

PROSJEKTRAPPORT GRUPPE 17

IN2000 – VÅR 2019

BØLGEN – VÆRET TIL SJØS



Abenayan Ketheeswaran – abenayak

Eirik Størdal – eirigs

Vilde Husby – viderhu

Ahmed Abdulrahman – aaabbas

Kristian Gyene – kristsgy

Celine ten Dam – celinete

Gruppelærer – Erlend Stenlund



Innholdsfortegnelse

Innledende sammendrag.....	4
Presentasjon	4
<i>Utviklingsteamet.....</i>	<i>4</i>
<i>Oppdragsgiver.....</i>	<i>4</i>
<i>Presentasjon av case.....</i>	<i>4</i>
<i>Vår løsning.....</i>	<i>5</i>
Brukerdokumentasjon.....	5
<i>Systemets funksjonalitet.....</i>	<i>5</i>
Målgruppe og aktører	5
Plattform og aksessering av applikasjonen.....	5
Skjermbilder av applikasjonen	6
Kravspesifikasjon og modellering.....	9
<i>Use case diagram</i>	<i>9</i>
<i>Tekstlig beskrivelse.....</i>	<i>11</i>
<i>Sekvensdiagram.....</i>	<i>12</i>
<i>Klassediagram.....</i>	<i>13</i>
<i>Design pattern.....</i>	<i>14</i>
<i>Objektorienterte prinsipper.....</i>	<i>15</i>
Produktdokumentasjon.....	16
<i>Løsningen fra et teknisk ståsted</i>	<i>16</i>
<i>Teknologier og arkitektur.....</i>	<i>18</i>
Testdokumentasjon.....	18
<i>Testing for hver sprint</i>	<i>18</i>
Statisk testing	19
Enhetstesting.....	19
Integrasjonstesting	20
<i>Testing av ikke-funksjonelle krav</i>	<i>20</i>
<i>Testing med brukere.....</i>	<i>22</i>
Brukertest	22
Akseptansestest.....	22
<i>Universell utforming.....</i>	<i>23</i>
<i>Sikkerhet.....</i>	<i>24</i>
Prosessdokumentasjon.....	25
<i>Metode og utviklingsmetodikk</i>	<i>25</i>
<i>Beskrivelse av oppstartsprosessen</i>	<i>26</i>

<i>Sprint 1</i>	28
Sprint planleggingsmøte.....	28
Sprint Backlog.....	28
Datainnsamling.....	28
Prototype.....	29
Sprint Review.....	30
Sprint Retrospective.....	30
Product Backlog etter sprint 1:.....	31
<i>Sprint 2</i>	32
Sprint planleggingsmøte.....	32
Sprint Backlog.....	32
Daglig Scrum.....	33
Sprint Review.....	34
Sprint Retrospective.....	34
Product Backlog etter sprint 2:.....	34
<i>Sprint 3</i>	36
Sprint planleggingsmøte.....	36
Sprint Backlog.....	36
Daglig Scrum.....	37
Sprint Review.....	37
Sprint Retrospective.....	37
Product Backlog etter sprint 3:.....	38
<i>Sprint 4</i>	39
Sprint planleggingsmøte.....	39
Sprint Backlog.....	39
Daglig Scrum.....	40
Sprint Review.....	40
Sprint Retrospective.....	41
Product Backlog etter sprint 4:.....	41
<i>Sprint 5</i>	42
Sprint planleggingsmøte.....	42
Sprint Backlog.....	43
Daglig Scrum.....	44
Sprint Review.....	44
Sprint Retrospective.....	44
Product Backlog etter sprint 5:.....	45
<i>Sprint 6</i>	46
Sprint planleggingsmøte.....	46
Sprint Backlog.....	46
Daglig Scrum.....	47
Sprint Review.....	48
Sprint Retrospective.....	48
Product Backlog etter sprint 6:.....	48
<i>Sprint 7</i>	49
Sprint planleggingsmøte.....	49
Sprint Backlog.....	50
Daglig Scrum.....	51
Sprint Review.....	51
Sprint Retrospective.....	51

Product Backlog etter sprint 7:.....	52
<i>Sprint 8</i>	53
Sprint planleggingsmøte.....	53
Sprint Backlog.....	54
Daglig Scrum.....	54
Sprint Review.....	54
Sprint Retrospective.....	55
Product Backlog etter sprint 8:.....	55
<i>Sprint 9</i>	56
Planleggingsmøte	56
Sprint Backlog.....	56
Daglig Scrum.....	58
Sprint Review.....	58
Sprint Retrospective.....	58
Product Backlog etter sprint 9:.....	58
I hvilken grad ble kravspesifikasjonene oppfylt?	59
Avsluttende refleksjon	60
Kilder	62
Vedlegg	63

Innledende sammendrag

Vi er en gruppe studenter ved Universitetet i Oslo som tar emnet IN2000 - Software Engineering med prosjektarbeid. Denne rapporten tar for seg vårt prosjekt og vår prosess som har resultert i applikasjonen "Bølgen".

Presentasjon

Utviklingsteamet

Teamet vårt består av seks studenter. Vilde og Celine fra design, bruk og interaksjon, og Eirik, Abenayan, Kristian og Ahmed fra programmering og systemarkitektur. Ikke alle på gruppa kjente hverandre fra før, men å jobbe sammen med nye mennesker så vi på som en positiv erfaring for alle. Sammen er vi en god blanding med ulike erfaringer og interesser, men med like ambisjoner for faget og hva vi ønsker å oppnå. Dette så vi på som et fint utgangspunkt for et godt samarbeid i prosjektet.

Oppdragsgiver

Oppdragsgiveren for caset og vårt prosjekt er faglærerene Yngve Lindsjørn, Stein Michael Storleer og Henrik Hillestad Løvold. Videre har også gruppens product owner en rolle som oppdragsgiver overfor resten av teamet.

Presentasjon av case

For dette prosjektet har vi valgt å utvikle en app for case 1 som er følgende:

Norge er en kystnasjon, og i alle tider har nordmenn vært på havet, både i arbeid og som fritidssyssel. Været til sjøs kan være lunefullt, og raske skifter fra pent og vindstille til sludd og storm i kastene er noe alle som ferdes til sjøs må kunne håndtere. Et pålitelig og godt varsel for været på sjøen er livsviktig for alle som ferdes der, enten det er i arbeid eller for fornøyelsens skyld. I dette prosjektet skal dere utvikle en app for Android som varsler været på havet fremover i tid og strekning.

Grunnen til at vi valgte dette caset var først og fremst fordi vi syntes det var et spennende case. Vi følte også det var lettere å komme med kreative valg siden været til sjøs omfatter mye forskjellige type informasjon. En annen faktor som gjorde at vi valgte dette caset var målgruppen. Det var enklere for oss å få kontakt med potensielle brukere i dette caset. Vi kjenner en del som bruker eller har brukt båt og som kanskje kunne hatt bruk for en sånn applikasjon. En annen fordel med dette caset er vi ser for oss at vi kan lage en applikasjon som vil være relevant for bruk hele året.

Vår løsning

Vår løsning er “Bølgen”. Her presenteres fire sider som alle gir forskjellig informasjon til brukeren. En “nå”-side, “time”-side, “uke”-side og “kart”-side. I tillegg til disse har vi også en sidemeny med meny. applikasjonen tar utgangspunkt i brukerens posisjon, posisjonen kan også endres ved å velge sted på kartet. Nå-siden gir brukeren en oversikt over forskjellige værparametre der brukeren kan selv velge hvilke værparametre han vil se. Vind, bølgehøyde og sikt har vi satt som standard fordi disse er ansett som de viktigste. På nå-siden har vi også en trygghetsskala som en tilleggsfunksjon. Denne illustrerer hvor trygt det er på den valgte posisjonen basert på typen til båten som brukeren har. Båttypen kan bruker endre på i menyen. På time-siden viser applikasjonen en time for time varsling. Her vises værmelding for vind og bølgehøyde i egne “bokser” for hver time nedover siden. Ved å klikke på disse får bruker opp mer informasjon om været for den timen. I uke-siden er det samme layout, men her er det kun informasjon om bølgehøyde og vind. På kart-siden kan bruker klikke på et tilfeldig sted på kartet og se værvarsler for valgt sted. Som en tilleggsfunksjon i kartet har vi lagt til en visuell oversikt over regn og vind på valgt posisjon og område rundt. I tillegg til en oversikt over de største havnene i Norge. Bruker kan selv velge hvilken måleenhet i menyen. En annen tilleggsfunksjon vi har tatt med er at brukeren kan kontakte vaksentral ved hjelp av en SOS slide-knapp. Denne sender melding til vaksentral med brukerens koordinater.

Brukerdokumentasjon

Systemets funksjonalitet

Målgruppe og aktører

Målgruppen for vår app er nordmenn som bruker båt i arbeid eller som fritidssyssel. Målgruppen er systemets primær aktør. En sekundær aktør til applikasjonen vår er en vaksentral til sjøs som blir kontaktet om en bruker av applikasjonen havner i trøbbel. Vi har ikke satt oss inn i hvilken vaksentral vi eventuelt ville kontaktet dersom applikasjonen skulle blitt realisert, dermed referer vi til denne interessen som ”vaksentral”. En annen sekundær aktør til systemet vårt er MET sine APIer. Spesifikt benytter vi oss av oceanforecast, locationforecast, spotwinds, radar, textforecast og tidalwater. En tredje sekundær aktør til systemet vårt er Google. Vi benytter oss av deres kart, og via dette bruker vi også deres funksjonalitet for innhenting av ”min posisjon”.

