



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Båtreise Analyse Verktøy

Vessel Voyage Analysis Tool

Glenn Robert Strandos – 181206

Lars Erik Kleiven - 181183

Dataingeniør

Institutt for data- og realfag

Avdeling for ingeniør og økonomi

Innleveringsdato: 03.06.2019

Antall ord: 12 338

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 10.

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Vessel Voyage Analysis Tool	<i>Dato:</i> 03.06.2019
<i>Forfatter(e):</i> Glenn Robert Strandos og Lars Erik Kleiven	<i>Antall sider u/vedlegg:</i> 44
<i>Studieretning:</i> Dataingeniør	<i>Antall sider vedlegg:</i> 5
<i>Kontaktperson ved studieretning:</i> Bjarte Kileng	<i>Antall disketter/CD-er:</i> 0
<i>Merknader:</i> Ingen	<i>Gradering:</i> Ingen

<i>Oppdragsgiver:</i> Dataloy Systems AS	<i>Oppdragsgivers referanse:</i> Ingen
<i>Oppdragsgivers kontaktperson:</i> Bernard A. Hafting	<i>Telefon:</i> 93023640

Sammendrag:

Dette prosjektet tar for seg utviklingen av Vessel Voyage Analysis Tool, på vegne av Dataloy Systems AS. Denne applikasjonen skal eventuelt kunne erstatte lignende applikasjoner, og integreres i Dataloy sine systemer, for å gjøre jobben med å overvåke fartøy enklere for kundene til Dataloy. Applikasjonen tar i bruk både AIS-signal for å posisjonere fartøyene, samt værdata for å gi relevant informasjon angående status på fartøyene.

I denne rapporten vil det bli grundig gjennomgått hvordan prosjektet ble gjennomført, og hvordan resultatet ble. Det vil bli mer forklart hvordan applikasjonen fungerer i praksis.

Stikkord:

TypeScript	React	Redux
------------	-------	-------

I. Forord

Denne rapporten beskriver arbeidet som er gjort i forbindelse med bachelorprosjektet Vessel Voyage Analysis Tool. Prosjekteier for prosjektet er Dataloy Systems. I rapporten har gruppen beskrevet prosessen med å lage en modul, eller applikasjon, som på sikt skal kunne integreres i Dataloy sitt VMS (Voyage Management System). Rapporten er skrevet av Glenn Robert Strandos og Lars Erik Kleiven. Prosjektet er gjennomført av de samme personene.

Uten hjelp og veiledning ville ikke prosjektet og rapporten sett ut slik det gjør i dag. Det er derfor viktig for oss å takke de som har bistått underveis. Det rettes en stor takk til Bjarte Kileng som har vært mentor i forhold til rapportskrivning og som har hjulpet oss delvis med planlegging/status av prosjektet. Det rettes en stor takk til de ansatte hos Dataloy som har bidratt med kunnskap og hjulpet oss videre i riktig retning når det har vært tekniske utfordringer i prosjektet. Vi har også fått innføring i agile utviklingsmetoder gjennom å arbeide med prosjektet hos Dataloy. Vi er takknemlige for at vi har hatt kontor plass hos Dataloy, noe som har bidratt til en tett dialog mellom deltagerne i prosjektet og prosjekteier. Vi ønsker også å takke Dataloy for å ha gitt oss muligheten til å gjennomføre dette prosjektet. Til slutt vil vi takke Peak Shipping som har vært pilotkunde for den applikasjonen vi har laget. Det har vært til stor hjelp for forståelsen av bruken av en slik applikasjon å få snakke med noen som kan tenkes å bruke denne en gang i fremtiden.

II. Innholdsfortegnelse

1. INTRODUKSJON.....	1
1.1 MÅL OG MOTIVASJON.....	1
1.1.1. <i>Problemstillingen</i>	1
1.1.2. <i>Mål</i>	2
1.1.3. <i>Dagens situasjon</i>	3
1.2 KONTEKST	4
1.2.1 <i>Prosjektet</i>	4
1.2.2 <i>Bedriften og prosjektgruppen</i>	4
1.3 BEGRENSNINGER.....	4
1.3.1 <i>Pilot-prosjekt</i>	5
1.3.2 <i>Tjener / Klient</i>	5
1.3.3 <i>Bruk av eksterne leverandører</i>	5
1.4 RESSURSER.....	5
1.4.1 <i>Litterære ressurser</i>	5
1.4.2 <i>Veileder ved HVL</i>	6
1.4.3 <i>Web-baserte kurs</i>	6
1.4.4 <i>Medarbeidere i bedriften</i>	6
1.5 ORGANISERING AV RAPPORTEN	6
2. PROSJEKTBEKRIVELSE	7
2.1 PRAKTISK BAKGRUNN	7
2.1.1 <i>Eier av prosjekt og resultat</i>	7
2.1.2 <i>Tidligere arbeid</i>	7
2.1.3 <i>Kravspesifikasjon</i>	8
2.1.4 <i>Første løsningsutkast</i>	9
2.2 BAKGRUNN	10
2.2.1 <i>React med Redux onlinekurs</i>	10
2.2.2 <i>Eksisterende applikasjoner</i>	10
2.2.3 <i>AIS-signaler</i>	11
3. PROSJEKTDESIGN.....	12
3.1 MULIGE LØSNINGER	12
3.1.1 <i>Alternative kartløsninger</i>	12
3.1.2 <i>Komponenter i applikasjonen</i>	14
3.1.3 <i>Lagring av AIS-data fra skipene</i>	15
3.1.4 <i>Alternative kilder til værdata</i>	15
3.1.5 <i>Diskusjon angående mulige løsninger</i>	17
3.2 SPESIFIKASJON	18
3.3 VALG AV VERKTØY OG PROGRAMMERINGSSPRÅK.....	19
3.3.1 <i>Biblioteker</i>	19



3.3.2	<i>Git</i>	19
3.3.3	<i>Verktøy</i>	19
3.4	PROSJEKTUTVIKLINGSMETODE.....	19
3.4.1	<i>Utviklingsmetode</i>	19
3.4.2	<i>Risikohåndtering</i>	20
3.5	EVALUERING.....	20
3.5.1	<i>Mål for prosjektet</i>	20
3.5.2	<i>Hvordan nå målet</i>	21
3.5.3	<i>Hvordan evaluere om målet er oppnådd</i>	22
4.	DETALJERT DESIGN	24
4.1	STRUKTUR.....	24
4.2	VVAT SOM EN DEL AV VMS.....	24
4.2.1	<i>Integrering av applikasjonen</i>	24
4.2.2	<i>Arkitektur</i>	25
4.3	HOVEDKOMPONENTER I APPLIKASJONEN.....	26
4.3.1	<i>LeafletMap</i>	27
4.3.2	<i>VoyageAnalysisModalBase</i>	28
4.3.3	<i>VoyageAnalysisDrawer</i>	29
4.4	LEAFLET.JS.....	30
4.5	BRUKERGRENSESNIITT.....	32
5.	EVALUERING	33
5.1	EVALUERINGSMETODE.....	33
5.2	EVALUERINGSRESULTAT.....	34
6.	DISKUSJON	36
6.1	GJENNOMFØRING AV PROSJEKTET.....	36
6.2	DRØFTING AV RESULTATER.....	37
6.3	REFLEKSJON.....	38
7.	KONKLUSJONER OG VIDERE ARBEID	40
7.1	KONKLUSJON.....	40
7.2	VIDERE ARBEID.....	40
8.	LITTERATUR/REFERANSER	41
8.1	REFERANSER.....	41
8.2	ORDLISTE.....	43
8.3	AKRONYMER.....	44
9.	VEDLEGG	45
9.1	PROSJEKTPLAN.....	45
9.2	RISIKOANALYSE.....	46
9.3	MØTEREFERATER.....	47
9.3.1	<i>Møtereferat fra evalueringsmøte med pilotkunde</i>	47
9.3.2	<i>Møtereferat fra evalueringsmøte med oppdragsgiver</i>	49

III. Bildeliste

Figur 1: Dataloy Distance Table	3
Figur 2: Utklipp av kartløsningen OpenStreetMap ved bruk av Leaflet	14
Figur 3: Viser arkitekturen på den totale løsningen	25
Figur 4: ECA-sone i Nordsjøen tegnet på kartet som røde linjer. Utklipp fra applikasjonen.	27
Figur 5: Utklipp av modal-komponenten med ekte data.	28
Figur 6: Utklipp av applikasjonen som viser skuffen til høyre.	29
Figur 7: Utklipp som viser hvordan applikasjonen ser ut når den lastes inn	30
Figur 8: Utklipp fra applikasjonen med kartet sentret på Bergen ved zoom-nivå 14.	31
Figur 9: Viser kartet uten filter	32
Figur 10: Viser kartet med filter "In Port" (i havn)	32
Figur 11: Viser kartet med filter "En Route" (i rute)	32
Figur 12: Gantt-diagram	45

1. INTRODUKSJON

1.1 Mål og motivasjon

Øystein Malt (UI arkitekt) fra Dataloy hadde en gjesteforelesning ved Høgskulen på Vestlandet i faget grafiske metoder høsten 2018. Lars Erik kontaktet Øystein i første omgang og deretter ledelsen i Dataloy med tanke på å skrive en bacheloroppgave for Dataloy. Lars Erik har lang og bred erfaring fra shipping og kundegruppen til Dataloy er slike bedrifter som Lars Erik har jobbet for tidligere. Under samtalene rundt prosjektet bestemte Dataloy seg for å tilby Lars Erik deltidsstilling frem til sommeren 2019, samtidig som et passende prosjekt ble laget. Glenn ble forespurt om å delta på prosjektet med bakgrunn i at han har interesser for front-end utvikling.

Dataloy utvikler systemer for operasjon og befraktning av skip. Kundene til Dataloy er i hovedsak rederier eller andre selskaper som driver og/eller operer skip. Dataloy tilbyr løsninger for å planlegge sjøreiser med lasting/lossing, bunkersoperasjoner og annet. Dataloy har også system for gjennomføring av nevnte sjøreiser og operasjoner. Videre har Dataloy en distansetabell som viser den antatt korteste ruten mellom to havner, hvor det er praktisk mulig å seile.

Det finnes i dag en mengde applikasjoner som for eksempel MarineTraffic som gir kunder informasjon som baserer seg på AIS-signaler som sendes ut fra skip. Dataloy har ikke tatt i bruk AIS-data i sine applikasjoner til nå.

Ved å koble sammen AIS-data med distansetabellen som allerede eksisterer hos Dataloy, samt bruke AIS-data fra skipene er det rimelig å anta at man kan gi merverdi til kundene til Dataloy, og at systemene som Dataloy tilbyr sine kunder blir enda mer komplett. Dette vil vi komme tilbake til senere.

1.1.1. Problemstillingen

Dataloy Systems har som fokusområde å utvikle, levere, markedsføre og selge sine avanserte supportsystemer til rederinæringen. Visjonen til selskapet er å være den foretrukne løsningen for integrert befraktning- og operasjons-programvare i hele verden (Dataloy, u.d.).

Dataloy sine systemer består i dag av flere forskjellige moduler som distansetabell, kommisjoner, kalkyleverktøy, osv. For å kunne gi økt verdi til kundene sine ønsker Dataloy å

utvikle en modul som kombinerer AIS-data fra skip med den allerede eksisterende distansetabellen. Den nye modulen skal kunne gi kundene nytteverdi gjennom grafisk fremstilling av utseilt rute sammenlignet med ruter fra distansetabellen. Dataloy ønsker også at den nye modulen skal kunne gi kundene informasjon om skipenes fart i sanntid sammenlignet med den farten skipene har ordre om å seile i. For å kunne gi et riktig bilde av fart i forhold til beordret fart vil det også være nødvendig å hente inn værdata for det spesifikke området da dårlig vær i stor grad påvirker et skips evne til å opprettholde fart.

1.1.2.Mål

Målet med prosjektet er å utvikle en applikasjon med grunnleggende informasjon hentet fra AIS-data til skip, og fremstille dette på en brukervennlig måte slik at kundene til Dataloy kan få nytteverdi av applikasjonen. Det er ønskelig at applikasjonen blir laget skalerbar slik at den kan utvides med flere egenskaper etter at dette prosjektet er gjennomført.

I første omgang er målet at applikasjonen skal hente ned AIS-data fra skip og basert på dette kunne vise posisjonen til skipene på en dertil egnet karttjeneste fra en tredjepart. Ved å lagre posisjonen til et skip med et forhåndsbestemt intervall skal applikasjonen kunne vise hvor skipet har seilt fra forrige havn til nåværende posisjon. Applikasjonen skal vise utseilt rute på kartet og sammenligne denne med ruten fra Dataloys distansetabell. Basert på denne sammenligningen er målet at applikasjonen skal hjelpe de som opererer skipene å avgjøre om skipene følger den mest effektive ruten eller ikke. Videre gir AIS-signalet til skipene informasjon om skipets fart. Ved å overvåke denne data er målet at applikasjonen skal kunne gi en indikasjon til de som opererer skipet at noe kan være galt dersom skipet ikke holder normal fart.

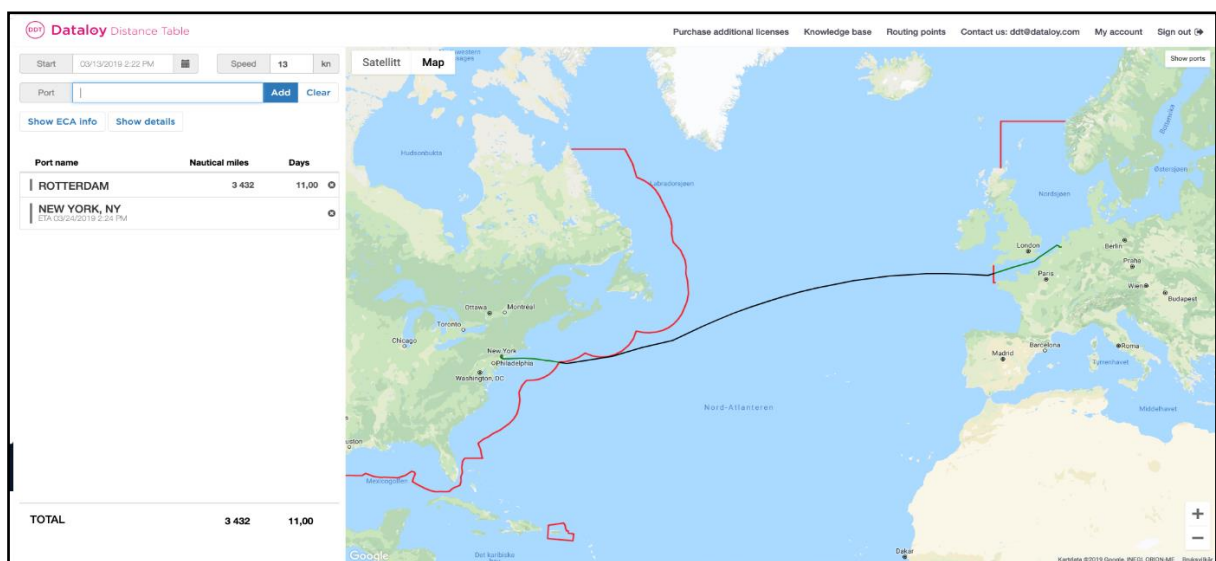
Det er ønskelig at applikasjonen skal kunne sende push-varsel til brukerne ved eventuelle avvik fra rute eller at farten er lavere enn normalt. I videre iterasjoner er det ønskelig å sende push-varsel dersom et fartøy endrer status på AIS fra for eksempel «underveis» til «fortøyd».

