Verdierklæring	Som en bruker vil jeg få beskjed når prosessen er ferdig og data er sendt til eposten min, slik at jeg vet at jeg kan gå til mailen min og hente dataen
Akseptansekriterier	<ul style="list-style-type: none"> - En side som bruker tekst og visuelle elementer for å fortelle brukeren at prosessen er ferdig og at brukeren mottar treningsdata på e-post
Tittel	Motta treningsdata
Verdierklæring	Som en bruker vil jeg motta treningsdata fra det valgte utsnittet på e-post, slik at jeg kan ta i bruk treningsdataen
Akseptansekriterier	<ul style="list-style-type: none"> - Brukeren får tilsendt en mappe med alle genererte treningsdatafiler
Tittel	Bruke NGIS-OpenAPI for uthenting av bygningsdata fra SFKB
Verdierklæring	Som bruker vil jeg at min data skal komme fra SFKB slik at jeg er sikker på at det er riktig i forhold til matrikkelen
Akseptansekriterier	<ul style="list-style-type: none"> - Script for uthenting av bygningsdata ved bruk av NGIS-OpenAPI - Gjøre bygningsdata klar for bruk av KartAi
Tittel	Ta i bruk KartAi for generering av treningsdata
Verdierklæring	Som bruker vil jeg ha treningsdata slik at jeg kan trene mine KI-modeller
Akseptansekriterier	<ul style="list-style-type: none"> - Få KartAi til å generere treningsdata - Kunne distribuere denne dataen til brukeren -
Tittel	<ul style="list-style-type: none"> - Brukervennlig grensesnitt
Verdierklæring	<ul style="list-style-type: none"> - Som bruker ønsker jeg et webgrensesnitt som er enkelt å bruke, slik at jeg enkelt kan bestille treningsdata
Akseptansekriterier	<ul style="list-style-type: none"> - Webgrensesnittet er brukertestet og eventuelle forbedringsområder er forbedret i endelig løsning

VEDLEGG 10: DATA FRA BRUKERTEST: OPPGAVESETT

[Link til oppgavesett her](#)



UTTALELSE FRA OPPDRAGSGIVER

TietoEVRY 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. TietoEVRYs 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 Tobias Ekholt, Kasper Johan Idland Skjæveland, Niklas Fugledal, Ine Mari Bredesen, og Sondre Slettene fra UiA gjennomførte et bachelorprosjekt hos TietoEVRY 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 web baserte løsningen som gruppen utviklet var basert på en backend i Python hvor det ble benyttet FastAPI, NGISopenAPI, PostgreSQL og Geospatial Database. Frontend ble prototypet ved hjelp av bl.a. Figma og utviklet med JavaScript, HTML/CSS, Bootstrap og Leafletts, for å nevne noen av de viktigste teknologiene.

Omfang

Hver av studenten 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 pr student.

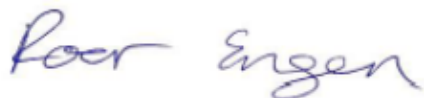
Resultat

Studentene har jobbet med teknologi og domene som i veldig stor grad var ukjent for dem ved starten av semesteret. De har jobbet målbevist og godt og endt opp med å levere mer enn forventet i opprinnelig scope. Studentegruppa 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 mer samarbeidet og resultatene som studengruppa har levert. Det kan også nevnes at studentgruppa har stilt villig opp og holdt flere presentasjoner i geodata miljøet på Sørlandet og nasjonalt underveis i prosjektet, et eksempel er beskrevet i denne saken fra Kartverket: <https://kartverket.no/om-kartverket/fylkeskartkontorene/kartverket-agder/nyheter/2023/februar/inspirerende-faglunsi>

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

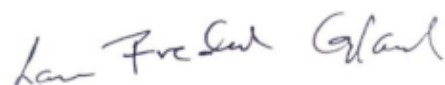


Roar Engen

Manager

Tietoevry Create Kristiansand

Kartverket



Lars Fredrik Gyland

Fylkeskartsjef/avdelingsdirektør

Kartverket i Agder

VEDLEGG 12: SELVEVALUERING

Sondre Slettene:

Jeg føler vi har gjennomført et veldig spennende prosjekt, og jeg føler vi har fordelt oppgaver på en god måte. Min rolle i prosjektet var scrum master, og mine oppgaver var knyttet til å følge scrum rammeverket, kommunisere med veiledere, avholde og

lede møter, lage modeller og skrive rapport. Mine personlige egenskaper stiller ikke veldig sterkt i forhold til programmering i front og backend, men allikevel er jeg god på å finne ut hvordan problemer kan løses med kode uten å skrive det. Derfor ønsket jeg å utnytte mine styrker som er ledelse, holde kontroll og problemløsning, og dette føler jeg er godt utført. Jeg er stolt over det endelige resultatet vi har oppnådd i forhold til hvor usikre vi var de første dagene, da alt føltes umulig. Vi har vært gode til å holde kontroll i prosjektet, og har aldri vært bakpå i forhold til å levere det vi skal, noe jeg mener grunner i god planlegging og struktur.