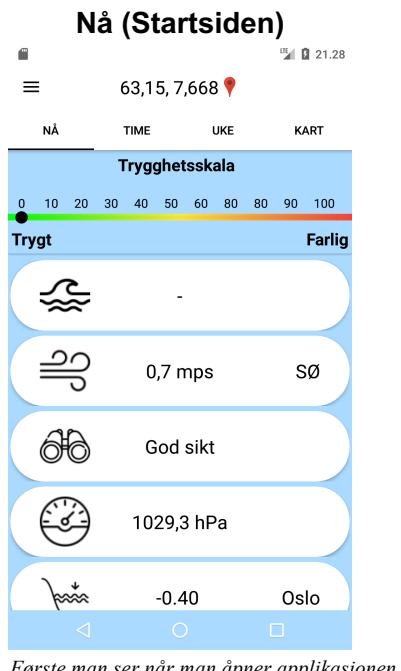
Plattform og aksessering av applikasjonen

Applikasjonen vår er kun tilgjengelig på Android plattformen. Videre krever den minst API nivå 24 for at Google Maps skal fungere. Applikasjonen vår ligger tilgjengelig i gruppens git repository ”Gruppe17”. For å innhente informasjon fra MET sine APIer bruker vi en proxy-server som tilhører UiO i stedet for å gå direkte via api.met.no. Dette er et krav vi har fått fra oppdragsgiver. Når emnet er ferdig vil denne proxy-serveren bli tatt ned, og man vil ikke lengre få tilgang til informasjon i applikasjonen. Første gang man kjører applikasjonen skal

det kommer opp vinduer som spør etter tilgang til å sende SMS og tilgang til posisjon. Disse er ikke nødvendig for at applikasjonen skal fungere, men kreves for å få full funksjonalitet. Videre må bruker velge båttype for at trygghetsskalaen, en funksjon som viser om det er trygt for en båt eller ikke, skal fungere.

Skjermbilder av applikasjonen

Systemets funksjonalitet har blitt løst på følgende måte på hensyn til use-case diagrammet:



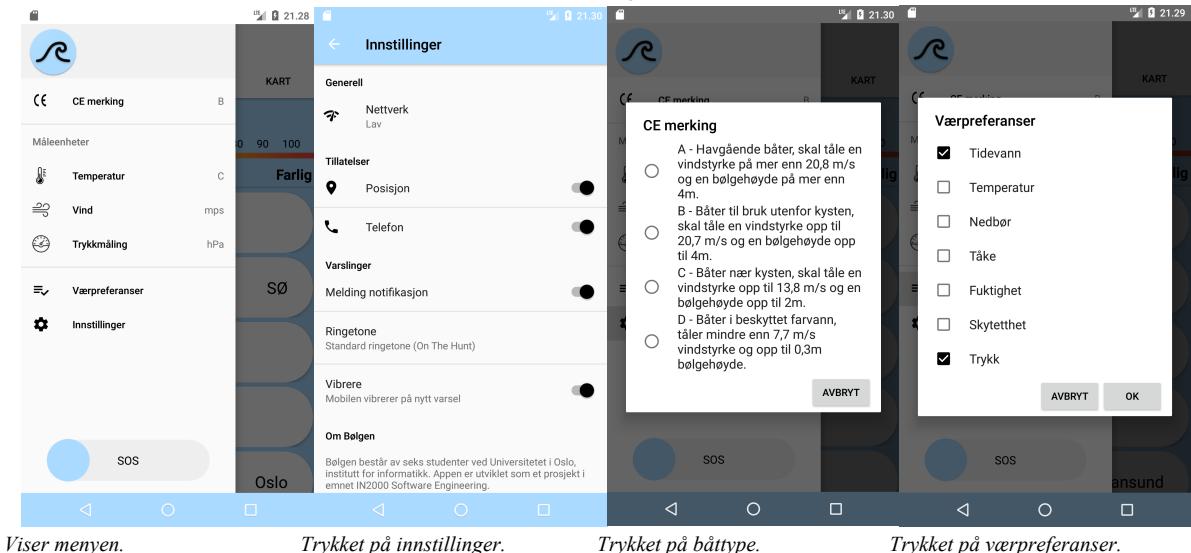
Første man ser når man åpner applikasjonen.

Første gang man åpner applikasjonen får man opp spørsmål om bruk av tillatelse til bruk av posisjon og om appen kan sende SMS. Deretter får man opp en valgmeny der man må velge båttype.

Startsiden gir brukeren data om de ulike værparametere: vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet. Disse dataene hentes fra MET sitt API og visualiseres på nå-siden med passende ikoner. Siden har også en trygghetsskala som regner ut om det er trygt for brukeren. På toppen av vinduet vises nåværende koordinater som oppdateres hver gang bruker åpner applikasjonen. Koordinatene endres dersom brukeren ønsker å se værvarsling for et valgt sted. For å komme seg til de andre sidene kan bruker velge å sveipe til venstre eller trykke på de ulike fanene under koordinatene. Trykker brukeren på ikonet øverst til venstre kommer opp en meny med ulike filter og innstillinger.

Krav som oppfylles fra product backlog: "se værmelding for nåværende/valgt posisjon", "se trygghetsskala for nåværende/valgt posisjon", og "oppdaterer posisjon når applikasjonen åpnes".

Meny



Viser menyen.

Trykket på innstillingen.

Trykket på båttype.

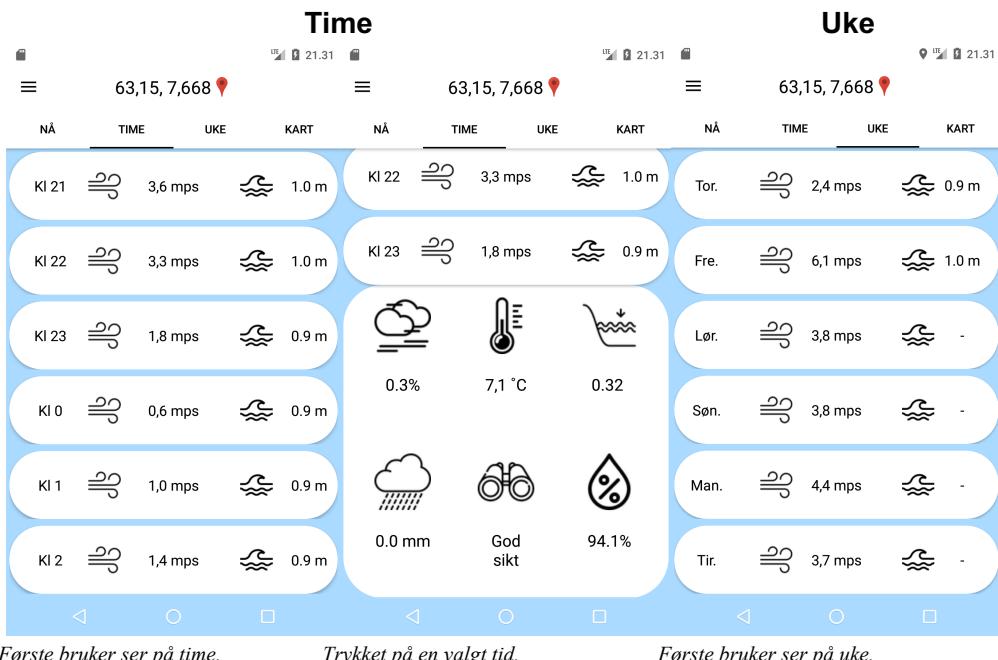
Trykket på værpreferanser.

Menyen gir brukeren mulighet til å endre preferanser. Ved å trykke på båttype og huke av én endres båttypen. Det er fire forskjellige type båter fra A til D der A tåler mest og D tåler minst., For båttype A fantes det ingen “maks” verdi, derfor satt vi denne til å tale alt fra Beauforts skala definerer som orkan. Da tåler de bølgehøyde på 16 meter og vindstyrke på 32,6 meter per sekund. Den største av disse to verdiene blir brukt for å gi verdi til trygghetsskalaen. Det er også mulig å endre måleenheter for temperatur, vind og trykk. For temperatur kan bruker velge mellom Celsius eller fahrenheit, for vind kan bruker velge mellom meter per sekund eller kilometer i timen. Når bruker velger trykk er valgene hektopascal (hPa), millibar, bar eller millimeter kvikksølv (mmHg). Første gang bruker starter applikasjonen er standardverdiene Celsius, meter per sekund og hektopascal. I værpreferanser kan bruker velge hvilke værparametere som skal vises på nå-siden. Værpreferansene vind, sikt og bølgehøyde er standard på nå-siden så de kan ikke fjernes.