Dersom det blir mulighet med tanke på tid og andre begrensninger i bachelorprosjektet, er det ønskelig å implementere værdata fra en egnet værtjeneste for å kunne hjelpe å gi kundene mer informasjon om hvorfor et skip eventuelt ikke seiler med normal fart. Målet er at applikasjonen skal kunne integreres i Dataloys system.

1.1.3. Dagens situasjon

Dataloy har i dag en applikasjon som heter Dataloy Distance Table. Denne applikasjonen gir kundene mulighet til å finne distansen mellom to eller flere havner. Applikasjonen viser havnene og ruten mellom havnene på et kart fra Google Maps som er integrert i applikasjonen. På applikasjonen viser man også grensene for ECA områder som er tredd i kraft. Denne applikasjonen kan hjelpe kunder å beregne tid og kostnad på en sjøreise som planlegges, samt være med å hjelpe til med fremtidig planlegging for å vite når en båt er ledig for neste oppdrag.

Det finnes en rekke applikasjoner som i dag som tilbyr kundene verdi basert på AIS-data fra skip. MarineTraffic er en side som tilbyr dette. Dersom man betaler for tjenestene til MarineTraffic kan man for eksempel filtrere andre skip enn skipene man opererer selv. Gjennom å betale for de ulike tjenestene som MarineTraffic tilbyr kan man få en rekke opplysninger gjennom denne applikasjonen.



Figur 1: Dataloy Distance Table

1.2 Kontekst

1.2.1 Prosjektet

Dataloy har informasjon om at flere kunder bruker MarineTraffic eller tilsvarende tjenester i dag for å overvåke posisjonene til skipene sine. På bakgrunn av den allerede eksisterende distansetabellen Dataloy har og behovet til kundene for å kunne overvåke posisjoner til skip, har Dataloy spurt kunder om det vil gi økt brukerverdi for kundene dersom Dataloy implementerer noe tilsvarende som MarineTraffic tilbyr, men med diverse andre brukergrensesnitt, som for eksempel sammenligning av utseilt rute med den eksisterende distansetabellen. På bakgrunn av dette behovet ble prosjektet etablert.

1.2.2 Bedriften og prosjektgruppen

Prosjektgruppen blir et eget utviklerteam i bedriften i løpet av prosjektperioden. Det vil bli jobbet selvstendig med prosjektet, men medlemmer av de andre teamene hos oppdragsgiver vil være tilgjengelig dersom det oppstår tekniske utfordringer i løpet av prosjektet.

1.3 Begrensninger

Prosjektet i seg selv er begrenset i forhold til tid ettersom det er en innleveringsfrist fra HVL for den endelige prosjektrapporten og det endelige resultatet av prosjektet. Enkelte begrensninger i prosjektet vil derfor kunne relateres direkte til dette faktum. Det betyr at det overordnede prosjektet for Dataloy sin del kan ha færre eller mindre begrensninger enn dette spesifikke prosjektet. Mer spesifikt kan man si at noen av funnene ved brukerundersøkelser vil være naturlig å utelate fra målet i dette prosjektet. Det er likevel viktig å formidle disse funnene videre til Dataloy slik at det kan implementeres dersom applikasjonen skal videreutvikles hos Dataloy etter endt bachelor-prosjekt.

1.3.1 Pilot-prosjekt

Det er avtalt å gjøre et pilot-prosjekt med en av kundene til Dataloy. Kunden har flere skip og ønsker naturligvis å kunne fremstille alle skip i applikasjonen. Det er avtalt at det under prosjektet vil det i utgangspunktet bli begrenset til å vise et par skip i applikasjonen. Dette er gjort for å sikre at fremstilling og funksjonalitet skal optimaliseres før man implementerer flere skip for samme bruker i applikasjonen.

1.3.2 Tjener / Klient

Det er avtalt at Dataloy skal implementere tjeneren som skal brukes i prosjektet, herunder også sette opp database for lagring av data. Prosjektgruppen har fått en klar begrensning gjennom at den skal kun lage klienten til denne applikasjonen. Tjeneren og data som blir lagret vil bli satt opp på et test-miljø av oppdragsgiveren, og oppdragsgiveren skal sørge for drift og vedlikehold av tjeneren under prosjektet.

1.3.3 Bruk av eksterne leverandører

Det er lagt opp til å bruke AISHub for å få tilgang til et API for AIS-data fra skip og det er avtalt med kunde fra pilot-prosjekt at man har tillatelse til å bruke deres AIS-data i applikasjonen. Dataloy stiller en AIS-antenne til disposisjon under prosjektet. Dette er nødvendig for å kunne gjøre nytte av API med AIS-data. Prosjektgruppen er ikke ansvarlig for å finne nye leverandører av eksternt data dersom AISHub eller kunden fra pilotprosjektet skal falle fra og ikke være tilgjengelig under prosjektet.

1.4 Ressurser

1.4.1 Litterære ressurser

Litterære ressurser blir brukt til å innhente grunnleggende forståelse for det å skrive en bachelor-oppgave. Videre vil litteratur bli brukt til å underbygge teori i oppgaven og underbygge de valgene som tas i løpet av prosjektet.

1.4.2 Veileder ved HVL

Veilederen vår ved HVL vil være en viktig ressurs i forhold til skriving av rapporten under prosjektet. Veilederen vil bidra med å henvise oss til litteratur og andre ressurser dersom det er behov for dette. Sist men ikke minst vil veilederen være en viktig ressurs med tanke på å sikre kvaliteten i rapporten ved hjelp av gjennomgang av materiale før innleveringer.

1.4.3 Web-baserte kurs

Web-baserte kurs vil bli brukt for å innhente kunnskap om React og Redux, samt for å få en forståelse av hvordan større applikasjoner er satt sammen. Dette vil bidra til å gi kunnskap til å løse de tekniske utfordringene man møter på i løpet av prosjektet.

1.4.4 Medarbeidere i bedriften

Medarbeiderne i bedriften vil være en særdeles viktig ressurs med tanke på å oppnå ønsket resultat. Medarbeiderne i bedriften har lang erfaring både med arkitektur, UX/UI og koding av klienter. Det vil derfor være naturlig både å søke hjelp når vi støter på utfordringer som vi ikke klarer å løse på egenhånd, men også å be om en gjennomgang av det som blir utført for å sikre kvalitet på kode og arkitektur.

1.5 Organisering av rapporten

Denne forprosjektrapporten inneholder 3 kapitler. Rapporten beskriver forarbeidet som er gjort og hvilke forventninger og krav som stilles til prosjektet som skal gjennomføres.

Kapittel 1 beskriver problemstillingen, oppdragsgiveren, motivasjonen og mål for prosjektet.

Kapittel 2 inneholder prosjektbeskrivelsen. Her går også rapporten nærmere inn på hvorfor dette prosjektet er viktig for oppdragsgiveren.

Kapittel 3 gir en oversikt over prosjekt designet.

2. PROSJEKTBEKRIVELSE

2.1 Praktisk bakgrunn

Kapittel 2.1 beskriver bakgrunnen for dette prosjektet. Kapitlet vil gå nærmere inn på hvem som eier produktet, hvilket arbeid som er utført i forkant av prosjektet. I tillegg vil det bli tatt for seg hvilke metoder som blir brukt under utviklingen av applikasjonen, hvilke verktøy som blir brukt for å løse problemet og oppnå målene, samt at løsningen og funksjonaliteten forklares nærmere.

2.1.1 Eier av prosjekt og resultat

Dataloy Systems eier alle rettigheter til løsningen som kommer frem i denne rapporten.

Dataloy Systems eier også alle rettigheter til bruk eller salg av applikasjonen som blir laget i dette prosjektet, samt rettigheter til å videreutvikle resultatet av dette prosjektet.

2.1.2 Tidligere arbeid

Dataloy Systems har gjennom en årrekke bygget opp en applikasjon som viser den antatt beste ruten mellom to havner. Denne applikasjonen heter Dataloy Distance Table (DDT). DDT skal brukes i applikasjonen som lages i dette prosjektet da applikasjonen skal sammenligne utseilt distanse mot DDT sin distanse mellom to havner. Applikasjonen skal også kunne vise faktisk seilt rute på kartet sammenlignet med ruten som ligger i DDT.

Dataloy Systems har tidligere gjort enkle tester på å hente ut AIS informasjon ved at de installerte en AIS-antenne på kontoret, og laget en brukerkonto på aishub.net. Denne brukeren har ikke vært aktiv på en god stund, så dette er noe som skal gjøres på nytt i prosjektet, men man kan likevel hevde at det er gjort undersøkelser på om dette er enkelt å sette opp eller ikke.

Dataloy Systems har snakket med noen kunder som de bruker i pilot-prosjekter for å undersøke om dette er en applikasjon som kan gi verdi for kundene. I starten var kundene noe kjølig til ideen, men etter å ha fått presentert mulighetene og en mer grundig innføring av ideen så var det positive tilbakemeldinger fra kundene.

2.1.3 Kravspesifikasjon

Før prosjektet ble realisert ble det gitt noen generelle krav fra oppdragsgiveren. Det ble avtalt et møte en kunde som hadde takket ja til å være pilot-kunde i prosjektet, og etter dette møtet ble det laget en mer detaljert beskrivelse av hva prosjektet skulle ha som mål og delmål. Det ble også satt begrensninger for hva som ikke skulle være med i prosjektet, men som kunder kan antas å ønske implementert i fremtiden. Det som er utenfor prosjektet og eventuelt kan implementeres i iterasjoner etter prosjektet vil ikke bli gjennomgått detaljert i denne rapporten da det er ønskelig at slike detaljer ikke deles med offentligheten.

Viktige krav som skal implementeres i applikasjonen i dette prosjektet er å vise posisjonen til skipene i sanntid, basert på AIS-data hentet fra skipenes AIS-sender via AISHub sitt API. Når man trykker på et ikon som representerer en båt skal det vises informasjon om den aktuelle båten. Denne informasjonen skal minimum vise fart, kurs og posisjon som er automatisk implementert gjennom AIS-signaler sendt fra båten. Applikasjonen skal også vise hva som er neste havn, og om skipet er underveis eller er fortøyd/ligger til ankers. Dette er informasjon som gis manuelt fra mannskapet på skipet til AIS-enheten om bord. Denne informasjonen vil alltid være tilgjengelig gjennom AIS-signalet, men applikasjonen er likevel avhengig av at denne informasjonen som er puttet inn manuelt er gjort riktig av mannskapet for at det skal vises riktig informasjon i applikasjonen.

Informasjon om båtens fart skal kunne sjekkes mot en referanse-fart for det aktuelle skipet og applikasjonen skal gi varsel til brukeren av applikasjonen dersom farten avviker i vesentlig grad fra den farten båten skal holde normalt sett.

Seilingsruten til båten skal kunne sammenlignes med seilingsruten som ligger i Dataloy sin distanse tabell. Dataloy sin distanse tabell antas å være en av de mest tidseffektive sjøreisene mellom to havner. Dersom båten avviker i vesentlig grad fra den seilingsruten som ligger i distanse tabellen skal applikasjonen varsle om dette.

Varsel om avvik i fart og/eller seilingsrute skal gis i form av at ikonet som representerer båten på kartet skal endre farge.

Det er ønskelig at man også får opp værdata for båten for den spesifikke posisjonen båten har i det man trykker på ikonet som representerer båten.

2.1.4 Første løsningsutkast

Applikasjonen skal skrives i React. Det er en applikasjon som på sikt skal integreres i Dataloys Voyage Management System (VMS). Det vil bli laget en web-side som på lik linje med alle de andre applikasjonene i Dataloys VMS fyller hele skjermen, bortsett fra en navigasjonsmeny helt øverst i skjermen. Siden som denne applikasjonen viser vil i hovedsak inneholde et kart som skal vise båtene. På kartet vil et ikon representere hver båt. Det er denne hovedsiden som vil kontrollere elementene som vises på siden. Over kartet vil det implementeres tre knapper som vil fungere som filter. Disse tre filtrene vil gi brukeren av applikasjonen mulighet til å vise enten båter som ligger i havn, båter som seiler eller båter som har avvik fra enten rute eller fart. Det er bare båter som seiler som kan ha avvik fra seilingsrute eller fart, slik at båter med avvik er en delmengde av båter som seiler.

Komponenten som er hovedsiden i applikasjonen vil lagre informasjon om skipene og værdata i «state» og vil sende denne informasjonen videre til underkomponenter som «props» til kartet, informasjonsmodal og informasjonsskuffen.

Applikasjonen vil hente AIS-data fra skipene som er konfigurert i applikasjonen. Disse skipene vil vises på kartet. Når man trykker på et skip vil man kunne få opp mer informasjon om det aktuelle skipet i en informasjonsmodal. Modalen skal gi informasjon om båtens nåværende status, herunder også kurs og fart, samt informasjon om reisen båten er på nå. Det skal også informeres om værdata i denne modalen. Samtidig som modalen vises skal også reiseruten skipet har hatt fra forrige havn vises i kartet. Denne ruten skal kunne sammenlignes visuelt med seilingsruten fra Dataloys distansetabell. Det skal være en knapp på modalen som kan brukes for å åpne informasjonsskuffen. Dette er en skuff som inneholder mer informasjon enn modalen. I skuffen kan man også kvittere ut eventuelle avvik slik at ikonet viser status uten avvik i applikasjonen.

Det er ikke blitt gitt noen strenge retningslinjer fra Dataloy for hvordan løsningen skal være. Under prosjektet vil det være stor frihet. Det er rimelig å anta at denne friheten er oppnådd med bakgrunn i at Lars Erik er deltidsansatt i Dataloy, samt at han har lang erfaring fra bransjen som er målgruppen for applikasjonen. Det legges også vekt på at det blir avholdt møter direkte med målgruppen som kan tenkes å ta i bruk en slik applikasjon og at dette også gjør at Dataloy ikke legger alt for sterke føringer for hvordan løsningen skal se ut eller hvordan man skal nå målene.

Dataloy ønsker imidlertid at applikasjonen skal være mulig å integrere i en totaløsning som en av flere applikasjoner i en dashboard-løsning som oppdragsgiver er i ferd med å utvikle. Dette gjør at layout på header, information drawer og kartvisning er bestemt av Dataloy.