Tobias Ekholt:

Når jeg så oppgaven for første gang ble jeg veldig spent på hvordan vi skulle klare å få et ferdig produkt, da alt var helt nytt for meg. Etersom tiden gikk, og prosjektet til slutt ble ferdig, er jeg veldig glad for oppgaven vi fikk utdelt. Jeg føler at jeg har lært veldig mye på kort tid, og er stolt over resultatet vi endte med. Jeg har hovedsakelig hatt fokus på backend-arbeidet i prosjektet. Her har jeg blant annet bidratt med å lage script som overfører GeoJSON-fil inn i PostgreSQL-databasen, satt meg inn i KartAi-koden og laget config-fil som henter data fra riktig database og WMS-server. Jeg har også bidratt med sammenkobling mellom frontend og backend. Har hatt ansvar for at GitHub brukes riktig. Jobbet med å lage script som starter flere prosesser i riktig rekkefølge. Jeg har også hatt ansvar for å få prosjektet til å kunne kjøre i Docker. Jeg synes gruppen har vært veldig flinke til å samarbeide og fordele arbeid. Når problemer og utfordringer har oppstått har alle sammen bidratt med å hjelpe slik at ingen skulle stå fast. Alt i alt er jeg veldig fornøyd med sluttresultatet, og samarbeidet i gruppen.

Niklas Fugledal:

Under første møtet med oppdragsgiver ble jeg overveldet av oppgaven. Det virket til å begynne med som en utrolig vanskelig oppgave ettersom jeg ikke hadde noe erfaring innenfor geomatikk fra tidligere. Etter gode samtaler og kurs ble oppgaven mer tydelig og jeg fikk satt meg godt inn i teorien og fikk en forståelse for hvordan gruppen kunne få dette til. Mine personlige egenskaper er i stor grad innenfor programmering og kunne opparbeide god forståelse for teknologier. Min rolle i prosjektet var å jobbe med backend-programmering, samt ha god kontakt med produkteier for å opprettholde krav til produktet. Jeg tok ansvar for å sette opp database og rollen som ansvarlig å finne et rammeverk for sammenkobling av webgrensesnittet og backend-koden. Dette gjorde jeg ved bruk av FastAPI etter flere undersøkelser med flere forskjellige rammeverk for å finne det som var best egnet for vårt prosjekt. Jeg er fornøyd med hvordan gruppen har samarbeidet gjennom hele prosjektet i krevende og mindre krevende situasjoner og sørget for god flyt. Vi har vært effektive og jobbet mer enn det som er kravet, både for å lykkes med et godt produkt, men også for å vise hva denne gruppen kan gjøre sammen.

Kasper:

I utgangspunktet var min oppgave i dette prosjektet å jobbe i backend. Etter at prosjektet begynte å ta form ble derimot min rolle endret til en mer full-stack tilnærming. Å jobbe med både front- og backend har vært en utrolig lærerik og gøy opplevelse. Jeg har fått erfaringer innen en rekke programmeringsspråk og teknologier og føler meg mer rustet for arbeidslivet. Som gruppe mener jeg at vi har fungert meget bra. Vi har alle hatt våre roller og interesser og spilt hverandre gode. Det har vært lite interne konflikter og vi har hatt mange interessante og lærerike diskusjoner. Samarbeidet med veilederne fra Tietoevry og Kartverket har også fungert utrolig bra. De har kommet med mange gode innspill og vært til stor hjelp for prosjektet. Jeg håper kommende bachelor-grupper også kan få gleden av å jobbe med disse dyktige bedriftene.

Ine Mari Bredesen

Ved prosjektstart var det overveldende med mye informasjon og teknologier som skulle læres. Med stort ansvar innen frontend-utviklingen var det mye jeg måtte lære, jeg hadde erfaring med HTML og CSS, men Javascript hadde jeg kun satt meg litt inn i tiden før prosjektstart. Leaflet og å implementere kart og kartfunksjoner på et nettsted var helt nytt. UX/UI-design og Figma var noe jeg hadde mye erfaring med så denne prosessen var lite utfordrende. Prosjektet hadde en bratt læringskurve og krevet mye arbeid, men har bidratt betydelig til å utvide min kompetanse innen frontend utvikling samt min forståelse av maskinlæring og kraftbransjen. Det har også vært en spennende prosess, og et interessant prosjekt å jobbe med, som folk har vist stor interesse for. Samarbeidet har fungert veldig bra, ed en dedikert innsats fire dager i uken gjennom hele semesteret. Som et resultat sitter vi igjen med verdifull kunnskap og erfaring samt et produkt både vi, produkteier og aktørene er fornøyde med.