Innstillinger fører brukeren til en ny side. Bruker kan endre hvor stort nettverk man ønsker av målingspunkter på kartet. Bruker kan endre applikasjonens tilgang til posisjon og SMS. Beskrivelsen av applikasjonen er en liten tekst om hvem som har utviklet applikasjonen og i hvilken sammenheng vi jobbet med applikasjonen. Det står også hvilken versjon av applikasjonen som er utviklet. Bruker kan også sende oss en tilbakemelding på mail ved å klikke på tilbakemelding. Det siste i innstillingen er en side med regler for bruk av applikasjonen, men denne er ikke ferdig implementert.

Dersom brukeren havner i en nødsituasjon, er det mulighet for å sveipe SOS-knappen nederst på meny-siden som sender brukerens nåværende koordinater til en vaktcentral. Brukeren blir spurta om bekrefteelse før melding er sendt.

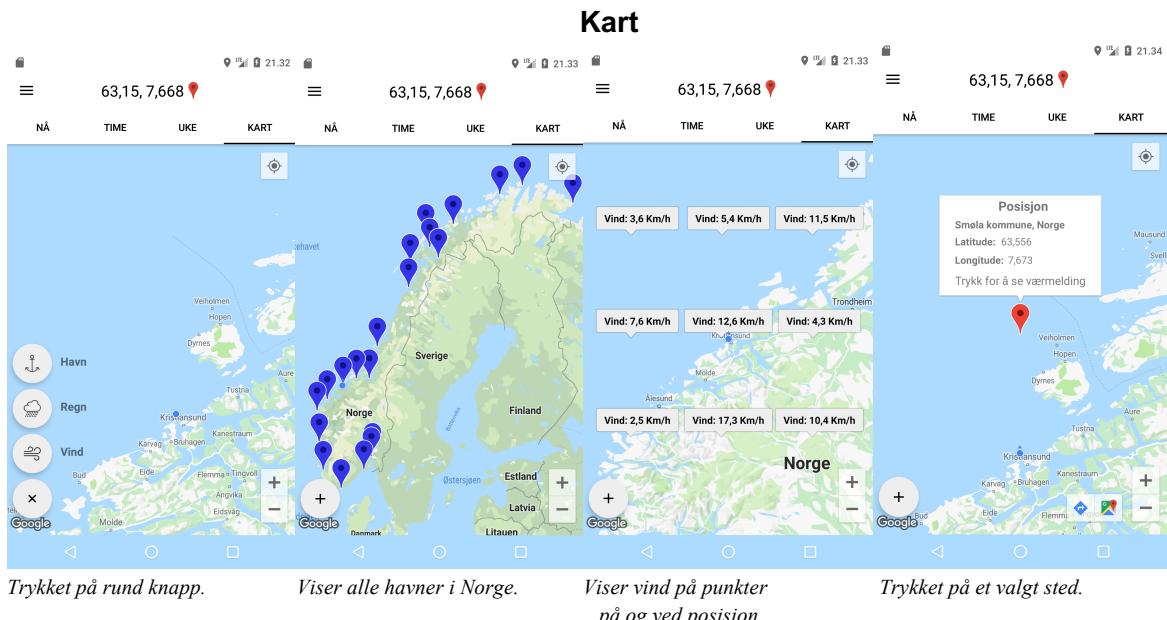
Krav som oppfylles fra product backlog: “filtrere informasjon”, “velge båttype”, “kontakte vaktcentral” og “velge måleenheter”.



Time-siden viser data 24 timer frem i tid. Det første som vises er vind og bølgehøyde. Til venstre finner bruker tidspunkt. Den øverste boksen er nåværende tidspunkt og bruker kan scrolle ned. Dersom bruker trykker på en av boksene kommer det frem et større vindu med tåke, temperatur, tidevann, nedbør, sikt og fuktighet for den valgte timen. Trykker bruker på samme boks forsvinner vinduet.

Uke-siden er bygd opp på samme måte som time-siden, men har ikke med et ekstra vindu. Den visualiserer kun vind og bølgehøyde i en liste, og gir brukeren en oversikt på 9 dager fremover i tid.

Krav som oppfylles fra product backlog: "se time-for-time varsel" og "se ukesvarsel".



På kart-siden er nåværende posisjon visualisert med en blå prikk på et kart. Det er mulig å zoome inn og ut av kartet med “+” og “-“ knappene. Om brukeren ønsker å gå tilbake til nåværende posisjon kan bruker trykke på lokasjons-knappen øverst til høyre i kartet. Nederst i venstre hjørnet er det en knapp som får opp knapper med valg av visualiseringer til bruker. Trykker brukeren på “Havn” vises alle havner i Norge på kartet. Trykker brukeren på “Vind” vises punkter på kartet med verdier for vind inkludert nåværende posisjon. Trykker brukeren på “Regn” skjer det samme som “Vind” bare med verdier for vind. Dersom bruker trykker på et sted på kartet og deretter på posisjons-ikonet, kommer posisjonen til det valgte stedet opp på et lite vindu. Trykker bruker på vinduet, går applikasjonen tilbake til nå-siden og viser vær-verdier for den valgte posisjonen.

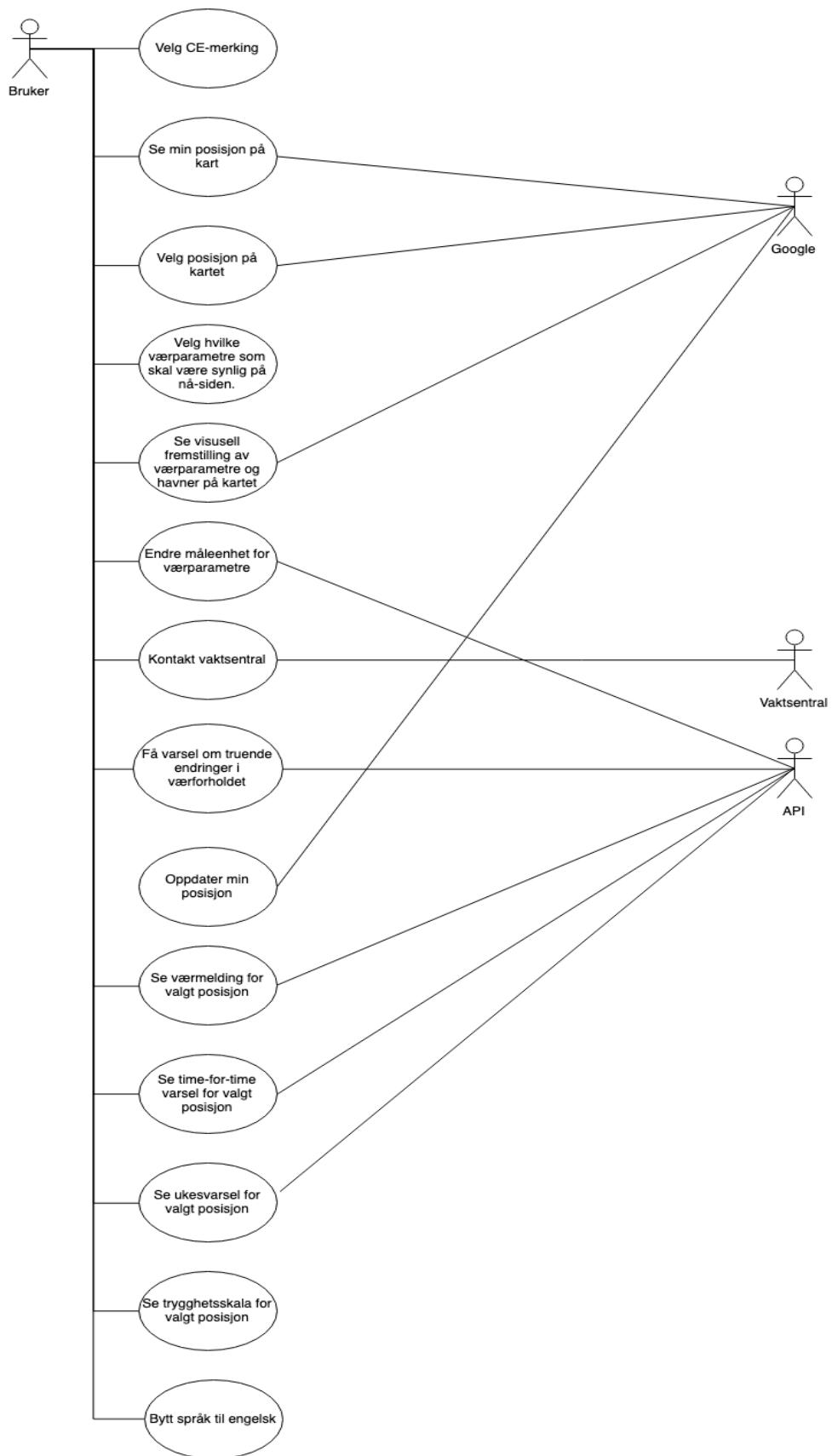
Krav som oppfylles fra product backlog: “finne nåværende posisjon på kart”, “visuell framstilling av regn og vind”, og “visuell oversikt over havner i Norge”.