Dataloy ønsker at applikasjonen skal skrives i React/JavaScript og SCSS slik at det er enkelt for dem å videreutvikle og vedlikeholde applikasjonen. Dette samsvarer med språkene som selskapet bruker i dag. Det stilles krav til god programmeringsskikk og at koden dokumenteres med JavaDoc.

Kort oppsummert kan man si at løsningen baserer seg på brukerhistorier som blir laget basert på kundemøter, og at brukergrensesnittet baserer seg på kundemøter samt selskapets generiske design.

2.2 Bakgrunn

2.2.1 React med Redux onlinekurs

Før utvikling av prosjektet kunne begynne ble det gjennomført et onlinekurs i React. Dette kurset ble anbefalt av Dataloy, og er et kurs beregnet for nybegynnere. Ved å gjennomføre hele kurset tilegner man seg nok kunnskap til å gjennomføre dette prosjektet, ettersom det er en grundig innføring i React og Redux. Det var en totaltid på 47,5 timer med video, der man programmerer hele veien. På denne måten tilegner man seg kunnskap til å bygge web-applikasjoner fra grunnen. Gjennomføringen av kurset ble prioritert i første del av semesteret, ettersom det anses som en nødvendig faktor for at prosjektet skulle være gjennomførbart.

2.2.2 Eksisterende applikasjoner

Det finnes flere ulike applikasjoner for real-time AIS-data på markedet. Noen av disse applikasjonene er VesselFinder og MarineTraffic. Dette er applikasjoner som viser de fleste båter i verden på et kart, basert på båtenes AIS-signaler. Et av de største problemene med disse applikasjonene er at de viser for mange båter, som fører til at applikasjonene blir uoversiktlige. Peak Shipping informerte under et møte at de ønsket en applikasjon som viser sine egne båter, og eventuelt konkurrentenes båter. Dermed ser man et behov for applikasjoner som tar hensyn til dette, og som viser data på en mer oversiktlig måte.

Hente ut riktig informasjon og kunne vise frem informasjonen på hovedskjermen er et annet punkt som mangler i flere applikasjoner. Eksempler på data som kan vises på hovedskjermen kan være fart, navn på båtene og status. Dette er noe som ikke er mulig i de store applikasjonene nevnt ovenfor ettersom det er for mange båter der, og dataen ville overlappet slik at man ikke ser hva som står på skjermen.

2.2.3 AIS-signaler

AIS-signaler får man fra AISHub dersom man deler AIS-signaler med dem. Dette gjør at hvem som helst har tilgang på AIS-signaler fra alle båter så lenge de har en måte å dele AIS-signaler på. AIS er et automatisk sporingssystem som sender ut navigasjonssignaler fra båter. I desember 2004 ble det satt et krav om at alle passasjerfartøy, samt alle kommersielle fartøy over 299 bruttotonn må ha klasse A AIS-transpondere om bord for å sende og motta AIS-signaler (MarineTraffic, u.d.). Mindre fartøy kan ha klasse B AIS-transpondere. Dette er for å unngå kollisjoner blant fartøy, og er en del av SOLAS (Safety of Life at Sea) avtalen.

AIS-signaler inneholder informasjon som kan deles inn i to typer; statisk og dynamisk. Statisk informasjon er informasjon som blir sett ved installasjon og forandres ikke med mindre fartøyet undergår en forandring til en annen type fartøy.

Informasjon som inngår i den statiske delen av AIS-signaler er: *Maritime Mobile Service Identity, call sign, name, International Maritime Organization number, dimensions, location of the electronic position fixing system antenna, type of positioning system and type of ship* (Lines, 2015).

Informasjon som er dynamisk kan videre deles inn i to typer, informasjon som oppdateres automatisk og informasjon som må manuell oppdateres.

Automatisk oppdatert informasjon inneholder: *ship position, position time stamp in UTC-format, course over ground, speed over ground, heading and rate of turn.*

Manuell oppdatert informasjon inneholder: *navigational status, ship's draught, hazardous cargo, destination, ETA and route plan.*

3. PROSJEKTDESIGN

3.1 Mulige løsninger

3.1.1 Alternative kartløsninger

Det første spørsmålet som blir diskutert i sammenheng med kartet som skulle brukes var om det var behov for sjøkart eller om det ville være tilstrekkelig med et ordinært kart. Formålet med applikasjonen er å vise hvor skipene befinner seg, samt å vise hvor de har seilt fra forrige havn. Man kan tenke seg tilfeller der det ville være en fordel å vise skipene på et sjøkart, dersom man skulle forklare en seilingsrute som ikke er den antatt korteste ruten mellom to punkter. Det kan tenkes at den korteste ruten mellom to punkter på sjøen går over områder hvor det ikke er farbart farvann. Hovedformålet i denne applikasjonen er å vise hvor skipene befinner seg og hvilke ruter skipene har valgt. Sammenligningen av historiske ruter skal i denne applikasjonen sammenlignes med seilingsruten som ligger i Dataloy sin distansetabell. Det er også et viktig poeng at applikasjonen ikke skal brukes som et navigasjonshjelpemiddel. Med bakgrunn i disse vurderingene, samt at et sjøkart vil bety en merkostnad som man antar ikke vil gi nok verdi for kundene, falt valget på at man skal bruke et ordinært kart for å vise posisjoner og historiske seilingsruter i applikasjonen. Det skal likevel nevnes at det ikke er undersøkt med kunder av Dataloy om det er behov for å ha denne applikasjonen på et sjøkart, slik at en slik undersøkelse hos kundene kan anbefales Dataloy å gjøre i fremtiden dersom applikasjonen skal videreutvikles.

Når man skal ha et kart å vise frem båtene på var det helt unaturlig å tenke på å lage eget kart. Det er flere forskjellige leverandører av karttjenester som eksisterer og som dekker behovet for denne applikasjonen. Ifølge AppleWorld har Google Maps en markedsandel på 67% når det gjelder kart i navigasjonstjenester i dag (Sellers, 2018). Det er derfor rimelig å anta at Google Maps er den mest kjente karttjenesten som leveres i dag. Dataloy bruker Google Maps for å fremstille visuelt rutene som ligger i distansetabellen som selskapet har i dag. Kostnaden ved Google Maps har økt ganske betydelig de senere årene, uten at man i prosjektet har undersøkt eller fått tilgang til denne informasjonen i detalj fra Dataloy, av forretningsmessige årsaker. Med bakgrunn i dette ønsket Dataloy at man i dette prosjektet skulle prøve andre leverandører enn Google Maps som leverandør av karttjenester. Med

bakgrunn i at Google Maps var en kandidat man ikke skulle ta hensyn til i dette prosjektet har man vurdert et par andre aktører i markedet.

OpenLayers med OpenStreetMap er et av alternativene som er vurdert. OpenLayers er basen som man kan bruke for å implementere OpenStreetMap (OpenStreetMap, u.d.). Dette er en kartapplikasjon hvor man kan legge inn forskjellige lag i kartet man skal vise frem (OpenLayers, u.d.). Denne tjenesten er helt gratis. OpenStreetMap operer som et lag i OpenLayers, som også er et kart det er gratis å bruke. OpenStreetMap viser landegrenser og større byer, samt økonomiske grenser i havet. Det er et kart som ikke viser noen form for navigasjonshjelpemidler eller annet som kan forbindes med sjøfart, annet enn at man kan se hva som er den antatt korteste veien mellom to punkter på kartet. Ettersom at applikasjonen ikke skal brukes til planlegging av sjøreiser er det rimelig å anta at det ikke vil være noen stor ulempe at det ikke vises navigasjonssymboler på kartet. En svakhet ved kartet er at havnebyer ikke er markert i kartet. Det betyr at byer som er ansett som mindre byer i sivil sammenheng, men som kan være store havnebyer, vil ikke være avmerket på dette kartet. Det er også et kart hvor man kan rendre linjer og ikoner på kartet. Ulike linjer på kartet kan brukes for å vise hvor ulike soner går, eller å vise hvor et skip har seilt. Det kan også brukes til å fremstille den ruten som ligger i Dataloy sin database fra distansetabellen. Ikoner kan brukes til å illustrere skip på kartet.

Mapbox er en karttjeneste som brukes av både Facebook og Snapchat (Bush, 2018). Det er et kart som kan brukes til å søke etter lokasjoner, til navigering på land og man kan lage egne features på kartet. Dette kartet har flere muligheter enn OpenLayers i utgangspunktet, men disse mulighetene er først og fremst landbaserte. Det er et kart som egner seg godt til å bruke for navigering på land. Disse ekstra features som er tilgjengelig på denne karttjenesten er imidlertid spesifikke for annen bruk enn det som er tenkt i applikasjonen som skal lages i dette prosjektet. Karttjenesten er gratis opp til 50.000 kall på tjeneren og prismessig vil dette være rimeligere enn tjenester som for eksempel Google Maps.



Figur 2: Utklipp av kartløsningen OpenStreetMap ved bruk av Leaflet

3.1.2 Komponenter i applikasjonen

Ved valg av komponenter som brukes i en applikasjon i React må man først vurdere om man skal lage komponenter selv, eller om man skal bruke noen av de bibliotekene som finnes med komponenter. Dette var et viktig veivalg helt i starten i prosjektet og det var flere faktorer som spilte inn før man landet på et valg. Ved å lage egne komponenter har man mindre avhengigheter til andre kilder, som man i praksis ikke kan kontrollere. Dersom man lager egne komponenter har man også frihet i større grad til å gjøre ting akkurat slik man ønsker, ettersom man noen ganger kan oppleve begrensninger med komponenter som er levert av andre aktører. Det finnes flere gratis biblioteker slik at det er ingen kostnader forbundet med å velge komponenter som blir levert av eksterne leverandører. Det vil være en kostnad ved å lage egne komponenter ettersom dette vil ta tid. Normalt sett ville det også være en kostnad i form av lønn til ansatte for å utvikle slike komponenter, men det faller bort i dette prosjektet da ingen som jobber i prosjektet er lønnet. Med tanke på at tid er en knapp ressurs i dette prosjektet er det valgt å legge bort dette alternativet. Da stod man tilbake med å velge hvilke komponenter man skulle bruke. Det valget falt naturlig på Material-UI, siden det er disse komponentene som er valgt hos oppdragsgiver Dataloy for fremtidige applikasjoner de skal lage, samt at de er i gang med å bytte ut gamle komponenter med komponenter fra Material-UI.

3.1.3 Lagring av AIS-data fra skipene

Når det gjelder lagring av AIS-data fra skipene kan man se for seg to hovedmåter å gjøre dette på, og hver av disse hovedmåtene vil ha forskjellige alternativer under seg igjen. Først vil rapporten peke på de to forskjellige hovedalternativene, deretter vil rapporten gå mer i dybden og se på underalternativer på den måten som er aktuell i dette prosjektet. Man kunne sett for seg å bruke eksterne leverandører av historiske AIS-data fra skipene, og på den måten kan man også si at et alternativ ville være lagring av data hos en ekstern leverandør. Det er ikke avdekket noen behov som ikke blir dekket ved å bruke en ekstern leverandør av lagring av data, men det kan tenkes at det i fremtiden kan være begrensninger som man ikke har kontroll over. Kostnaden ved å lagre data i en egen database er relativt lav på bakgrunn av at selskapet allerede har serverløsning og kan lagre data i en database i eksisterende løsning. Det er derfor valgt at man skal lagre data i eget system. Når man skal lagre data i eget system kan man også se for seg forskjellige alternativer.

Et alternativ kan være bruk av fysisk maskinvare som man installerer på dertil egnet sted. I pilotprosjektet skal AIS-data lagres hos selskapet selv, slik at den løsning som allerede eksisterer for test-data er den mest naturlige å velge. Ved implementasjon hos kunder når applikasjonen ved en senere anledning skal lanseres vil kunden bruke den løsning som kunden allerede har for å lagre sine AIS-data. Noen kunder hos Dataloy har egne fysiske servere, mens andre kunder har egne skyløsninger. Andre kunder igjen betaler Dataloy for at selskapet skal drive servere, og da blir data lagret i AWS-servere. Med bakgrunn i at kunder har forskjellige løsninger for hvordan data blir lagret med tanke på den fysiske delen, ser man ikke behov for å gå i videre i dybden på disse alternativene.

3.1.4 Alternative kilder til værdata

Kunder som skal bruke applikasjonen har ytret ønske om å kunne se værdata i informasjonsmodalen og/eller i informasjonsskuffen. Været er en viktig faktor som påvirker farten til et skip.

Det er flere faktorer ved været som har betydning for påvirkningen på skip som seiler på havet. Applikasjonen som skal lages i dette prosjektet skal ikke ta hensyn til sikkerhet for skip og last, men applikasjonen skal kunne gi skipets reder eller operatør en mulighet til å overvåke om skipet for eksempel seiler til den oppgitte farten i et certeparti. For å kunne avgjøre dette

må man vite hvilken fart som er oppgitt i certepartiet, men også ved under hvilke værforhold denne farten er garantert fra eieren av skipet. Skip som seiler på havet blir utsatt for blant annet vind og bølger. Ved vindhastigheter under 20 knop vil skipet oppleve redusert hastighet ved vind mot fartsretningen, mens det vil oppleve øket hastighet med vind i samme retning som fartsretningen (Cai & Wen, 2014). Ved vindhastigheter over 20 knop vil skipets fart bli redusert i begge vindretninger, blant annet på grunn av bølger.

Bølgehøyden har stor påvirkning på skipets fart. Retningen på sjøen har samme effekt som for vinden som er nevnt i avsnittet over. Effekten fra bølgehøyde og retning er større enn den fra vind.

En annen ting som påvirker et skip som seiler på sjøen er strømmen i havet. Denne strømmen er lettere å forutse. I denne applikasjonen vil ikke strøm bli tatt hensyn til da den kan antas å være forutsigbar og lettere å identifisere for redere eller operatører dersom det skal avgjøres om den har hatt påvirkning på skipets fart.

Dersom denne applikasjonen skulle bli brukt til planlegging av sjøreiser så ville værdata spilt en mer vesentlig rolle. Det kan også da argumenteres for at det ville vært naturlig å ha med informasjon om strøm i havet, da dette kunne påvirke valg av rute. Denne applikasjonen er ikke ment som et planleggingsverktøy, og det er derfor ikke like naturlig å ta med data om strøm i havet. Det er mulig å se for seg scenarier der det vil være vesentlig informasjon for reder eller operatør å vite hvilket vær et skip har opplevd på sjøreisen, ved for eksempel lavere fart enn antatt. Man kan også se for seg scenarier der det ville være en fordel for bruker av applikasjonen å kunne sammenligne værdata fra faktisk seilt rute, med den ruten som den faktiske ruten sammenlignes med. Dette vil spesifikt være dersom man oppdager at et skip har seilt en lenger rute enn den korteste ruten mellom to punkter, hvor kapteinen på det aktuelle skipet informerer at rutevalget er gjort med bakgrunn i vær-situasjonen på det aktuelle tidspunktet. I denne iterasjonen av applikasjonen vil dette ikke bli inkludert ettersom det vil være en krevende sak å gjennomføre, samt at lagret data vil øke i omfang. Dersom det skulle bli aktuelt å se på vær tilbake i tid for et spesifikt område kan man dessuten kjøpe tilgang til historisk værdata fra diverse tjenester.

Når man har etablert hvilke værdata som trengs i applikasjonen gjenstår det å velge en leverandør av værdata. Det kan nevnes PassageWeather, Ocean Weather Services, Ocean Weather Services og Storm Glass. Uten å gå i dybden på de forskjellige applikasjonene er det

ikke lett å skille dem fra hverandre i utgangspunktet. Dette prosjektet har ikke som hensikt å kartlegge forskjellene mellom de forskjellige vær-tjenestene, men å finne en som gir de data man trenger, og som samler data fra nok forskjellige meteorologiske tjenester til at man kan anta at det er pålitelig data man får levert. Det er også slik at alle vær-tjenestene har vær-data fra flere forskjellige meteorologiske tjenester. Med den bakgrunn at man skal ha pålitelig vær-data i applikasjonen som skal lages i dette prosjektet er det ingen vesentlig forskjell på de ulike vær-tjenestene som er tilgjengelig.

3.1.5 Diskusjon angående mulige løsninger

På bakgrunn av argumentene som kommer frem i de foregående kapitlene er det gjort valg angående de ulike hoveddelene i applikasjonen.

Ved valg av karttjeneste er det lagt stor vekt på at det skal være så rimelig som mulig. Man antar at det er et viktig poeng for Dataloy å holde kostnadene så lave som mulig ettersom at de ønsker å se på alternativer til Google Maps med tanke på prisen man betaler for denne tjenesten i dag gjennom distansetabellen. Begge alternativene som er vurdert er mindre kostbare enn Google Maps, hvor det ene alternativet er gratis, og det andre er gratis opp til 50.000 kall på tjeneren per måned. For prosjektet sin del så ville man trolig ikke kommet over 50.000 kall på en måned da det bare er en kunde som skal teste applikasjonen i første omgang. Det er valgt å benytte OpenLayers/OpenStreetMap ettersom det er aktuelt for Dataloy å videreutvikle denne applikasjonen etter endt prosjekt. Bakgrunnen for dette er at det med dagens ordning ikke vil bli noen kostnad selv med mange kunder som eventuelt bruker applikasjonen ved en senere anledning. Det er vurdert at de ekstra tjenestene man kan få ved å bruke tjenesten som koster penger ikke er vesentlig for den bruk man antar at kartet vil ha i denne applikasjonen og disse tilleggstjenestene er derfor lite vektlagt i beslutningen ved valg av kart.

Ved valg av komponenter har prosjekteier, Dataloy, gitt prosjektgruppen frihet til å velge selv, men man har også ytret ønske om at det skal være enklest mulig å videreutvikle applikasjonen som en del av plattformen som allerede tilbys kundene i Dataloy i dag. Prosjektet har vurdert alternativene og kommet frem til at Material-UI har komponenter som dekker behovet som applikasjonen har for komponenter. På bakgrunn av at Material-

UI dekker behovet og at Dataloy har besluttet å bruke Material-UI på sine komponenter i fremtiden er det valgt Material-UI i denne applikasjonen.

Vedrørende lagring av data fra AIS så har man lagt vekt på å ha kontroll på data i egne systemer. I første omgang er det lagt opp til å lagre AIS-data annethvert minutt for sjøreisene som pilot-kunden gjør med sine skip. Pilot-kunden opererer i Nord- og Østersjøen slik at dette er kortere sjøreiser. Man kan ved lengre sjøreiser, som for eksempel ved kryssing av Stillehavet anta at det ikke er behov for å lagre data like ofte da. Ved å ha en egen tjener og database hvor data lagres vil man ha kontroll over dette i fremtiden, men dersom man baserte seg på å hente data lagret hos en ekstern leverandør vil man kanskje få store mengder data for en lang sjøreise, og det kan tenkes at det kan bli utfordringer med effektiviteten på klientsiden når dette skal rendres dersom det er ekstremt mange punkter det skal trekkes linjer mellom på en lang sjøreise.

Ved valg av kilde til værdata er det vesentlig at man får den værdata man trenger i applikasjonen og at denne er basert på nok målinger til at det er data som gir et riktig bilde av situasjonen. Det er flere applikasjoner som tilbyr tjenester som baserer seg på målinger fra flere meteorologiske stasjoner. Det var derfor ingen klar kandidat som skilte seg ut fra de andre tjenestene når det gjaldt værdata, og man har valgt en tjeneste som man ser leverer det man trenger. Her kunne man trolig fått like gode resultater om man hadde valgt en av de andre tjenestene.

3.2 Spesifikasjon

Applikasjonen skal skrives i React med Redux. Applikasjonen skal bruke leaflet som karttjeneste for å kunne visuelt fremstille skipene på et kart. AIS-hub sitt API blit brukt for å hente informasjon om skipenes posisjon. Ikonene som skal fremstille båter er laget som objekter som sendes inn i kartet med posisjon og dermed tegnes på riktig sted i kartet. For å kunne gi informasjon om været der hvor båtene befinner seg er det bestemt at applikasjonen skal bruke StormGlass sitt API for dette. De forskjellige komponentene som for eksempel knapper for filter, eller pop-up og skuff på siden av skjermen er komponenter som er hentet fra Material-UI. Annet design i komponentene som inneholder tekst som informasjon bruker også Material-UI komponenter, dette inkluderer blant annet tabeller og tekstfelder.

3.3 Valg av verktøy og programmeringsspråk

De ulike temaene som blir forklart nedenfor er tatt basert på at oppdragsgiver har et sterkt ønske om å videreutvikle applikasjonen, og dermed er det hensiktsmessig å benytte de samme verktøyene og bibliotekene som de bruker.

3.3.1 Biblioteker

I dette prosjektet er det valgt å ta i bruk React, som er et Javascript bibliotek for å utvikle applikasjonen (*React.*, 2019). I tillegg er det nødvendig å benytte Redux for å håndtere data og tilstander i applikasjonen (*Redux*, u.d.). Siden Dataloy skriver i TypeScript, blir dette også brukt i prosjektet.

3.3.2 Git

Dataloy benytter Bitbucket for å håndtere kildekoden deres. Dette er et versjonskontrollsystem i likhet med blant annet Github, som er kjent fra studiene. Det er nødvendig å bruke Bitbucket for å kunne gjennomføre prosjektet. Bakgrunnen for dette er at Dataloys systemer er essensielle for å utvikle applikasjonen.

3.3.3 Verktøy

Microsoft Visual Studio Code er et IDE som støtter TypeScript. Det er dette IDE-et som skal brukes i prosjektet. Det er også stilt krav om at det skal benyttes formateringsverktøy, slik at koden er formatert på samme vis som utviklerene ved Dataloy gjør. Dette hindrer unødvendige «merge conflicts» i Bitbucket. I prosjektet blir det benyttet Prettier, et formateringsverktøy som støtter TypeScript (*Prettier.*, u.d.).

3.4 Prosjektutviklingsmetode

3.4.1 Utviklingsmetode

Som diskutert tidligere ble det bestemt at Scrum ville være prosjektets utviklingsmetodikk. Gjennom prosjektet vil det gjennomføres sprint planlegging ved starten av hvert intervall. Denne planleggingen vil ta for seg mål for sprinten, og fastsette hva fokuset skal være. I løpet av sprinten vil det holdes daglige møter på 5-10 minutter. Gjennom møtene blir det

oppsummert hva som ble gjort dagen før, og hva som skal gjøres gjeldende dag. På slutten av sprinten vil det bli avholdt et *Sprint Review* som oppsummerer hele sprinten. Under dette møtet vil bedriften få en prosjektoppdatering. Dette er for å få tilbakemeldinger på hva som mangler, og hva som må prioriteres.

Det vil gjennomføres demonstrasjoner for kunden, men her passer det seg ikke å ha demonstrasjoner ved hver endt sprint. Derfor vil kunden bli oppdatert når det passer for kunden, og dette blir lagt utenom Scrum sprinten. På slutten av hver sprint vil det være hensiktsmessig å avtale møte med intern veileder, for å få tilbakemeldinger fra skolen.

3.4.2 Risikohåndtering

Det er laget et risikoanalyseskjema for å se på sannsynlighet, konsekvens og tiltak for ulike risikofaktorer. Det er i tillegg oppgitt hvilken interessent en eventuell risiko vil gå ut over. Dette skjemaet bruker en skala på 1 – 5, hvor 5 er høyeste grad. Tiltakene vil kort si noe om hvordan man kan forhindre at en risiko oppstår. En risikofaktor finner man ved å multiplisere sannsynligheten med konsekvensen, der høyeste verdi er 25. En risikoverdi under 10 vil være lite problematisk, mens en risikoverdi over 20 vil kunne føre til store problemer. Det er derfor viktig at det blir ført tiltak og tatt tak i risikoen tidlig. Skjemaet ligger under kapittel «[9.1 Risikoanalyse](#)».

3.5 Evaluering

3.5.1 Mål for prosjektet

Hovedmålet for prosjektet er å utvikle en applikasjon som har noe funksjonalitet, som det skal snakkes mer om etterhvert. Samtidig er det et viktig mål at applikasjonen kan videreutvikles etter at prosjektet er fullført. Applikasjonen som utvikles i selve prosjektet skal være brukervennlig og ha noen viktige funksjoner implementert. Først og fremst skal applikasjonen kunne vise skip med posisjon i nåtid i et kart. Dette er det mest elementære med funksjonen. Videre skal det være et dynamisk kart hvor man kan endre posisjon og zoom-nivå i kartet. Skipenes posisjon i kartet skal basere seg på data fra AIS-senderen som skipet har. Applikasjonen skal vise informasjon om skipene, herunder informasjon om kurs og fart, havner på reisen og været på posisjonen til det aktuelle skipet. Applikasjonen skal vise hvor

skipet har seilt fra forrige havn og frem til nåværende posisjon. Dette skal vises visuelt på kartet. Applikasjonen skal også ha mulighet for å sammenligne den virkelig seilte ruten med ruten som ligger i Dataloy sin distansetabell. Dersom det er avvik mellom ruten fra distansetabellen og faktisk seilt rute skal applikasjonen gi et visuelt varsel om dette. Det skal også gis et visuelt varsel dersom skipet ikke seiler med den farten det er antatt å seile med. Dette visuelle varselet skal komme i form av at ikonet som illustrerer skipet i kartet forandrer farge fra grønn til rød. Applikasjonen skal også skille mellom skip som seiler og skip som ligger i ro i havn, enten til ankers eller til kai. Skip som er fortøyd i havn skal illustreres med en sirkel, mens skip som seiler skal illustreres med et ikon som minner om et skip i 2D sett ovenfra. I applikasjonen skal det være mulig å filtrere på skip ut fra tre parametere. Man skal kunne filtrere på skip som enten er fortøyd i havn, skip som seiler eller skip som seiler men har en deviasjon. Skip som seiler og har deviasjon er en delmengde av skip som seiler, slik at filteret som viser skip som seiler vil også vise skip som seiler men har deviasjon.

3.5.2 Hvordan nå målet

Det er to interessentgrupper for denne applikasjonen. Den ene er prosjekteier og den andre er kundene som eventuelt vil kjøpe produktet fra prosjekteier.

For å nå målet om brukervennlighet og funksjoner som brukere ønsker er det nødvendig å snakke med de som skal bruke applikasjonen. Det ble gjennomført møte med en pilotkunde i forkant av prosjektet for å kartlegge om dette var en applikasjon det ville være behov for i den aktuelle bedriften. Etter at dette behovet var kartlagt ble det laget en prototype som ble demonstrert for kunden, hvor kunden fikk mulighet til å kommentere prototypen og komme med innspill til denne. Disse innspillene har vært viktige for å vite hva man skulle prioritere å implementere og har vært vektlagt i prioriteringen av oppgavene som har blitt gjort under prosjektet.

Den andre interessenten, Dataloy, har vi hatt daglig kontakt med etter at prosjektet startet. Deltagerne i prosjektet har sittet på kontoret til Dataloy under hele prosjektet. Dataloy er en relativt liten bedrift med mellom 15 og 20 ansatte, og det er en veldig flat struktur i bedriften i forhold til organisasjon. Det fører til at det er enkelt for prosjektet å få raske svar og tilbakemeldinger underveis. Dette har også eliminert behovet for å ha formelle møter der man har sett på progresjon og diskutere veien videre mot målet. Dette har blitt gjort der det

har vært nødvendig, og prototype og applikasjonen selv har blitt demonstrert for prosjekteier på ukentlig basis underveis i prosjektet etterhvert som det har blitt utviklet.

Det vil være viktig å sikre at AIS fungerer slik det er forventet tidlig i prosjektet for å unngå at det oppstår problemer senere. Dersom AISHub viser seg å ikke oppfylle kravene må det vurderes å finne en annen leverandør av AIS.

For å nå målet om en applikasjon som kan videreutvikles av bedriften er det nødvendig at kravene angående bruk av språk, verktøy og utviklingsmetodikk følges strengt. Bedriften oppretter et brett i Jira for å planlegge sprintene som gjennomføres i prosjektet. I tillegg blir det opprettet et repository på Bitbucket det vil være mulig å laste opp koden til. Dette gjør det enkelt å sikre at det blir arbeidet på nyeste versjonen av koden hele tiden. Samtidig gjør det at koden ligger tilgjengelig for bedriften når de senere skal hente den fram for å videreutvikle applikasjonen. Hardkoding i applikasjonen er noe som må unngås til enhver tid, ettersom dette gjør videreutvikling vanskeligere. Det er hensiktsmessig at dette blir fulgt for å oppnå målet om at bedriften skal kunne videreutvikle applikasjonen.

3.5.3 Hvordan evaluere om målet er oppnådd

Det vil bli gjennomført en demonstrasjon av applikasjonen for en pilotkunde og for prosjekteier. I forbindelse med denne demonstrasjonen vil prosjektdeltagerne stille spørsmål til kunde og prosjekteier for å verifisere enkelte punkter i målbeskrivelsen i kapittel 3.5.1. Et mål som kan være vanskelig å verifisere med kun en demonstrasjon og et slikt møte er nytteverdien for kunden. For å virkelig verifisere om applikasjonen har nytteverdi for kunden burde den trolig vært testet over lenger tid. I forhold til dette punktet vil spørsmålet være om kunden tror at denne applikasjonen har nytteverdi. Med tanke på at applikasjonen er tenkt å erstatte noe som kunden allerede bruker er det ikke urimelig å anta at kunden allerede gjennom en slik demonstrasjon vil kunne gi et godt og entydig svar på om applikasjonen har nytteverdi eller ikke.