Kravspesifikasjon og modellering

I denne delen av rapporten har vi inkludert ulike beskrivelser av kravspesifikasjonen vår. Vi benyttet oss av brukerhistorier for å formulere kravene i product backloggen. Disse er mindre detaljerte og passer dermed godt i smidig utvikling. I tillegg til disse lagde vi use case-diagram, tekstlig beskrivelse, klassediagram og sekvensdiagram. Flere typer modeller synliggjør ulike aspekter ved systemet, noe som gir en mer fullstendig representasjon av systemet.

Use case diagram

I starten av prosjektet formulerte vi kravene som use cases i et use case diagram(bilde 1). Underveis har vi oppdatert diagrammet med nye krav, og fjernet krav vi ikke lengre har med.



Bilde 1

Tekstlig beskrivelse

I forbindelse med use-case diagrammet lagde vi også tre tekstlige beskrivelser i starten av prosjektet. En tekstlig beskrivelse tar for seg interaksjonen mellom systemet og bruker og gjør use-caset ”komplett”.

Navn: Se min posisjon på kart

Aktører: Bruker, Google

Prebetingelse: Bruker har lastet ned applikasjonen og har tilgang til nett

Postbetingelse: Bruker ser min posisjon på kartet

Hovedflyt:

1. Bruker går inn på applikasjonen
2. Systemet ber om tillatelse til bruk av min posisjon
3. Bruker godtar bruk av min posisjon
4. Bruker trykker på kart
5. Systemet viser kart og peker på ”min posisjon”

Alternativ flyt:

- 3.1 Bruker godtar ikke bruk av min posisjon
- 3.2 Bruker trykker på kart
- 3.3 Systemet viser kart uten ”min posisjon”

Navn: Se værmelding for valgt posisjon

Aktører: Bruker, API

Prebetingelse: Bruker har lastet ned applikasjonen og har tilgang til nett. Brukeren er inne på applikasjonen og har godtatt bruk av posisjon

Postbetingelse: Bruker ser værmelding for valgt posisjon.

Hovedflyt:

1. Bruker går til ”kart”.
2. Systemet viser kart og peker på ”min posisjon”
3. Bruker trykker på ønsket posisjon på kartet
4. Systemet viser valgt posisjon på kartet.
5. Bruker går til nå-siden og sjekker værmelding

Alternativ flyt:

- 4.1 Systemet har ikke data for valgt posisjon
- 4.2. Går tilbake til steg 3

Navn: Velg båttype

Aktører: Bruker

Prebetingelse: Bruker har lastet ned applikasjonen og har tilgang til nett. Brukeren er inne på applikasjonen og har godtatt bruk av posisjon

Postbetingelse: Bruker har valgt ny båttype

Hovedflyt:

1. Bruker går inn på meny
2. Systemet viser meny-siden

3. Bruker trykker på “båttype”
4. Systemet viser valgmeny med de ulike båttypene
5. Bruker trykker på et nytt merke
6. Systemet oppdaterer trygghetsskalaen basert på ny båttype

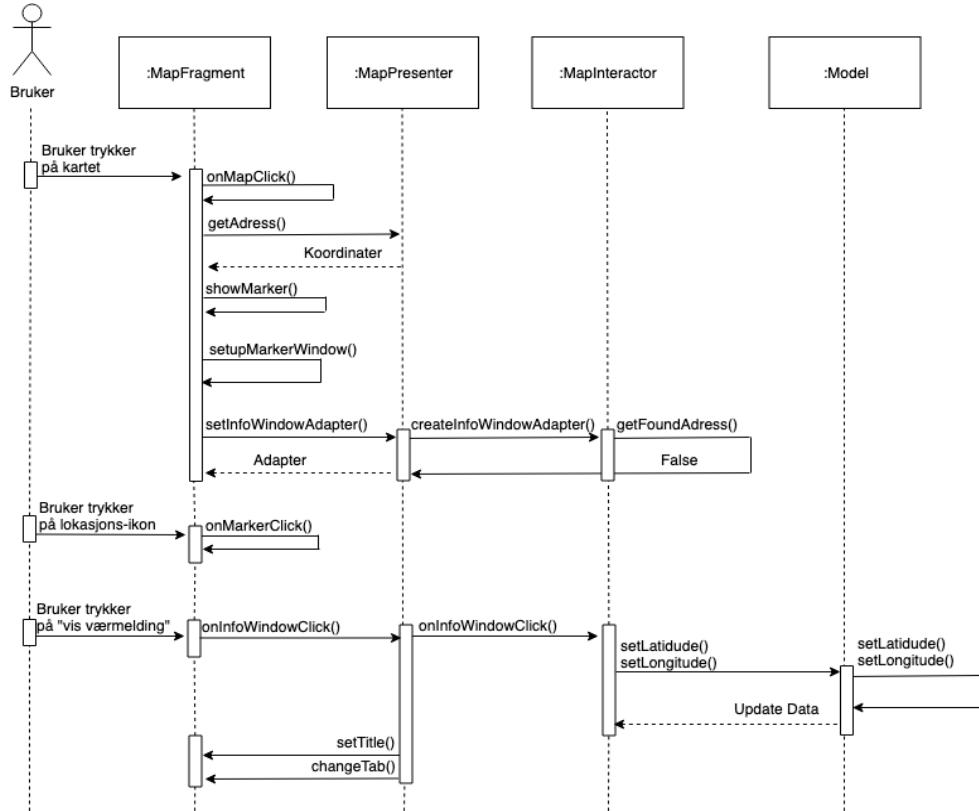
Alternativ flyt:

- 5.2 Bruker trykker “Avbryt”

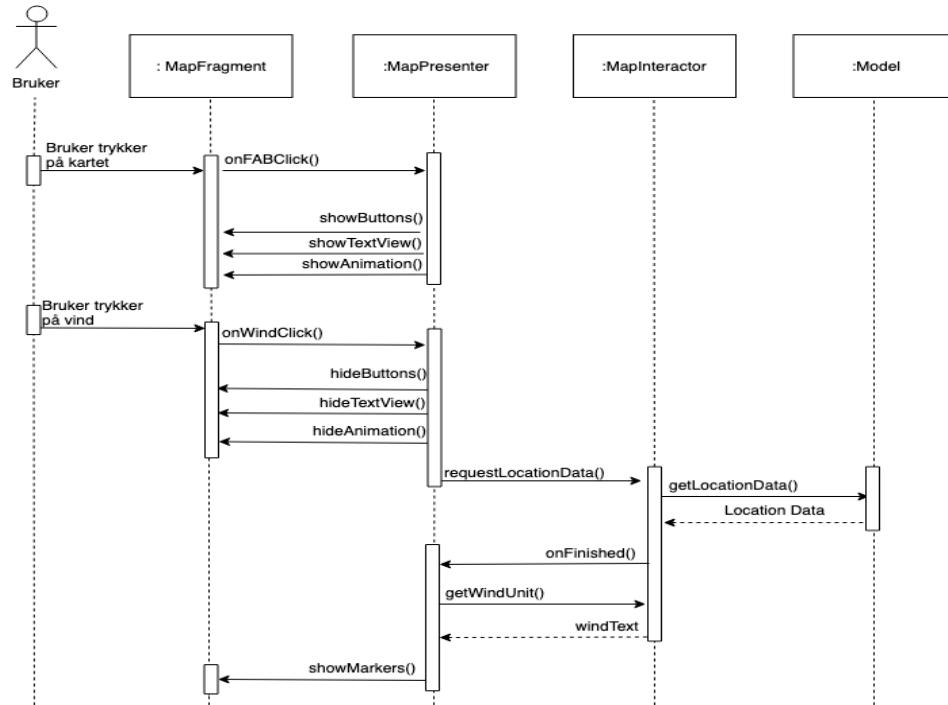
Sekvensdiagram

Vi lagde sekvensdiagram for tre av de viktigste use casene i diagrammet for å modellere flyten i use-casene. Disse har vi også oppdatert underveis i prosjektet.

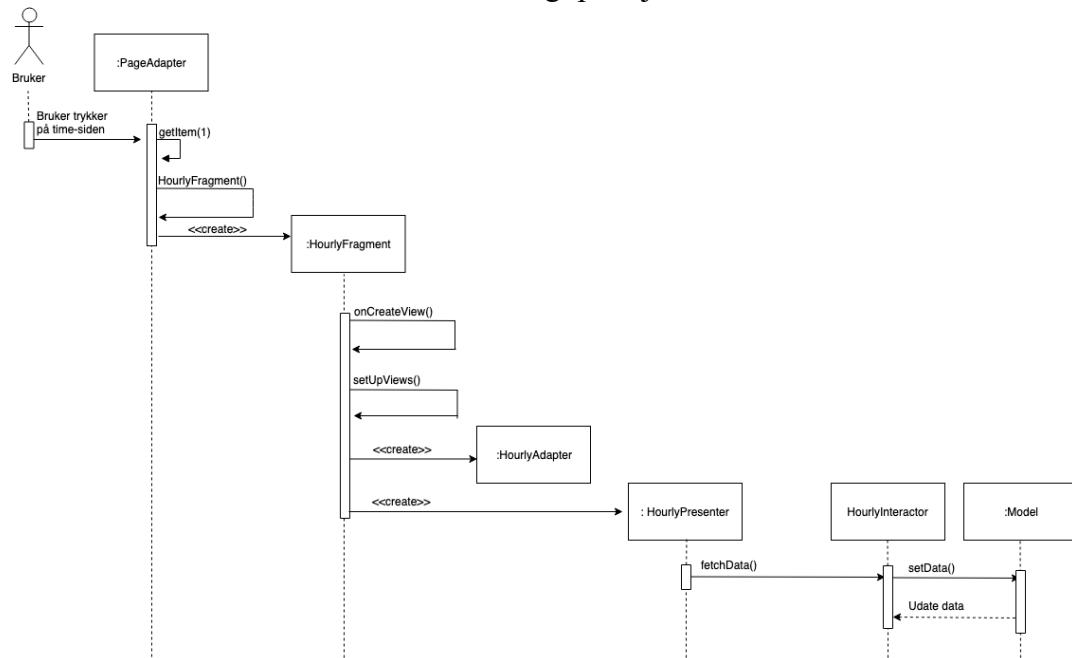
Use case: “Se værmedding for valgt posisjon”