For å verifisere om applikasjonen har et bra brukergrensesnitt og om den har de funksjonene som er satt opp som mål vil det bli spurt spesifikt til både kunder og prosjekteier om disse punktene. Svarene her vil gi en klar indikasjon på om målet er oppnådd eller ikke. Med bakgrunn i at det kun er en kunde som blir spurt og applikasjonen er ment for flere kunder er det ikke tilstrekkelig med data fra kun en kunde for å kunne fastslå med sikkerhet at målene

er nådd, men for dette prosjektets del er det ansett som tilstrekkelig for å kunne trekke en konklusjon. Prosjekteier mener også det er tilstrekkelig å spørre denne ene kunden for å trekke konklusjoner på det nåværende tidspunkt.

For å vurdere om prosjektet har oppnådd målet om en applikasjon som kan videreutvikles vil det bli stilt spørsmål til begge interessentene rundt dette. Spørsmålene vil naturlig nok ha litt ulik utforming rundt dette. For kunden sin del vil det bli spurt om applikasjonen er noe de kan se for seg å bruke dersom det blir utviklet videre på applikasjonen. Dette vil være viktig for prosjekteier i forhold til om det er behov for å videreutvikle applikasjonen eller om applikasjonen skal være i nåværende tilstand eller eventuelt bare skrinlegges. I forhold til prosjekteier er det vesentlig å spørre om applikasjonen er noe de ser på som kan videreutvikles og om de ser på dette som noe de ønsker å bruke ressurser på å videreutvikle.

4. DETALJERT DESIGN

Dette kapittelet vil begynne med å kort beskrive planleggingen som ble gjort før prosjektets oppstart. Deretter vil det bli beskrevet hvordan mappestrukturen i prosjektet er, og kort forklare hvordan applikasjonen er tenkt integrert i Dataloys VMS. Til slutt vil det forklares hvordan arkitekturen og brukergrensesnittet for applikasjonen er designet.

4.1 Struktur

Ettersom applikasjonen er skrevet i React vil det bli tatt i bruk gjenbrukbare komponenter. På denne måten blir det lettere å gjøre endringer senere. I tillegg kan man bruke komponenten flere plasser, uten å måtte skrive ny kode for den. Det er satt stort fokus på å holde koden gjenbrukbar, og dermed er alt som kan plasseres i en egen klasse blitt plassert i en egen klasse.

Funksjoner som oppretter kartet eller objekter på kartet er lagt i egne klasser i en samlet mappe. Dette er også gjort for komponentene, der de er plassert i mapper med relevante navn. Dette gjør det lett å finne frem i koden, og gir en god struktur på de ulike filene.

4.2 VVAT som en del av VMS

4.2.1 Integrering av applikasjonen

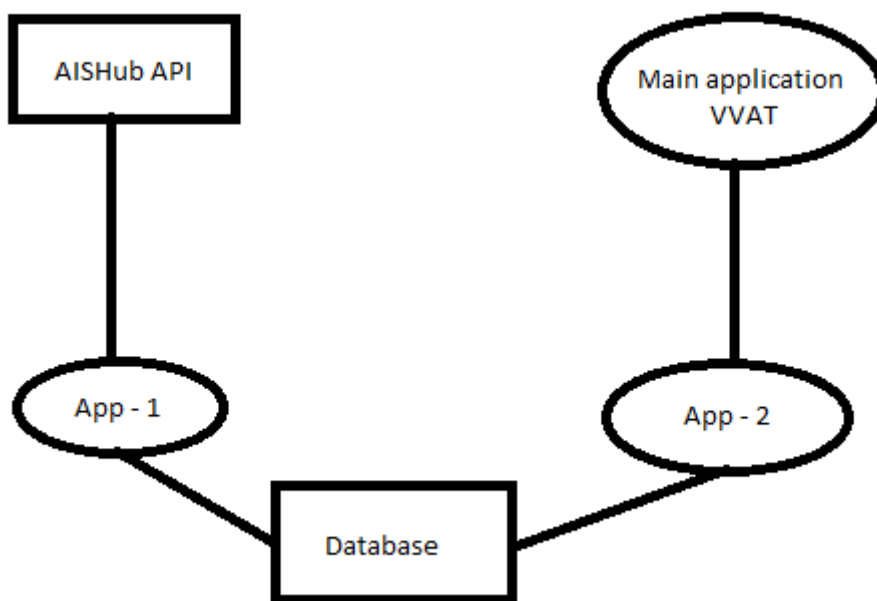
For å lett kunne teste applikasjonen og demonstrere den for eventuelle kunder var det nødvendig å linke til den i VMS. Det er kun mulig å komme til applikasjonen ved å skrive direkte i adressefeltet i nettleseren, ettersom det ikke er lagt inn en lenke til denne i selve VMS. Dette er gjort med tanke på at applikasjonen ikke har behov for å ligge tilgjengelig i VMS, siden applikasjonen er langt fra produksjonsklar.

På et senere tidspunkt vil det være hensiktsmessig å linke til applikasjonen fra en modul i VMS, uten å måtte skrive inn adressen manuelt. Dette er nok noe som kommer til å bli implementert etter prosjektet er ferdig, og dermed ikke noe som skal tas hensyn til i løpet av prosjektet.

4.2.2 Arkitektur

I dette prosjektet var det ønsket at arkitekturen var tilnærmet lik den Dataloy bruker ellers i systemet sitt.

Selve applikasjonen som blir beskrevet i detalj i dette kapitlet trenger AIS-data. Applikasjonen kunne hentet dataen selv, men det er nødvendig å lagre dataen over en lengre periode. Dermed er det tatt i bruk en database som lagrer disse dataene. Følgende bilde beskriver arkitekturen på hele løsningen som er satt opp for å håndtere dette problemet.



Figur 3: Viser arkitekturen på den totale løsningen

Det er satt opp to «stand-alone»-applikasjoner, som håndterer data til og fra databasen. Den første applikasjonen (App – 1) er en Java-applikasjon, satt opp til å hente data fra AISHub sitt API. Applikasjonen tar i bruk «Timer» klassen i Java for å hente dataen annethvert minutt. Når dataen er hentet leter applikasjonen etter kundens båter basert på IMO-nummeret. Informasjonen til disse båtene puttes etter hvert inn i databasen. Hovedapplikasjonen sender forespørsler til den andre «stand-alone»-applikasjonen (App – 2). Når denne applikasjonen får en forespørsel henter den data fra databasen og returnerer det til hovedapplikasjonen slik at den har data å vise på kartet.

4.3 Hovedkomponenter i applikasjonen

Applikasjonen er bygd opp av tre hovedkomponenter, som blir opprettet av komponenten `VoyageAnalysisPage`. Den eneste delen applikasjonen som ligger utenfor de tre hovedkomponentene, er filterknappene på menylinjen ovenfor kartet. Disse vil det bli sett nærmere på i delkapittel 4.5. `VoyageAnalysisPage` fungerer som en konteiner for de tre hovedkomponentene, og styrer filtreringen av båtene gjennom filterknappene.

Det er lagt ved en kodesnutt nedenfor dette avsnittet. Denne kodesnutten inneholde JSX-kode, en utvidelse av JavaScript som ligner på XML. JSX blir brukt til å opprette selve komponentene i React. Kodesnutten er en del av «`render()`»-funksjonen til `VoyageAnalysisPage`. I koden ser man de tre hovedkomponentene bli opprettet med de tilhørende parameterne. Komponentene er egne JavaScript klasser som ligger utenfor `VoyageAnalysisPage`. Klassene inneholder alt innhold som er i komponenten, og det trengs dermed kun å kalle på de fra en annen komponent eller funksjon, som i dette tilfelle er `VoyageAnalysisPage`.

Den første komponenten som blir opprettet heter `DrawerRoutes`, dette er en mellomkomponent som sender brukeren til `VoyageAnalysisDrawer`, som er den komponenten som tilhører applikasjonen i dette prosjektet. I dette delkapitlet vil `LeafletMap`, `VoyageAnalysisModalBase` og `VoyageAnalysisDrawer` bli forklart i nærmer detalj.

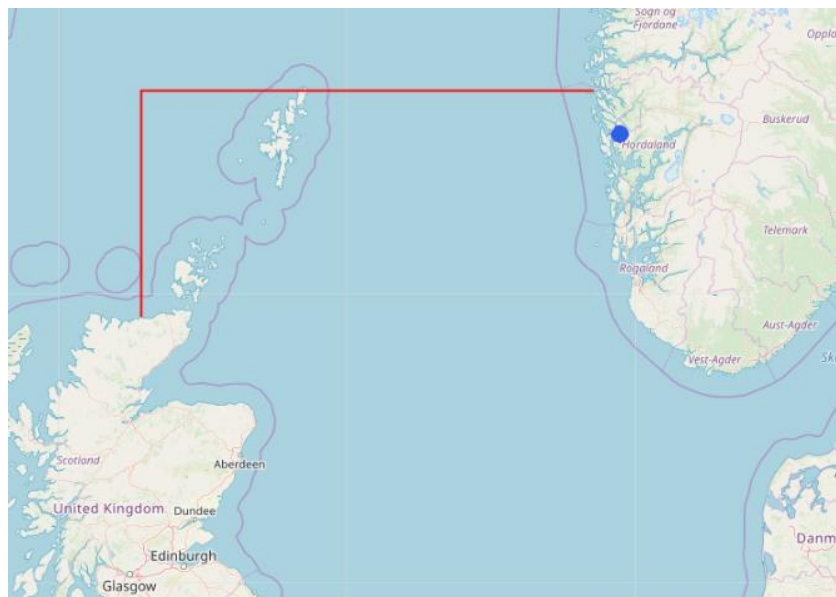
```
<DrawerRoutes />
<LeafletMap
  vesselList={this.state.vesselList}
  onClick={this.onClickVessel}
  mapUpdate={this.mapUpdate}
/>
<VoyageAnalysisModalBase
  isOpen={this.state.isOpen}
  handleClose={this.handleClose}
  handleOpenDrawer={this.handleOpenDrawer}
  xPos={this.state.xPos}
  yPos={this.state.yPos}
  waveDirection={this.state.waveDirection}
  waveHeight={this.state.waveHeight}
  windDirection={this.state.windDirection}
  windSpeed={this.state.windSpeed}
  imoNumber={this.state.imoNumber}
  vesselCourse={this.state.vesselCourse}
  vesselDestination={this.state.vesselDestination}
  vesselName={this.state.vesselName}
  vesselNavstat={this.state.vesselNavstat}
  vesselSpeed={this.state.vesselSpeed}
  vesselStatus={this.state.vesselStatus}
/>
```


4.3.1 LeafletMap

Den første av komponentene som skal diskuteres er LeafletMap. Denne komponenten inneholder det all annen del av brukergrensesnittet enn modalen, skuffen og filterknappene. Komponentene er ansvarlig for å sette opp kartet. Hvordan det gjøres blir forklart i delkapittel 4.4. I tillegg så er den ansvarlig for å hente data om fartøyene og været ved posisjonen til de ulike fartøyene. Når applikasjonen starter så vil denne komponenten kalle på en funksjon som henter disse dataene og plasserer de i en «Redux store».

I tillegg kaller den på en annen funksjon som bruker denne dataen til å opprette polygon- og sirkelikonene til båtene, og returner de til selve komponenten. Dataen blir returnert i form av et GeoJSON objekt. GeoJSON er et format som representerer enkle geografiske objekter, og er basert på JSON (JavaScript Object Notation). Dette gjør at det er lett å håndtere slike objekter i JavaScript. Komponentene oppretter et nytt lag i kartet hvor den plasserer dette objektet. Leaflet kartet vil enkelt kunne håndtere objektet, og plassere alle båtene på riktig kartet.

En annen viktig ting som tilhører komponenten er ECA-sonene (Emission Control Area) som er tegnet på kartet. En ECA-sone er et område der båter må bruke en dyrere form for drivstoff som forurensner mindre. Dette er for å minske forurensende utslipp i enkelte områder.



Figur 4: ECA-sone i Nordsjøen tegnet på kartet som røde linjer. Utklipp fra applikasjonen.

4.3.2 VoyageAnalysisModalBase

Den andre komponenten det skal snakkes om heter VoyageAnalysisModalBase, og er selve modalen som «popper» opp når man trykker på et fartøy. Denne komponenten inneholder mest relative informasjon angående fartøyet som ble trykket på, blant annet hvilket vær det er der fartøyet befinner seg.. Brukeren har mulighet til å lukke modalen ved å trykke utenfor selve modalen, eller på «Close»-knappen.

Parameterne som er tatt med i modalen er plukket ut etter ønske fra Peak Shipping. De ønsket også at modalen skulle være oversiktlig og lett å forstå. Designet nedenfor er basert på Dataloy sitt eget design, og inneholder de nødvendige parameterne. Alle parameterne under er hentet fra en av Peak Shippings egne båter, med unntak av «Previous Port». Denne parameteren er ikke lagret i AIS-signalet som båtene sender ut, men er mulig å få tak i igjennom Dataloy sine egne systemer. Dette er etterspurt av kunden, og parameteren er tatt med slik at dette kan implementeres når applikasjonen kobles opp mot de systemene.

Vessel ID: 9518414

Vessel Info	
Vessel	AIS status
PEAK DUBLIN	Moored
Previous Port	Course
No port was found!	110
Next Port	Speed
HAMBURG	0 kt
Weather info	
Wind	Sea
Direction	Direction
304	336
Force	Height
2.19 m/s	0.15 m

Close
View More

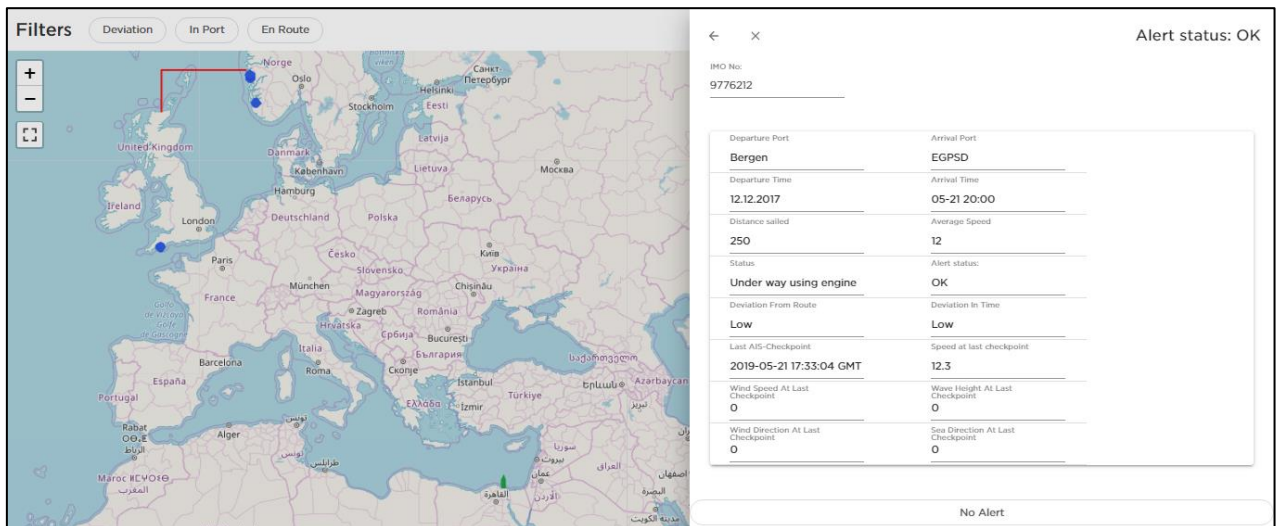
Figur 5: Utklipp av modal-komponenten med ekte data.