Use case: "se visuell fremstilling av værparametre og havner på kartet"



Use case: "Se time-for-time varsel for valgt posisjon"

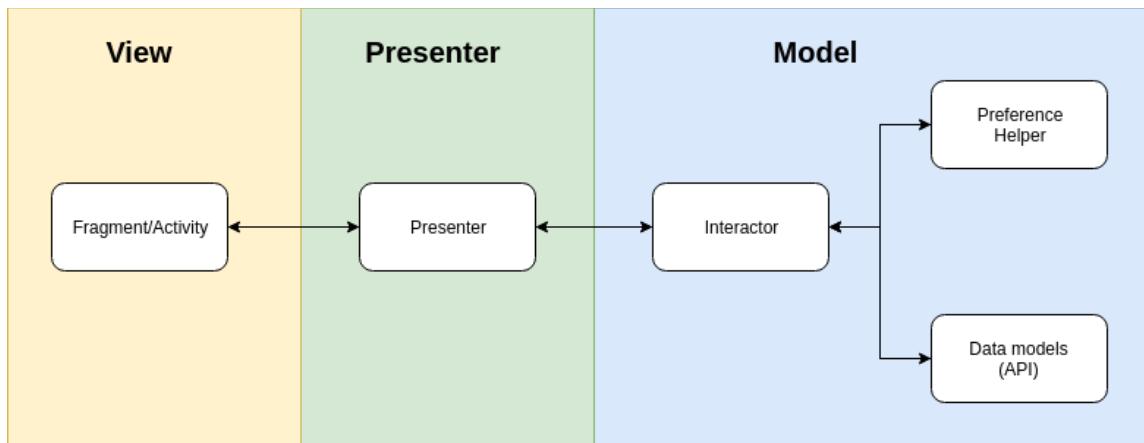


Klassediagram

I starten av prosjektet lagde vi en domenemodell basert på de funksjonelle kravene vi hadde fra starten. Denne modellen utviklet vi i løpet av prosjektet som har resultert i et klassediagram. Vi har brukt diagrammet for å få oversikt over objektenes tilstander/attributter og hvilke metoder de har. Dette ligger i vedlegg 1.

Design pattern

Design patternet vi har benyttet oss av heter Model View Presenter (MVP). MVP er en enkel måte å bygge og utvikle strukturen på koden i applikasjonen vår. Vi deler koden i fire forskjellige deler (bilde 2).



Bilde 2

Den første delen er Model. Denne delen vil inneholde dataen som vi henter eksternt (API kall) og lokalt (sharedpreference). Når vi gjør API kall og henter dataen blir den lagt inn i enkle klasser som de andre komponentene i MVP kan bruke. (Jinn Kim, 2018)

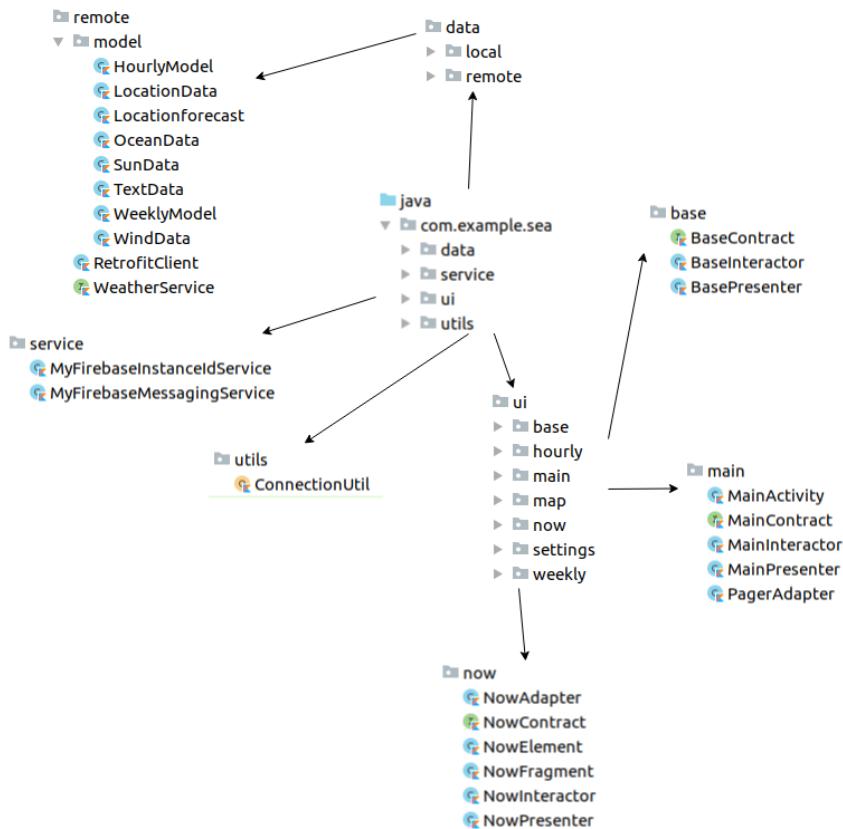
Andre delen er View. Her ligger dataene som vises til brukeren. Eksempler på ansvarsområder er å oppdatere views som recyclerview og textview, og lytte til eventhandlers. I Android vil View være en activity eller et fragment. (Rakshit Soral, 2018)

Tredje delen er Presenter. Her skal alt som har med logikk å gjøre ligge. Presenter håndterer grensesnitt oppdateringer basert på endringer i Model og behandler brukerens input. Presenter blir "mellommannen" mellom Model og View. Presenteren blir varslet om at dataen har endret seg fra Interactoren, og da vil Presenteren sende dataen til View. I vårt tilfelle vil det være en Fragment som vil oppdatere viewene i Fragmentet med dataen fra Presenteren. Dette vil også funke ved å gå andre veien. Når en bruker trykker på en av måleenhetene i navigation draweren så lytter View. Da vil en metode i Presenteren bli kalt, hvor Presenteren håndterer logikken og kaller på metoder i Viewet. Viewene oppdateres med dataen fra Model som den får fra Presenteren. (Rakshit Soral, 2018)

Fjerde delen er Interactor. Denne delen er valgfri å ha med i MVP. Interactor holder kontroll over alle API kall og håndterer data. Det er et mellomledd mellom Presenter og Model. Når applikasjonen åpnes vil View kalle på en metode i Presenter som igjen vil kalle på Interactor. Interactor vil utføre API kallet og hente data fra for eksempel OceanForecast. Dataen fra API-kallet vil bli parset og lagt inn i enkle klasser. Deretter vil Interactor videreføre dataen til Presenteren, som vil håndtere dataen og utføre logikken. Dataen videreføres til View som oppdaterer viewene. (Antonio Leiva, 2019)

Ved å dele applikasjonen i flere lag, blir det klare ansvarsfordelinger mellom de ulike komponentene. Dette fører til enklere forståelse og vedlikehold av kodebase. Med veldefinerte grenser mellom komponentene blir det mye enklere å teste hver komponent og koden blir mer ryddig.

Under har vi lagt til en forenklet representasjon av mappestrukturen(bilde 3). For eksempel vises ikke alle filene i time-siden, uke-siden og alle filene i local mappen med lokal data.



Bilde 3

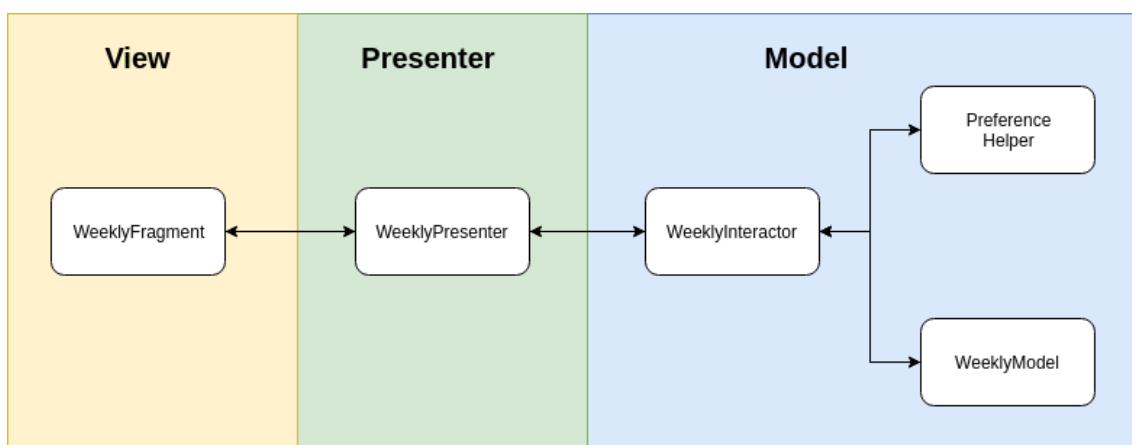
Objektorienterte prinsipper

Kohesjon er et mål på hva slags ansvar et objekt har og hvor fokusert ansvaret er. Et objekt som har moderat ansvar og utfører et begrenset antall oppgaver innenfor et funksjonelt område har høy kohesjon. (Y. Lindsjørn, 2018).

Vi har prøvd å ivareta dette ved bruk av MVP og objektorienterte prinsipper som arv og polymorfisme. Som nevnt har View bare ansvar for brukergrensesnittet. Hver komponent har også et grensesnitt som det implementerer. Vi valgte også å ha en Interactor slik at vi oppnår enda høyere kohesjon. Da vil Presenter bare ha ansvar for og utfører bare logikk. Vi begrenser antall oppgaver Presenter skal ha, slik at det utfører arbeid innenfor bare ett funksjonelt område. Presenteren har da ansvar over logikken og Interactoren har ansvar for håndtering av data og kommunikasjon med Model. Interactoren gjør API kall og får data fra datamodellene og sharedpreference.

Kobling er et mål på hvor sterkt et objekt er knyttet til andre objekter. Lav kobling vil si at et objekt samarbeider med et begrenset antall andre objekter. Et objekt med sterk kobling er avhengig av mange andre objekter, noe som kan gjøre endring vanskelig, og testing vanskelig. (Y. Lindsjørn, 2018).

Igjen ved bruk av MVP så har hvert objekt eller komponent lav kobling. Objektene er ikke avhengig av mange andre objekter, dette fører til at det er både lettere å gjøre endringer og teste objektene. I bildet under (bilde 4) vises hvordan koblingen mellom komponentene ser ut. View har for eksempel kobling til Presenteren, men Presenteren har en kobling mellom View og Interactoren. Bildet viser hvordan vi har ivaretatt det med kobling i uke-siden, men vi har fulgt samme strukturen gjennom hele applikasjonen.



Bilde 4

I starten av prosjektet brukte vi ingen design patterns, men jo mer vi begynte å programmere og jo større applikasjonene og kodebasen ble, desto vanskeligere ble det å holde styr over kodebasen og teste. Vi bestemte oss derfor for å sette i gang med å bruke MVP.

Produktdokumentasjon

Løsningen fra et teknisk ståsted

Første gang man starter applikasjonen får man opp to permission dialoger hvor den ene spør man applikasjonen kan sende en melding for deg, den andre spør om applikasjonen kan bruke posisjonen din. Deretter kommer det opp en Alert Dialog hvor bruker må velge en båttype. Om bruker ikke lar oss bruke posisjon vil applikasjonen bare gi data fra koordinatene 60, 11.

Applikasjonen vår er bygget opp av fire fragments og en “navigation menu”. I “navigation menu” har vi fire inndelinger; båttype, måleenheter, værpreferanser og innstillinger. Både måleenhetene, valgt båttype og valgte værpreferanser blir lagret i en SharedPreference.

Om bruker klikker på værpreferanser får bruker opp en AlertDialog hvor han kan velge hvilke værparametere han vil se i nå fragmentet. Tidevann vises kun om valgt posisjon er under 30 km til en havn, om avstand er mer enn 30 km viser den ingenting. Hvilke bokser bruker velger å se blir også lagret i en SharedPreferences. Første gang bruker åpner applikasjonen vil ingen av værparametere være valgt.

Innstillinger åpner en ny Activity. Under nettverk kan bruker velge radiusen rundt posisjonen bruker i kartet. Bruker kan velge mellom lav, medium og høy. Om den er høy bruker den en radius på 4° nord og sør, og 8° øst og vest. Som medium er det 3° nord og sør, og 6° øst og vest. For lav er det 2° nord og sør, og 4° øst og vest. Om du skal endre på tillatelsene om posisjon og telefon vil applikasjonen åpne en ny activity som heter "app info" som er innebygd i applikasjonen til android. Her kan bruker stoppe applikasjonen, avinstallerer, se hvor mye plass applikasjonen tar og hvor mye data applikasjonen bruker. Ved å klikke på tillatelser skrur man av tillatelsene som er blitt godtatt. Dette skjer kun om bruker har godtatt at vi bruker posisjonen eller telefonen. Om applikasjonen ikke kan bruke telefonen eller posisjonen vil den åpne en permissiondialog hvor han kan gi applikasjonen tillatelse til at vi kan bruke posisjon eller telefonen. Man kan skru av muligheten om å spørre om tillatelse ved å huke av "ikke spør igjen". Da vil den alltid være av. Dersom man trykker på "Tilbakemelding" åpnes en Mail klient.

Helt nederst i navigation menu har vi en Sidebar, hvor det står SOS. Om vi har tillatelse til å sende meldinger og applikasjonen har tilgang til din posisjon vil den åpne en Alert Dialog hvor bruker har to valg. Ved å klikke på ok vil den sende en melding til en av oss i teamet. Om man ikke har tilgang til enten telefonen eller posisjonen eller begge, vil applikasjonen åpne en annen Alert Dialog, ved å klikke ok her vil applikasjonen åpne meldings applikasjonen og man trykker send selv. Klikker man avbryt gjør applikasjonen ingenting.

I nå-fragmentet ligger det et recyclerview og en seekbar. Den dataen som ligger i nå-fragmentet blir hentet fra APIene OceanForecast, LocationForecast og TidalWater. De forskjellige dataene blir lagt inn i egne card views som legges inn i.recyclerviewet. Seekbaren er en skala fra 0 til 100. Seekbaren vil alltid være 0 om bølgehøyde er 0 fordi da er valgt posisjon på land. Utenom det blir maksverdien til seekbaren bestemt ut i fra hvilken båttype brukeren har valgt. Deretter setter vi inn den største av disse to verdiene inn i seekbaren.

I time-fragmentet ligger det et recyclerview og i hvert element i recyclerviewet ligger det et dropdown layout. APIene som blir brukt i time-fragmentet er de samme som blir brukt i nå-fragmentet. Ved å klikke på et element i recyclerviewet aktivere man drop down funksjonen som da åpner et dropdown layout.

Når man klikker på uke-fragmentet får man opp et recyclerview med ni elementer hvor hvert element gir data om vind og bølgehøyde. APIet for uke-fragmentet er LocationForecast og OceanForecast. Her velges data for kl.12.

Map-fragmentet åpner Google Maps. Nederst til venstre har vi lagt til en Floating Action Button (FAB). Ved å klikke på den får man opp tre nye FAB. Ved trykk på FAB for regn og vind settes markers med tilhørende verdi for det området den peker på. Ved å klikke på kartet får man opp en marker. Ved å trykke på markeren får man opp et CustomInfoWindow med koordinater og et valg om å se varsling for det området. Dette fragmentet bruker APIet LocationForecast. De koordinatene APIene henter data fra står alltid øverst i headeren, med unntak av når man bruker SOS-knappen.

Teknologier og arkitektur

I applikasjonen benytter vi oss av ni forskjellige teknologier. Disse er Firebase, Android Studio, GitHub, Git, Retrofit, RxJava, GSON, SimpleXML og ScalarsConverter. Vi brukte Firebase for å implementere et pushvarslingsystem ved å bruke Cloud Messaging. Dette kan vi benytte oss av til videre utvikling for å gi beskjed til brukerne om vi gjør endringer på applikasjonen eller om det er en oppdatering klar. Android Studio er IDEen vi koder i og kodespråket vi benytter oss av er Kotlin. Vi lagrer applikasjonen sin utvikling i GitHub og committer og pusher endringer til GitHub eller å hente den nyeste versjonen fra GitHub. For å gjøre trådprogrammeringen letter bruker vi RxJava som også effektiviserer trådene våre. Gson parser data i json format fra apiet, Retrofit mapper det inn i en klasse. SimpleXML gjør det samme som GSON, men det parser på XML data. ScalarsConverter bruker vi for å mappe primitive typer. Retrofit er en type sikker REST klient for Android som regnes bransjestandard. Biblioteket har god støtte for REST API, som MET sitt API. Det er også lett å teste, sette opp og gjøre nettverk kall. Vi bruker Retrofit for godkjenning og samhandling med MET sitt API, og sendinger av nettverkforespørsler med OkHttp. Retrofit er bygget opp på OkHttp. Siden MET API’er bruker ulike dataformat må vi bruke ulike parsere. Ved å hente dataen gjennom web API’et blir dataen parset i en klasse (POJO).

Arkitekturen til applikasjonen er bygget opp på Model View Presenter som ble forklart i design pattern, hvor View er frontend arkitekturen og backend arkitekturen er Presenter, Interactor og Model.