4.3.3 VoyageAnalysisDrawer

VoyageAnalysisDrawer er den siste komponenten. Dette er en skuff som åpnes fra høyre side av skjermen og dekker nesten halve skjermen. Denne skuffen inneholder mer detaljert informasjon enn det som finnes i modalen. Noe av dette er informasjon angående avvik fra ruten. Nederst i skuffen er det plassert en knapp med dynamisk tekst. Teksten er ulik basert på om båtene avviker, eller ikke. Dersom en båt har et avvik, vil denne knappen gi brukeren muligheten til å godkjenne avviket dersom det grunnen for avviket er kjent. Bildet under viser forholdet mellom skuff og skjerm. I tillegg er det mulig å se hvordan designet for skuffen er.

Skuffen er linket til gjennom DrawerRoutes komponenten, dette er fordi det er ønsket av Dataloy at skuffer skal linkes til her. Ved å linke til de gjennom denne komponenten, så må det sendes en lenke til DrawerRoutes. Denne lenken blir plassert på slutten av den URL-en brukeren ble sendt ifra.

Informasjon som blir vist i skuffen er hentet fra en «Redux store» ved hjelp av informasjon fra den nye URL-en. Den lenken som ble sendt er dynamisk og inneholder IMO-nummeret til den båten som ble trykket på. Det er dette IMO-nummeret som brukes til å søke etter informasjon i Redux store.



Figur 6: Utklipp av applikasjonen som viser skuffen til høyre.

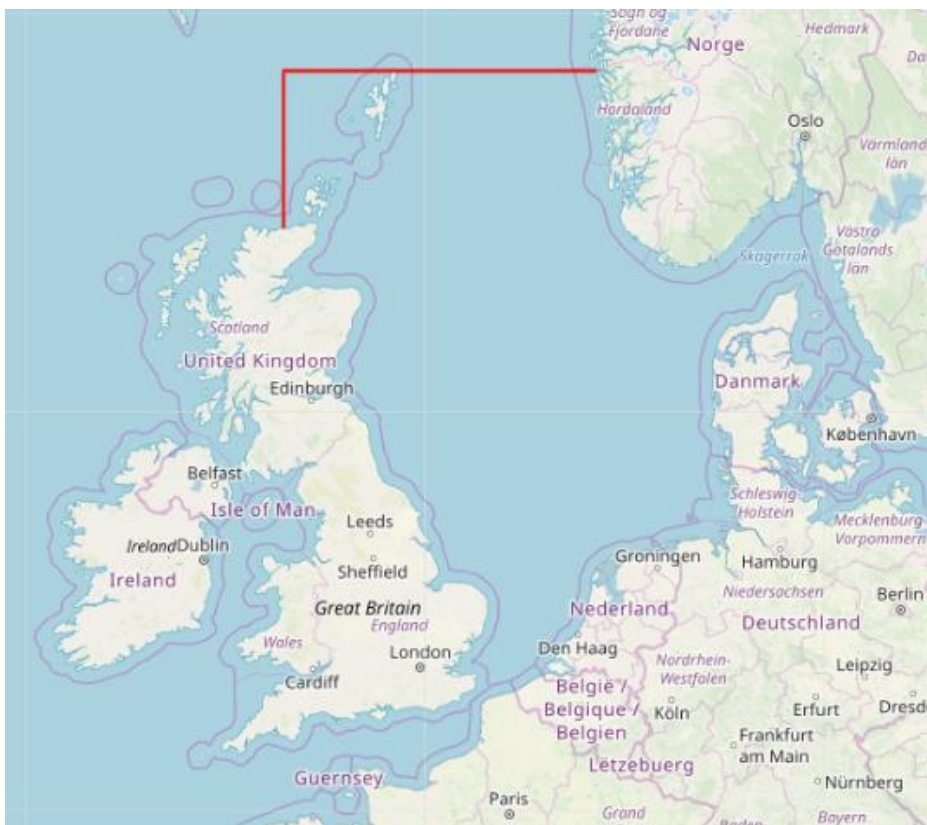
4.4 Leaflet.js

Som diskutert i kapittel 3 så ble det besluttet å ta i bruk Leaflet biblioteket for å opprette kartet. Dette biblioteket er et relativt lite bibliotek på kun 38KB med JavaScript (Leaflet, u.d.).

Dette gjør at kartet er meget lett å ta i bruk, og krever minimalt med kode. For å opprette et tomt kart kreves det kun å opprette en variabel med «L.map()» funksjonen, slik vist i kodesnutten under.

```
this.map = L.map('mapContainer', {  
  fullscreenControl: true,  
}).setView([config.latitude, config.longitude], config.zoom);
```

Denne funksjonen tar en rekke ulike parametere, men i prosjektet er det kun nødvendig å ta i bruk «fullscreenControl», siden dette er etterspurt av kunder fra Dataloy. I tillegg er det lagt til en «setView» funksjon på slutten for å bestemme hvor kartet skal sentreres når det lastes inn. Config.latitude og config.longitude er koordinater som ligger i Nordsjøen. Config.zoom bestemmer hvilket zoom-nivå applikasjonen skal lastes inn med. Den har en verdi på 4,5, som tilsvarer det zoom-nivået på bildet under.

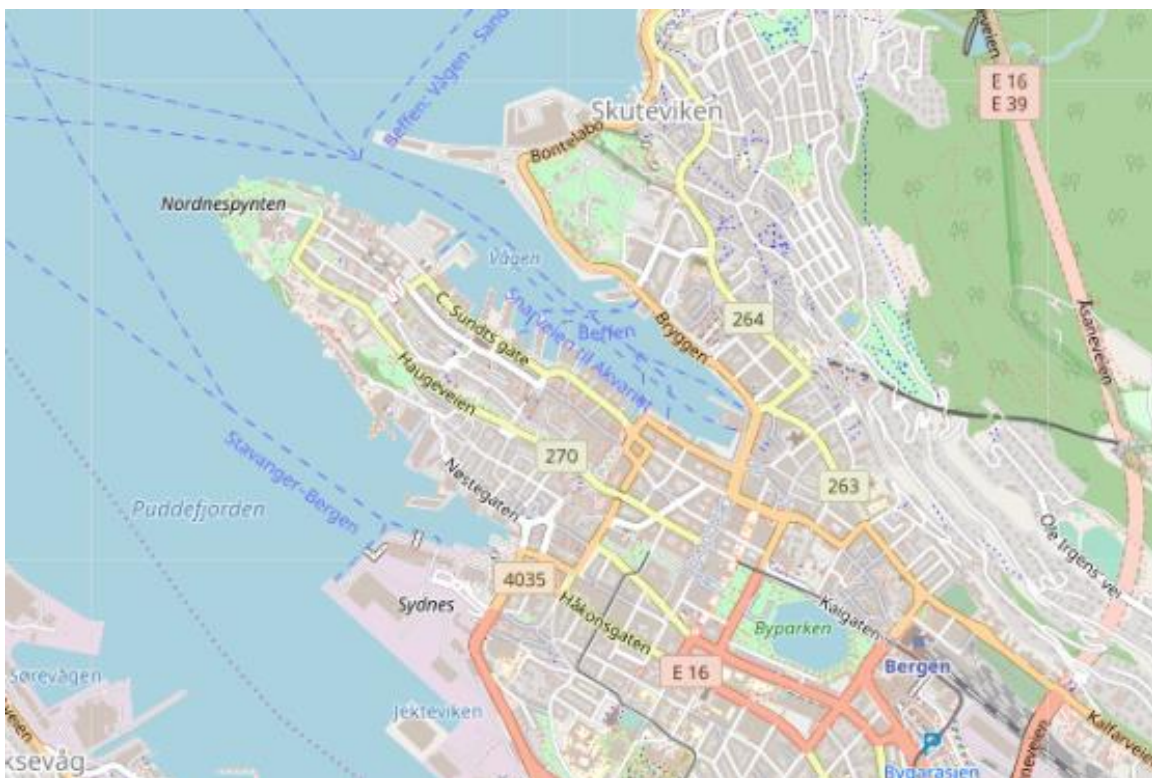


Figur 7: Utklipp som viser hvordan applikasjonen ser ut når den lastes inn

For å legge til tekstur til kartet slik at kartet ikke er helt hvitt kreves det at man legger til «TileLayers» som et lag i kartet. Vi har valgt å bruke OpenStreetMap sin løsning for dette. Det krever en lenke til OpenStreetMap som inneholder fire dynamiske parametere som Leaflet setter selv for å finne hvilke bilder som skal vises i kartet.

```
const tileLayer = L.tileLayer(  
  'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',  
  {  
    attribution:  
      '&copy; <a  
href="http://www.openstreetmap.org/copyright">OpenStreetMap-contributors</a>',  
    maxZoom: config.maxZoom,  
    minZoom: config.minZoom,  
  },  
)  
.addTo(this.map);
```

I tillegg må det settes en minimums- og maksimumsgrense for zoom-nivået. I prosjektet er det ikke nødvendig å kunne zoome inn på gatenivå og maksimumsgrensen er dermed satt til 14, som tillater brukeren å zoome inn til et nivå hvor man tydelig vil se en båt sin plassering i havner. Brukeren vil på denne måten ha mulighet til å zoome raskt inn, uten å ende opp i et zoom-nivå som ikke er til nytte for brukeren ved en slik applikasjon.

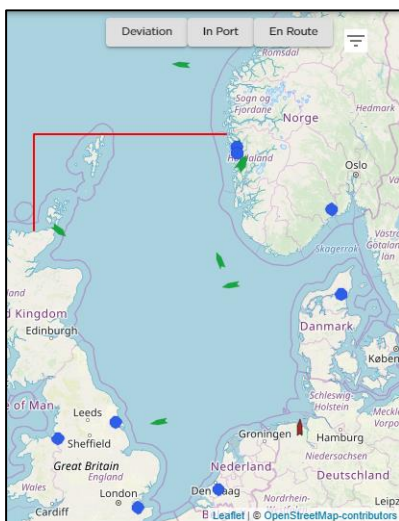


Figur 8: Utklipp fra applikasjonen med kartet sentret på Bergen ved zoom-nivå 14.

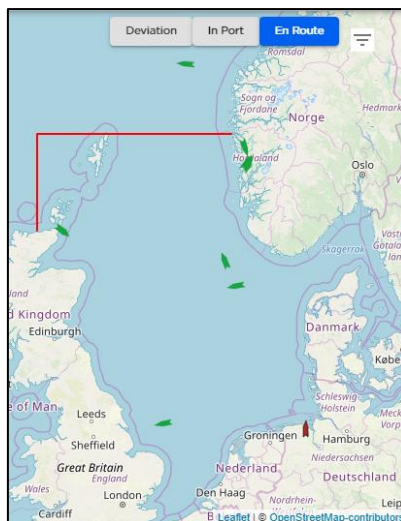
4.5 Brukergrensesnitt

Brukergrensesnittet i applikasjonen består av blant annet de tre komponentene som ble nevnt i delkapittel 4.3. I tillegg finner man en linje med tre filter-knapper på. Disse gir brukeren mulighet til å filtrere fartøy etter status. Brukeren vil kunne zoome inn og ut, samt velge å bruke applikasjonen i fullskjerm modus. I tillegg har brukeren mulighet til å fritt kunne dra rundt på kartet og trykke på de ulike fartøyene som er plassert på kartet. Ved å trykke på et fartøy vil modalen poppe opp, samtidig som kartet rundt får en lett gråfarge over seg. På denne måten er det lett å se hva som er i fokus. Dersom brukeren skulle ønske å få mer informasjon kan det trykkes på «View More»-knappen for å åpne skuffen. De blå fartøyene har rundt form for å skille de fra fartøy på reise, ettersom dette var et ønske fra en kunde av Dataloy. Det er valgt å bruke samme ikon som på det grønne fartøyet på alle fartøy som ikke ligger i havn.

På bildeserien under finner vi tre ulike skjermbilder av applikasjonen. Øverst på applikasjonen ligger «Filters»-linjen. På det første bildet er alle knappene hvite, noe som indikerer at ingen filter er aktiverte. På de to neste bildene er to ulike filter aktiverte, og man ser at båtene på kartet blir filtrerte.



Figur 9: Viser kartet uten filter



Figur 11: Viser kartet med filter "En Route" (i rute)



Figur 10: Viser kartet med filter "In Port" (i havn)

5. EVALUERING

5.1 Evalueringsmetode

Man kan gjerne dele opp evalueringen i to hoveddeler. Evaluering underveis i utviklingen og evaluering av applikasjonen når prosjektet anses for å være ferdig, enten applikasjonen er ferdig eller ikke.

Som tidligere nevnt har prosjektgruppen sittet i åpent landskap sammen med utviklere og sjefen for selskapet og sjefen for utviklingsavdelingen. Dette har gitt en unik mulighet til å få evaluert underveis i prosjektet om design og løsninger er innenfor det som prosjekteieren ønsker seg. Dette er også i tråd med Scrum som er valgt som utviklingsmetode, der man skal levere noe nytt til brukeren underveis i iterasjoner under Scrum utviklingen. Det har ikke vært noen formelle møter eller formelle leveranser av produkt til prosjekteier underveis, hvilket ville vært naturlig dersom man hadde laget applikasjonen for et selskap som ikke var så nærme fysisk som det har vært i dette tilfellet.