Testdokumentasjon

Testing for hver sprint

Vi hadde en standard for hvordan vi skulle gjennomføre testing under hver sprint. Først oppretter man en implementasjons-branch. På denne branchen skal man gjennomføre statisk testing og enhetstesting. Dersom alt fungerer som det skal, commiter man dette. Så lager man en test-branch. Her gjennomfører man en “merge” mellom implementasjons-branchen og test-branchen. På test branchen skal man gjennomføre integrasjonstesting. Er det noe som er galt, endres dette i den opprinnelige implementasjons-branchen og man lager ny test-branch. Hvis alt funker fint, går man tilbake til implementasjons-branchen og pusher. Etter å ha pushet lager man en pull-request i GitHub. Denne pull-requesten skal en annen person ta seg av og håndterer eventuelle konflikter med master, før det til slutt bli merget til master.

Statisk testing

Statisk testing er første steg i testeprosessen. Statisk testing er testing som gjøres uten at man kjører koden. Her analyseres kildekoden med mål om å forbedre kvaliteten på koden. Typiske feil som vi har avdekket i slike tester er variabler som ikke blir brukt, unødvendig kode og brudd med organisasjons kravene. For eksempel bruk av snake_case i stedet for camelCase. Android Studio sine hjelpefunksjoner har vært veldig nyttige i denne prosessen. For eksempel vil Android Studio gi en warning og farge kodeblokken grått når man har kode som ikke blir brukt. Android studio sier også i fra om man har unødvendige kodeblokker. Et eksempel er om man har en if-sjekk som alltid er sann.

Enhetstesting

Enhetstesting går ut på at man tester individuelle enheter i et system eller komponenter av en applikasjon eller programvare. Dette gjøres for å sjekke at alle komponentene fungerer som det skal og at det ikke finnes noen feil. (Guru99, 2019). Vi har tatt i bruk automatisk enhetstesting. For å gjøre dette isolerte vi komponenter av koden, opprettet egne kodesnutter for å teste disse og sjekke om resultatet ble riktig. Variablene på bildet (bilde 5) som er markert med merknad “Mock” er “mock objects”. Dette er test objekter som man får fra Mockito biblioteket. Merknad “Test” (bilde 6) er det som lar seg gjøre at metoden blir kjørt individuelt. Test-metodene under har tre ulike API kall hvor vi prøver å hente data fra “met-api”. Her sjekker vi at dataen ikke er null. Dette er en veldig simpel test vi gjorde i starten av utviklingen av henting av data og bruk av Retrofit.

```
@Mock
var latitude: Double = 50.0

@Mock
val longitude: Double = 20.0

@Mock
private val harbors : HashMap<String, LatLng> = hashMapOf<String, LatLng>()

@Mock
private var closestHarbor : String? = null

@Mock
private var closestHarborValue : Double = Double.MAX_VALUE
```

Bilde 5

```
@Test
fun getTidalData() {
    val call : Call<String> = RetrofitClient().getClient( formatString: "string").getTidalWater(closestHarbor)

    assertEquals(tidalResponse.code(), actual: 200)
    assertEquals( expected: true, tidalResponse.isSuccessful)
    assertNotEquals(call, actual: null)
}
```

Bilde 6

Ved testing av user interface (UI) har vi hovedsakelig brukt manuell testing. Hver person tester en funksjon, deretter vil en annen teste funksjonen og eventuelt gi tilbakemelding. Vi benyttet Espresso i starten men i og med at det krever mye ytelse valgte vi bort dette etterhvert.

Ved testing av API brukte vi enhets- og integrasjonstesting hvor vi kombinerte dette med logging metoder. Vi har antatt at oppsettet av API og HTTP kommunikasjonen fungerer problemfritt, ettersom dette ikke er vårt ansvar. Vi tester om POJO klassene våre er annotert riktig, slik at SimpleXML kan de-serialisere dataen i XML-format fra APIene. I dette tilfellet så “mocket” vi server-responsen ved bruk av testing rammeverket MockWebServer. Vi har også testet om vi håndterte alle HTTP statuskoder riktig. Her måtte vi også “mocke” MET sitt API fordi det ikke var lett å simulere at MET API responderte med en spesifikk statuskode. Vi tok i bruk en HTTP logging interceptor (bilde 7) slik at vi kunne observere dataen som blir overført over nettverket, og annen viktig informasjon som headere.

```
val logging : HttpLoggingInterceptor = HttpLoggingInterceptor().setLevel(HttpLoggingInterceptor.Level.BODY)
val okHttpClient : OkHttpClient = OkHttpClient.Builder()
    .addInterceptor(logging)
    .build()
```

Bilde 7

Integrasjonstesting

Målet vårt med å gjennomføre integrasjonstesting er å finne feil i grensesnitt og i samspillet mellom integrerte komponenter i systemet. Selv om alle disse komponentene er enhetstestet betyr ikke det at de vil fungere feilfritt som en helhet. For å sjekke dette tester vi de individuelle komponentene i systemet som en helhet fra bunnen og opp. (Guru99, 2019)

I integrasjonstestinga har vi brukt to ulike metoder, big bang og bottom up. Bottom up er inkrementell, som vil si at man tester på laveste nivå av systemet og jobber seg oppover. Deretter foretar vi big bang hvor vi tester hele systemet på en gang og sjekker at det også fungerer. Bottom up er en fordel fordi det er lettere å lokalisere feil og diverse problemer i de individuelle komponentene. Et eksempel på hvordan vi har gjort dette er: Test Case Objective: Sjekke at når bruker endrer koordinater blir hele systemet oppdatert til de nye koordinatene. Test Case Description: Gå inn på kart-siden, trykke hvor som helst på kartet for å få opp posisjon ikonet. Trykk på ikonet for å få mer info i en alert-boks. Trykk på alert boksen for å se værmelding for disse koordinatene. Expected result: Hele systemet blir oppdatert til disse koordinatene. Header vil vise de nye koordinatene. Bruker blir omdirigert til nå-siden og man vil se den nye værmelding for valgte koordinater. (Guru99 Integrasonstesting)

Testing av ikke-funksjonelle krav

De ikke-funksjonelle kravene for systemet vårt er delt inn i tre typer krav: produktkrav, organisasjonskrav og eksterne krav. Kravene omhandler begrensninger på system-nivå. De er viktige fordi de påvirker design og testing av de fleste eller alle historiene i product

backlogen (Essential Scrum, 2012, s.138). Under har vi listet opp de ikke-funksjonelle kravene for systemet, og deretter hvordan vi har testet disse.

Produktkrav

1. Systemet skal ta under tre sekunder å laste inn hele applikasjonen.
2. Systemet bør la brukeren finne værmelding for et valgt sted på kartet på under 1 minutt.
3. Systemet bør ikke ha behov for opplæring for å bruke applikasjonen.
4. Systemet skal kunne tilby engelsk og norsk språk.

Organisasjonskrav

1. Systemet skal ha en dokumentasjon av koden slik at den er lett forståelig for utvikleren og slik at den kan vedlikeholdes uten vanskeligheter.
2. Systemet skal ha kommentarer på norsk
3. Systemet skal være kodet på engelsk.
4. Systemet skal utvikles ved bruk av Kotlin.
5. Systemet sin kodestil skal følge CamelCase
6. Systemet skal følge design patternet Model View Presenter
7. Systemet skal følge Materialdesign guidelines

Eksterne krav

1. Systemet skal følge WCAG 2-0-standarden for universell utforming.

Produktkrav 1. dreier seg om ytelse og kan testes ved å direkte måle tiden det tar å laste inn hele applikasjonen. Dette har vi testet manuelt både på emulator og på mobil. Tiden det tar for en emulator å laste applikasjonen er varierende fra gang til gang og avhenger også av hvilken datamaskin som brukes. Emulatoren bruker også tid på å laste inn selve enheten, derfor testet vi kun på mobil. Her forsøkte vi å teste med ulik belastning (nettverk, antall åpnede apper), resultatet var en gjennomsnittstid på 1,3 sekunder.

Produktkrav 2. og 3. handler om brukervennlighet og hvorvidt systemet er lett å bruke. Disse kan variere for ulike brukergrupper, men vi har fokusert på målgruppen vår. Krav 2. testet vi ved å direkte ta tiden det tar å finne værmelding for et valgt sted på kartet da vi gjennomførte akseptansetesting med brukere. Krav 3. testet vi også under akseptansetesten ved å notere ned antall feil brukeren gjorde. Resultatene fra begge disse testene er dokumentert under "akseptansetest".

Organisasjonskravene og de eksterne kravene er direkte målbare og kan besvares med ja/nei spørsmål.

Testing med brukere

Brukertest

Vi gjennomførte en brukertest under sprint 4 der målet var å teste ulike ikoner vi ønsket å benytte i applikasjonen vår. Vi inkluderte de vi allerede hadde tatt i bruk og andre alternativer. Vi ønsket å finne ut hvilke av ikonene som var mest forståelige.

Vi lagde en plan for datainnsamlingen (vedlegg 4) der vi hadde bilde av alle ikonene, med de som skulle representere det samme ved siden av hverandre. For hvert ikon spurte vi deltaker om de forsto hva ikonet skulle representere. Om de ikke gjorde det, viste vi et tilfeldig mål på det ikonet skulle representere. Deretter spurte vi hvilket ikon de syns representerte parameteren best. Om ikonet var vanskelig å forstå spurte vi om de hadde noen innspill til forbedringer.