I prosjektets slutfase ble det gjennomført demonstrasjon for pilot-kunde og prosjekteier av den fungerende applikasjonen. Det ble stilt fire like spørsmål til de to, og det siste spørsmålet var differensiert ut fra hvem som ble spurt. Spørsmål og svar finner man som vedlegg i kapittel «9.3 – Møtereferater» i denne rapporten. Det ble ikke gjennomført i form av et intervju, men spørsmålene som var skrevet ned på forhånd var ment som hjelpemidler for prosjektdeltagerne som temaer som skulle prates om under evalueringsmøtet. Det ble vurdert å gjennomføre evalueringsmøtet som et intervju. På bakgrunn av at deltagerne i prosjektet og prosjekteier samt pilotkunde anses å være et nært forhold ble det vurdert at det var mer naturlig å ha en åpen prat der deltagerne i prosjektet tok referat fra punktene som kom frem under møtet. Ved å gjennomføre møtet uten fastsatte former håpet man på å få frem mest mulig, og ikke bare være begrenset til å få direkte svar på spørsmålene som ble stilt. Ingen av deltagerne i prosjektgruppen har særlig erfaring fra intervju, og det var derfor usikkert om man ville klare å oppnå de ønskede resultatene fra et eventuelt intervju. Spørsmål ble likevel utformet og medbrakt på møtet for å sikre at de tema som skulle gjennomgås kom med på møtet. En annen viktig årsak til at denne formen ble valgt er at pilotkunden i dette prosjektet har vært en viktig samarbeidspartner for Dataloy i utvikling av nye systemer/moduler tidligere og det var ønskelig at evalueringen av dette prosjektet skulle gjennomføres i en form som var kjent og naturlig for pilotkunden i prosjektet.

Formålet med evalueringen er å dra nytte av erfaring fra prosjektet, samt å dokumentere resultater på en god måte (Difi, 2017).

5.2 Evalueringsresultat

Resultatene fra prosjektet er godt dokumentert gjennom møtereferatene som er vedlagt rapporten – «[9.3 Møtereferater](#)». Det kan være et lite grunnlag å bare spørre en kunde om hva de mener om applikasjonen, dermed burde den videre arbeid eventuelt gjennomføres å vise applikasjonen til noen flere kunder for å få et litt bredere grunnlag å bygge resultatene på.

Basert på referatene kan man konkludere med at begge interessentene stort sett er fornøyde med design og funksjonalitet på den applikasjonen som er laget. I tillegg kan det konkluderes med at begge interessentene ser en mangel i at applikasjonen foreløpig ikke viser historisk rute som båten har seilt, og at denne da ikke kan sammenlignes med ruten fra distansetabellen fra Dataloys systemer.

Det var forskjell på hva interessentene kunne tenke seg å implementere i fremtidige iterasjoner av applikasjonen. Dataloy har noen ideer for å videreutvikle denne applikasjonen som ikke kommer frem fra referatene da dette fortsatt er under evaluering og derfor ikke blir lagt frem. Det kan likevel ikke sies at det har noen stor innvirkning på resultatet av dette prosjektet akkurat hva disse ideene går ut på. Man bør heller vektlegge at det er en applikasjon som Dataloy ser positivt på og ønsker å utvikle videre når man skal evaluere resultatet av dette prosjektet. Dette bidrar til at resultatet i våre øyne styrkes i positiv retning i dette prosjektet. Når det gjelder ønsker fra pilotkunden sin side er disse åpne for allmennheten. Det var allerede klart før prosjektet startet at kunden ønsket å se en visuell fremstilling av ruten som skipet hadde seilt fra forrige havn, og det ble selvsagt også et tema på møtet. Prosjektgruppen informerte om at tiden ikke har strukket til, men at dette er nært forestående å bli implementert.

Pilotkunden ser i hovedsak for seg to bruksområder for denne applikasjonen. Det ene er å ha en visuell fremstilling av alle skipene i rederiet på en storskjerm på kontoret. Applikasjonen viser i dag skipenes posisjon på kartet, og man har implementert forskjellige farger for forskjellig status, og forskjellige ikoner for å vise om skipet seiler eller ikke. Dette var kunden meget fornøyd med og mener at applikasjonen allerede i dag tilfredsstillers minimumsbehovet for å kunne brukes på storskjermen på kontoret.

Videre sier pilotkunden at man kan se for seg at en operatør på kontoret i sitt daglige arbeid kan få god nytte av en slik applikasjon dersom den videreutvikles og videre integreres mer mot det eksisterende systemet som Dataloy leverer i dag. Det kom frem på møtet at operatører til tider bruker tre forskjellige applikasjoner i sitt daglige arbeid. Man bruker en applikasjon for å se hvor båtene befinner seg visuelt, en applikasjon for å sjekke værforhold og til slutt bruker man systemene som Dataloy leverer. Kunden mener det vil være av stor verdi, og trolig tidsbesparende dersom man kan få visuell fremstilling og værmelding inn i systemet som Dataloy leverer i dag. De var svært fornøyd med at applikasjonen allerede viser informasjon om været i akkurat den posisjonen båten befinner seg i, men ytret ønske om at det også skulle være mulig å hente værmelding for noen dager frem i tid.

Begge interessentene var fornøyde med funksjonaliteten at man kan trykke på et ikon så dukker det opp en pop-up på skjermen som viser mer informasjon om skipet. Det var nyttig å se kurs og fart og værdata i denne pop-up-en. Dette var et av målene for prosjektet at man skulle få implementert. Videre skulle det være mulig å trykke på en knapp i denne pop-upen for å åpne en skuff på siden av skjermen som viste mer informasjon. Det skulle være aktuelt dersom båten har en deviasjon og/eller ikke seiler til antatt fart. Begge interessentene var fornøyde med implementeringen av denne knappen og skuffen som vises på siden. I tillegg var de fornøyd med størrelsen på skuffen. Det var viktig at skuffen ikke dekket større del av kartet enn den gjorde i applikasjonen da man fortsatt kunne se store deler av kartet med skuffen på siden åpen.

6. DISKUSJON

6.1 Gjennomføring av prosjektet

Det har vært noen utfordringer med gjennomføringen av dette prosjektet. Enkelte har vært forutsett kunne inntreffe, mens andre har vært vanskelig å påpeke før prosjektstart. At det har vært vanskelig å peke på enkelte av utfordringene har nok med erfaring å gjøre. Begge utviklerne som har deltatt i prosjektet har liten eller i praksis ingen erfaring fra lignende prosjekter. En ting som man har lært gjennom dette prosjektet er at man alltid må være forberedt at uforutsette problemer kan oppstå.

Det var utfordringer med CORS-policy (Cross-Origin Resource Sharing) da man skulle hente AIS-data fra AISHub til klienten som var laget. Dette var en problemstilling deltagerne i prosjektet ikke var kjent med fra før, dermed var det nærmest umulig å forutse at dette skulle bli en utfordring. Det var ikke lett å finne løsningen på utfordringen heller da det var mange andre som hadde vært borti dette tidligere som hadde andre behov enn man hadde i dette prosjektet og de kunne dermed løse det på en annen måte enn det som ble gjort i klienten vår. Problemet var ikke i klienten, men heller i nettleseren, så løsningen på denne utfordringen var faktisk å starte nettleseren i en annen modus enn det man hadde gjort tidligere, ved å slå av et sikkerhetsfilter. Dette er selvsagt ikke noe man kan be kunder av en slik applikasjon om å gjøre. Dette er en utfordring som ikke vil være et problem i det man lagrer AIS-data i en egen database via en back-end-applikasjon ettersom dette utelukker problemer med CORS-policy når klienten henter data fra eget API og egen database.

Selve utviklingen av klienten har gått forholdsvis bra. Komponentene som er valgt i prosjektet har vært tilfredsstillende for denne applikasjonen. Begge deltagerne i prosjektet hadde noe erfaring fra React fra tidligere noe som gjorde at man kom godt i gang. Det var behov for å lese seg opp på kartet, og spesifikt hvordan man skulle få rendret ikoner og linjer på kartet, men det var ikke mer komplisert enn at det gikk når man kom inn i det.

Man har fulgt Scrum i utviklingen og stort sett holdt den planen som var laget på forhånd. Det har vært noen unntak, so da det ble oppdaget underveis avhengighet mellom komponenter og hvilke komponenter som kontrollerte de ulike tingene, måtte man endre på rekkefølgen av hva som ble laget når.

Hierarkiet i prosjektgruppen har vært helt flat. Det har fungert veldig godt og det har sammenheng med at det har vært en liten gruppe med bare to deltagere, slik at man har hele tiden under gjennomføringen har diskutert med hverandre for å løse eventuelle utfordringer. Dersom gruppen hadde vært større ville det vært naturlig å fordele ansvaret på en annen måte. I dette prosjektet har det fungert godt at man har hatt en jevn ansvarsfordeling som har blitt til underveis i prosjektet. Det har stort sett blitt jobbet med det man har ønsket å jobbe med av de tilgjengelige oppgavene. Samarbeidet i prosjektgruppen har fungert meget tilfredsstillende.

6.2 Drøfting av resultater

Evalueringsresultatene fra kapittel 5.2 danner grunnlaget for drøftingen av resultatet i dette kapitlet. Med bakgrunn i evalueringen vil man kunne si at prosjektet har oppnådd ønsket resultat på de delene som ble gjennomført, men samtidig er det viktig å få med prosjektet ikke kom i mål med implementering av alle funksjoner. Det at man ikke kom i mål med alle funksjoner har sammenheng med at prosjektets innhold ble litt endret underveis gjennom at prosjektgruppen selv laget en back-end-løsning for applikasjonen. En del av utviklingstiden ble brukt på dette. Denne tiden kunne vært brukt til å implementere siste funksjonen med historisk data visuelt fremstilt og sammenlignet med ruter fra distansetabellen. Evalueringen viser at applikasjonen danner grunnlag for å gjøre videre arbeid med applikasjonen og at det er en applikasjon som om kort tid kan gi nytteverdi for kunder, og at det på lengre sikt kan gi enda større nytteverdi for kunder å bruke den videreutviklede applikasjonen.

Applikasjonen viser skip i kartet med «live» posisjon hentet fra AIS-data som skipene sender ut. Skipene har forskjellige farger etter status, og det er forskjellige ikoner som illustrere om skipet er underveis eller om det ligger i ro. Applikasjonen viser mer informasjon om skipet dersom man klikker på et ikon. Her får man se informasjon om skipets fart og kurs, samt forrige havn og neste havn. Videre er det implementert værdata for den posisjonen som skipet har. Dersom man trykker videre fra denne informasjonen får man en skuff på skjermen som viser enda mer informasjon om skipets AIS-signaler og informasjon om deviasjon. Med deviasjon menes altså om skipet har valgt en vesentlig lengre rute enn distansetabellen til

Dataloy, eller om skipet seiler med en vesentlig lavere fart enn skipet er oppgitt til å seile med.

Applikasjonen har fått implementert filterknapper som gjør at man kan filtrere på skip som enten seiler, ligger i ro eller som har deviasjon. Dette er en nyttig funksjon dersom det er flere skip på et lite område på kartet da disse skipene da vil vises oppå hverandre og det kan for eksempel skjule seg et skip med deviasjon under skip som ikke har deviasjon.

På tross av at alle målene for applikasjonen ikke ble oppfylt kan man si at prosjektet har vært en suksess. Både prosjekteier og pilotkunde er fornøyd med det som er utviklet så langt, og begge interessentene ser også potensiale i å videreutvikle denne applikasjonen. Det at pilotkunden mener at dagens versjon av applikasjonen allerede kan gi en begrenset nytteverdi må også vektlegges når man trekker konklusjonen at prosjektet har vært en suksess.

6.3 Refleksjon

Det at Dataloy er et relativt lite selskap har antageligvis hatt en påvirkning i forhold til at back-end løsningen ikke ble levert og måtte bli laget av prosjektgruppen. Det var uforutsette ting som skjedde hos Dataloy som krevde øyeblikkelig oppmerksomhet og prioritet over dette prosjektet som trolig var hovedårsaken til at det gikk slik med back-end-løsningen. Dersom det hadde vært et større selskap kan man gjerne anta at det ville vært mindre behov for å omrokere på ressurser dersom noe uforutsett hadde inntruffet, alt avhengig av størrelse og art på det uforutsette.

Det må også vektlegges at deltagerne i prosjektet har minimalt med erfaring fra å planlegge prosjekter og gjennomføre prosjekter. Erfaring med koding fra forskjellige språk og rammeverk er også begrenset for deltagerne i prosjektet. Med bredere bakgrunn innen koding i forskjellige språk kan det tenkes at å lage back-end løsningen ville blitt mindre tidkrevende enn den ble i dette prosjektet. Det har vært spennende og lærerikt å lære seg et helt nytt rammeverk som begge deltagerne har gjort i forbindelse med dette prosjektet. Det kan konkluderes med at å lære seg et rammeverk mer eller mindre parallelt med å gjennomføre et prosjekt er en utfordring i seg selv. Prosjektet hadde blitt mindre tidkrevende dersom man hadde brukt rammeverk som var godt innarbeidet allerede, men med bakgrunn i erfaringen til deltagerne i prosjektet var dette dessverre ikke en reell mulighet. Et av

poengene med prosjektet for deltagerne sin del har også vært å lære seg inngående et rammeverk som blir brukt til å lage klienter i dagens marked.

7. KONKLUSJONER OG VIDERE ARBEID

7.1 Konklusjon

Basert på kapittel 5 og 6 legger vi til grunn at prosjektet har laget en applikasjon som i sin nåværende form, har begrenset nytteverdi for Dataloy sine kunder. Videre konkluderer vi med at det er en applikasjon med stort potensiale for både Dataloy og deres kunder dersom den videreutvikles.

I den endelige applikasjonen inngår følgende elementer:

- Kart som viser skipene til kundene.
- Mulighet for å klikke på et skip for å få en modal med mere informasjon om skipet underveis.
- Kunne klikke på en knapp i modalen for å undersøke videre informasjon om deviasjon for skip.
- Kunne filtrere ut skip for å få et bedre overblikk over skip i kartet.

7.2 Videre arbeid

Prosjektet i seg selv avsluttes med at bacheloroppgaven leveres inn. Det er imidlertid bestemt hos Dataloy at applikasjonen skal videreutvikles, og man har sett andre områder i den eksisterende programvaren der fragmenter av denne applikasjonen kan brukes. I den eksisterende programvaren som Dataloy leverer til sine kunder i dag er det en del av programmet hvor man kan lese informasjon om generiske data for de forskjellige skipene. Det er ønske om at man skal bruke applikasjonen i dette prosjektet som en base for å lage en funksjon i denne modulen der man kan trykke på en knapp på siden for å vise båtens aktuelle posisjon «live» i et kartutsnitt. Man har også diskutert internt i selskapet, samt forhørt seg med kunder, og fått bekreftet at det vil være av nytteverdi dersom denne applikasjonen kan sende en melding til rederen dersom AIS-status skifter fra status om at båten seiler, til at den er fortøyd, slik at rederen kan få denne informasjonen om at båten er fortøyd mye raskere enn tidligere. Det vil likevel ikke være en fasit på når båten er fortøyd da det er noe som må oppdateres manuelt på broen av mannskapet om bord. For å videreutvikle applikasjonen etter endt prosjekt er begge deltakerne i prosjektet tilbudt jobb etter endt prosjekt for å jobbe videre med applikasjonen.

8. LITTERATUR/REFERANSER

8.1 Referanser

Automatisk identifikasjonssystem (2016) [Internett]

Tilgjengelig fra: https://no.wikipedia.org/wiki/Automatisk_identifikasjonssystem
[Lest 26. mars 2019].

Brækhus, S. (2018) *Certeparti*. [Internett] Tilgjengelig fra: <https://snl.no/certeparti>
[Lest 1. april 2019].

Bush, T. (2018) *5 Powerful Alternatives to Google Maps API*. [Internett]

Tilgjengelig fra: <https://nordicapis.com/5-powerful-alternatives-to-google-maps-api/>
[Lest 15. mai 2019].

Cai, Y. & Wen, Y. (2014) *Ship Route Design for Avoiding Heavy Weather and Sea Conditions*.
[Internett] Tilgjengelig fra:

http://www.transnav.eu/Article_Ship_Route_Design_for_Avoiding_Cai,32,540.html
[Lest 17. mai 2019].

Dataloy (u.d) *About Dataloy*. [Internett] Tilgjengelig fra: <http://dataloy-systems.com/aboutdataloy/>

[Lest 1. april 2019].

Difi (2017) *Evaluere prosjektet og utarbeide sluttrapport*. [Internett]

Tilgjengelig fra: <https://www.prosjektveiviseren.no/avslutningsfasen/evaluere-prosjektet-og-utarbeide-sluttrapport>

[Lest 1. juni 2019].

Leaflet (u.d) *Leaflet*. [Internett] Tilgjengelig fra: <https://leafletjs.com/>

[Lest 2. april 2019].

Lines, B. (2015) *What's transmitted on the AIS signal?*. [Internett]

Tilgjengelig fra: <https://www.marinetraffic.com/blog/information-transmitted-via-ais-signal/>
[Lest 26. mars 2019].

MarineTraffic (u.d) *What is the Automatic Identification System (AIS)?*. [Internett]

Tilgjengelig fra: <https://help.marinetraffic.com/hc/en-us/articles/204581828-What-is-the-Automatic-Identification-System-AIS->

[Lest 26. mars 2019].

OpenLayers (u.d) *OpenLayers*. [Internett]

Tilgjengelig fra: <https://openlayers.org>
[Lest 15. mai 2019].

OpenStreetMap (u.d) *OpenStreetMap*. [Internett] Tilgjengelig fra: <https://www.openstreetmap.org>

[Lest 15. mai 2019].

Prettier (u.d) [Internett] Tilgjengelig fra: <https://prettier.io/>

[Lest 1. juni 2019].

React (u.d) [Internett] Available at: <https://reactjs.org/>
[Lest 23. mars 2019].

Redux (u.d) [Internett] Tilgjengelig fra: <https://www.redux.js.org/>
[Lest 23. mars 2019].

Sellers, D. (2018) *Google Maps has 67% of the navigation app market compared to Apple Map's 11%*. [Internett] Tilgjengelig fra: <https://www.appleworld.today/blog/2018/7/10/google-maps-has-67-of-the-navigation-app-market-compared-to-apple-maps-11>
[Lest 7. mai 2019].

8.2 Ordliste

AWS-server – Amazon Web Services-server er Amazon sin skyløsning.

Backend - Tilsvarende serverapplikasjonene og databasene i en web-applikasjon, områder der dataen blir håndtert.

Bitbucket – Et versjonskontrollprogram.

Certeparti - Kontraktsdokument som utferdiges ved slutning av avtaler om leie av skip og visse fraktavtaler (Brækhus, 2018).

GeoJSON – GeoJSON er et objekt som beskriver en geografisk lokasjon i JSON-format

Hardkode – Å hardkode vil si at man plasserer linker eller andre verdier direkte i koden, i stedet for å opprette en variabel der man setter verdien gjerne i en egen fil.

Iterasjon – Å iterere vil si å repetere en oppgave flere ganger.

JIRA – Et program som brukes for å ha oversikt over fremdriften i den agile utviklingen av et program / en applikasjon.

Komponent – En del av en applikasjon.

Maskinvare – Fysiske deler i en datamaskin.

Modal – Vinduet som dukker opp når brukeren klikker på en båt.

Props – Verdier som blir sendt fra forelder-komponent til barn-komponent.

Rendre – Å tegne noe på skjermen.

Repository – Et område i et versjonskontrollprogram

Scrum – Scrum er en agil utviklingsmetodikk

Skuff – Skuffen som dukker opp når brukeren klikker på «View More» i modalen.

Skyløsning – Er en løsning for å lagre data i en «sky». Denne skyen er en server som leverandøren av skyløsningen har, og dermed slipper brukeren å ha servere for å lagre dataen selv.

Software – Programvare.

State – Tilstanden til en komponent i React-applikasjoner. Forandres denne rendres komponenten på nytt.

8.3 Akronymer

AIS – Automatic Identifikasjon System (automatisk identifikasjonssystem) er et antikollisjonshjelpemiddel for skipsfarten (Wikipedia, 2016).

HTML – HyperText Markup Language.

IDE – Integrated Development Environment er et programmeringsverktøy.

JSON – JavaScript Object Notation.

JSX – JavaScript XML.

VVAT – Vessel Voyage Analysis Tool er navnet på applikasjonen som blir utviklet gjennom prosjektet.

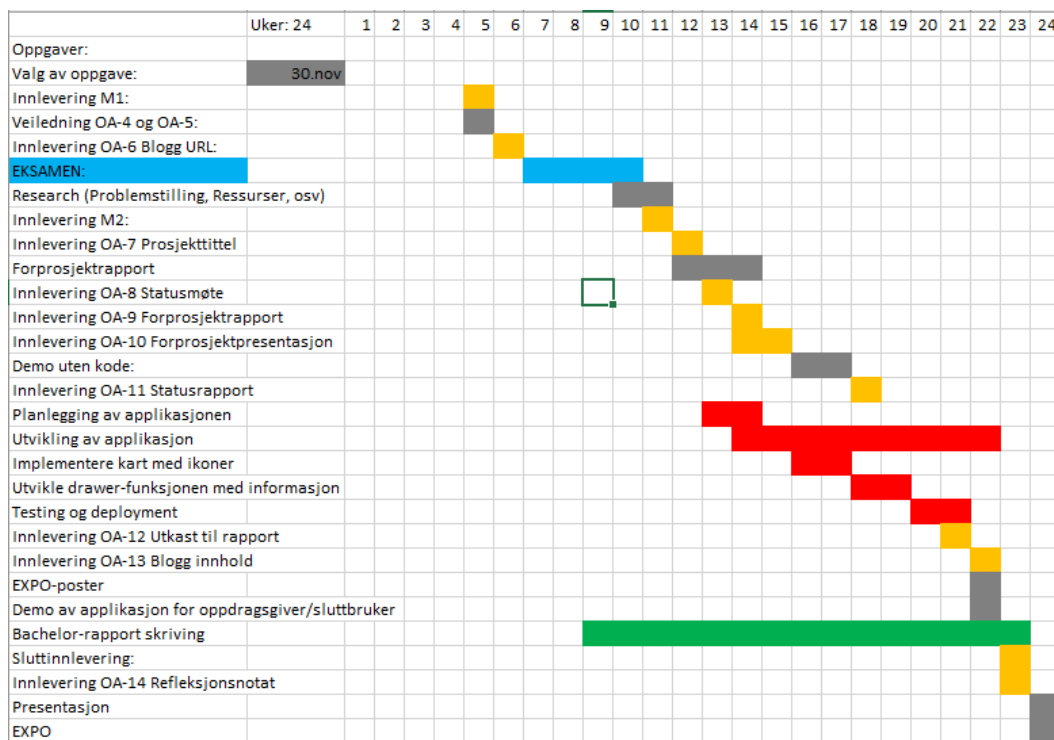
XML – Extensible Markup Language.

9. VEDLEGG

9.1 Prosjektplan

Vedlagt ligger en fremdriftsplan i form av et Gantt-diagram. Dette diagrammet inneholder ulike fargekoder:

- Oransje – Innleveringer
- Grå – Annet
- Blå – Eksamensperiode
- Rød – Utviklingsfase
- Grønn – Rapportskrivning



Figur 12: Gantt-diagram

9.2 Risikoanalyse

I tabellen under vil følgende forkortelser bli benyttet:

- S – Sannsynlighet
- K – Konsekvens
- RF – Risikofaktor, $S * K$

Risiko	S	K	RF	Tiltak	Interessent
Applikasjonen er ikke til nytte for bedriften	1	2	2	Møter med bedriften og kunder.	Bedriften
Ikke brukervennlig for kunden	3	4	12	Møter med kunden slik at de kan se hva som må til for at de skal kunne bruke den. Lage en prototype og demonstrer prototype til kunden, få tilbakemeldinger på prototypen og få bekreftelse på at design og funksjonalitet stemmer over ens med krav	Bedriften
Prosjektets omfang	2	4	8	Sikre at ressurser brukes effektivt, spesielt i starten. Godt utgangspunkt vil gjøre det lettere mot slutten.	Gruppen
Gruppesamarbeid	1	5	5	Møte på kontoret og ha kontinuerlig kommunikasjon, slik at man sikrer godt samarbeid. Kommunikasjon gjennom sosiale medier for å ha en åpen dialog hele veien.	Gruppen
Tekniske problemer	3	3	9	Kommunisere med bedriften slik at man kan få hjelp tidlig og sikre at problemene ikke blir for store.	Bedriften Gruppen
Ressursbehov	3	4	12	Dialog med bedriften for å kartlegge behov for ressurser.	Gruppen
Sykdom i gruppen	1	3	3	Jobbe hjemmefra til den grad det er mulig, og ta igjen det som ble tapt. God kommunikasjon.	Gruppen
Tilgang til ekstern data - AIS	3	5	15	Teste tidlig at man får til å hente AIS-data fra AIS-hub. Bytte til annen leverandør av AIS-data dersom AIS-hub ikke fungerer.	Gruppen
Server/Back-end	1	5	5	Holde dialogen med bedriften som skal sørge for at server og back-end fungerer som det skal, og at API fungerer tilfredsstillende	Bedriften Gruppen

9.3 Møtoreferater

9.3.1 Møtoreferat fra evalueringsmøte med pilotkunde

Spørsmål til pilotkunde:

1. Hva tenker dere om designet på applikasjonen?
2. Er det noe ved designet dere mener burde vært annerledes?
3. Har applikasjonen de funksjonene dere hadde forventet?
4. Er det noen funksjoner dere savner eller noen funksjoner dere mener er overflødige?
5. Dersom applikasjonen blir videreutviklet og deler av (eller alt) det som dere etterlyser i foregående spørsmål blir implementert, er dette en applikasjon dere kan ha nytte av i bedriften deres?

1.

- Designet er enkelt og bra.
- Designet er oversiktlig.
- Størrelse på pop-up og skuff er bra.
- Det er bra at det er lett å skille båter som seiler fra båter som ligger i havn ved at det er forskjellig ikon med forskjellig farge.

2.

- Det burde vært båtnavn på/ved ikonet for å lettere identifisere båtene på kartet.
- Hvis det er flere båter oppå hverandre burde de gjerne vært samlet til et ikon.
- Det hadde vært ønskelig om ikonene blir litt mindre når man zoomer langt ut på kartet.
- Det kunne med fordel vært fjernet litt informasjon på kartet fra «innlandet» på kontinentet for å gjøre utseende enda renere.

3.

- Det er bra at det er fullskjermsmodus.
- Zoom-funksjonen fungerer som den skal.
- Det er veldig nyttig å kunne se hvilket vær båten opplever i nåtid på sin nåværende posisjon.

4.

- Det er ønskelig at ikonene er klikkbare i fullskjermsmodus.
- Dersom mulig sjekke antatt ETA mot ETA som er rapportert i AIS og varsle ved avvik over en hvis grense.
- Det er ønskelig med hale med historikk på hvor skipet har seilt og kunne sammenligne denne med distansetabellen fra DDT.
- Det er ønskelig at man kan filtrere på hvilken type skip man ønsker å se i kartet.
- Det er ønskelig å definere en egen gruppe med skip som man kan filtrere på.
- Det er ønskelig å kunne klikke for å få frem værmelding for de neste 3-4 dagene når en båt ligger i havn, for å kunne gjøre antagelser om hvor lenge båten ligger i havn.
- Det er ønskelig med informasjon om hvilken last skipet frakter på den aktuelle reisen og det hadde vært fint å kunne klikke på en lenke i skuffen og komme direkte til den aktuelle reisen i VMS.

5.

- Dette er absolutt en interessant applikasjon. Versjonen som allerede er laget kan gi nytte ved å vise båtene på storskjerm i kontorlokalet.
- Dersom det blir integrert mer med VMS vil dette være noe som operatører kan få nytte av i sitt daglige arbeid på kontoret.

9.3.2 Møtereferat fra evalueringsmøte med oppdragsgiver

Spørsmål til oppdragsgiver:

1. Hva tenker dere om designet på applikasjonen?
2. Er det noe ved designet dere mener burde vært annerledes?
3. Har applikasjonen de funksjonene dere hadde forventet?
4. Er det noen funksjoner dere savner eller noen funksjoner dere mener er overflødige?
5. Er denne applikasjonen noe dere ser for dere at dere kan bygge videre på og bruke i deres kommersielle drift fremover etter at dette prosjektet er avsluttet?

1.

- Det er et design som ser bra ut og som vi tror vil være gjenkjennelig for Dataloys kunder som bruker andre systemer vi leverer.

2.

- Litt usikker på design/brukergrensesnitt på filterknappene da dette er noe som avviker fra våre andre filter funksjoner i andre applikasjoner. Det er imidlertid litt annerledes filter på kartet enn de andre applikasjonene ved at her er det bare et filter om gangen som slås av eller på, mens de andre applikasjonene har mulighet for å filtrere flere parametere og flere filter samtidig. Dette må undersøkes med flere kunder før man kan konkludere med noe endelig.

3.

- De funksjonene som er implementert fungerer som forventet og det er ikke noe som er overflødig på det nåværende tidspunkt.
- Vi hadde håpet at det skulle være mulig å vise historisk data visuelt i kartet på hvor båten har seilt, men har forståelse for at tiden som kunne vært brukt til dette har gått til å lage en back-end-løsning som egentlig ikke var ment å inngå i prosjektet fra starten av.

4.

- Det er ønskelig at applikasjonen skal vise historisk rute i kartet og visuelt sammenligne denne med ruten fra distansetabellen.

5.

- Dette er en applikasjon vi kommer til å fortsette å bygge videre på og en applikasjon som vi ser potensiale i.
- Etter at prosjektet startet opp har vi snakket med litt kunder og fått litt nye ideer til bruk av applikasjonen og også videreutvikling av applikasjonen.
- Etter at prosjektet startet opp har vi kommet opp med noen nye ideer for videreutvikling av denne applikasjonen.