Vi gjennomførte testen med tilfeldige brukere. Grunnen til at vi ikke valgte å kun teste med folk fra målgruppen var fordi vi syns at vær-ikoner er noe som de aller fleste kjenner til fra før. Vi ønsker at applikasjonen skal kunne brukes av de som også bruker båt som fritidssyssel, og derfor skal det ikke ligge noe kunnskapskrav til grunn.

Vi gjennomførte 8 individuelle tester. Vi stoppet opp når vi følte at vi ikke ville lære noe nytt. For de fleste ikonene var det åpenbart hvilket ikon som var mest intuitivt. De ikonene det var størst problemer med var tidevann, trykk og tåke. For tidevann var det forvirrende at pilen kun gikk opp, i og med at tidevann er noe som varierer i høyde. En kommentar vi merket oss var at tidevann vanligvis merkes når det er lavvann, mer enn høyvann, og derfor kan vi kanskje heller ha en pil ned. Trykk var vanskelig fordi det ikke er så mange som har noe kunnskap om det. Når det gjaldt tåke ble den forvekslet med både skyer og vind. Her var det vanskelig å finne et godt alternativ, men det var likevel en av forslagene som fikk flest "stemmer".

Akseptansetest

Etter å ha gjennomført enhetstesting og integrasjonstesting for den ferdige løsningen var siste steg å gjøre en brukertest. En brukerakseptansetest har som mål å sjekke om målgruppens behov blir møtt med den ferdige løsningen. Dette gjøres ved å teste kravspesifikasjonen. Planleggingen gikk ut på å lage oppgaver for hver brukerhistorie i product backloggen. Dette dokumentet har vi lagt ved som et vedlegg (vedlegg 5). Vi har valgt å ikke inkludere kravet som omhandler oppdatering av posisjon, siden vi ikke har mulighet til å teste applikasjonen i naturlige omgivelser. Samme gjelder kravet som omhandler oversetting fra norsk til engelsk, siden dette gjøres via innstillinger på telefonen og ikke applikasjonen.

Vi gjennomførte en pilottest for å eliminere eventuelle feil ved testen. Når vi ba bruker om å sjekke bølgehøyde var de på land, dette var ikke mulig i og med at bølgehøyde ikke vises på land. Samme problemet gjaldt for en annen oppgave, men her erstattet vi bølge med vind. En

annen ting vi merket oss var at vi brukte synonymer til parametrene i testen, i stedet for akkurat det værparameteren heter i applikasjonen, noe som førte til forvirring.

Vi gjennomførte 3 brukertester. For hver test målte vi to metrikker, tid og feil. Der tid er tid brukt på oppgavene for hvert krav og feil er når bruker trykker feil sted.

Oppgavene deltakerne brukte mest tid på, og hadde flest antall feil, var de som tilhørte krav 3. Her var det en oppgave som gikk ut på å se hva temperaturen er om to timer. For å få opp temperatur må man få ned dropdown layout. Det var ikke så intuitivt at denne ble tilgjengelig når man trykket på en av boksene. Her ville det kanskje vært mulig å legge til en tekst eller et tegn som viser at man kan trykke på boksen.

En annen ting vi merket oss var at en av deltakerne hadde langvarige trykk på kartet når han skulle endre posisjon. Han forklarte at grunnen til dette var fordi det er slik man må gjøre det i Google Maps. Det var kun en av deltakerne som gjorde dette, derfor mener vi at det må flere brukertester til for å se om det er noe vi burde endre.

Universell utforming

Gjennom utviklingsprosessen av applikasjonen vår har vi forsøkt å ha universell utforming i bakhodet. Retningslinjene WCAG 2.0 har kriterier som gjelder for mobile grensesnitt. Ikke alle disse lar seg gjøre å følge i dette kurset. For eksempel skal mobile-løsninger helst fungere på et utvalg mobile plattformer, men i dette kurset har vi kun utviklet for android plattform.

Når det gjelder utforming burde bruker kunne komme raskt til innholdet for en bedre brukeropplevelse. (<https://material.io/>). For å oppnå dette har vi forsøkt å presentere konkret innhold i hovedsidene og lagt inn annen funksjonalitet i sidemenyen. Der hvor scrollbart grensesnitt er benyttet sørger vi for at det viktigste blir presentert først. Et eksempel på dette er «nå» siden hvor vi har festet trygghetsskalaen øverst.

Videre skal klikkbare objekter ha stor klikkflate og bør dekke minst en tredjedel av skjermbredden, høyden bør tilsvare området av fingertuppen (<https://material.io/>). Disse prinsippene har vi for eksempel fulgt i menyen vår hvor hvert av de klikkbare valgene dekker én tredjedel av skjermen, og på «nå» siden hvor hele overflaten av objektene klikkbare for å få ned dropdown.

Det er viktig med gjenbruk av eksisterende konsepter og ikoner som er gjenkjennbare. Vi har forsøkt å velge ikoner som er mye brukt i andre vær-apper. Basert på resultatene fra brukertesten valgte vi de ikonene som var mest forståelige. Ulike fingerbevegelser er også noe som er viktig at fungerer slik en bruker forventer at det skal fungere. De mest kjente er å bruke én finger for å scrollle eller bla, og to fingre til å zoome inn og ut. For eksempel kan bruker scrollle nedover «nå» siden og swipe mellom de ulike sidene. Zoome-funksjonen er innebygd i Google Maps.

Det skal være enkelt for brukere å navigere i applikasjonen og finne frem. For å gjøre det tydelig for brukeren hvilken side de er på har vi markert dette i menyen øverst. Dersom brukeren er på siden med kartet vil «kart» være utevært i menyen. Vi har også sørget for at layouten er konsist. Dette innebærer blant annet at de samme ikonene, fargene og design blir brukt på alle steder i applikasjonen som representerer det samme. Knappen for å komme til sidemenyen er også plassert på samme sted uansett hvor i applikasjonen man er.

Sikkerhet

Sikker smidig programvareutvikling krever at man legger inn aktiviteter for sikkerhet ved oppstart i prosjektet, inn i hver sprint og ved avsluttende evaluering av systemet. Disse aktivitetene innebærer innhenting av generelle sikkerhetskrav, identifisere trusler som må forhindres og evaluering av system og sikkerhet. De generelle sikkerhetskravene kan formuleres på samme måte som en vanlig brukerhistorie, men fra en angripers perspektiv. (Jøsang, 2019, s.7-10).

Trusselbilde for vår app er begrenset. Vi har valgt og ikke ta i bruk noen form for brukerkonto for vår app, dermed lagrer vi heller ingen form for personopplysninger. I applikasjonen ber vi om samtykke til bruk av “min posisjon”. Innhenting av “min posisjon” er innebygd i Google sitt system, og det er dermed ikke noe vi kan håndtere selv. Et eksempel på et mulig misusecase for dette problemområde er:

- Som angriper ønsker jeg å hacke “Bølgen” for å få tak i posisjonen til brukere av applikasjonen slik at jeg kan stalke dem.

I applikasjonen ber vi også om samtykke til å sende SMS fra brukerens telefon. Personen vi har satt til å få denne meldingen, vil derfor få tilgang til brukerens telefonnummer. I realiteten vil det være vaksentralen som får tilgang til dette nummeret. Derfor vil lagring av dette telefonnummeret være ansvaret til vaksentralen.

Noe annet angriper kan utnytte er våre Google-kontoer. Ved å få tilgang til en av kontoene har de tilgang til Firebasen vår. Dette kan en angriper bruke for å utgi seg som en annen og dermed utnytte dette til sin fordel (spoofing).

- Som angriper ønsker jeg å få tilgang til firebasen gjennom google-kontoen, slik at jeg kan sende en link med virus.

Hvis vi hadde hadde hatt litt mer tid til å videreutvikle applikasjonen ville vi nok satt av mer tid til sikkerhet slik det er diktert over. Det har ikke vært et hovedfokus, spesielt fordi vi ikke tar vare på noe form for sensitiv informasjon slik applikasjonen er nå. En eventuell videreutvikling ville kanskje hatt behov for mer personlige opplysninger og dermed mer fokus på sikkerhet.

Prosessdokumentasjon

Metode og utviklingsmetodikk

Rammeverket vi har benyttet oss av i dette prosjektet for organisering og administrering er Scrum. Scrum er basert på et sett med verdier, prinsipper og praksiser som fungerer som et grunnlag for hvert individuelle prosjekt. (Essential Scrum, 2012, s.13).

Scrum utviklingsarbeidet vårt består av et scrum team, der vi har tre scrum roller: product owner, ScrumMaster og utviklingsteamet. I og med at vårt team kun består av seks personer er product owner og ScrumMaster en del av utviklingsteamet. Product owner er ansvarlig for hva som vil bli utviklet i hvilken rekkefølge. ScrumMaster er ansvarlig for å guide teamet i utvikling og etterfølge teamets prosess basert på det større Scrum rammeverket. Vår ScrumMaster hadde også ansvar for å gi beskjed om hvilket grupperom og hvilket tidspunkt vi skulle møtes dagen etter. Utviklingsteamet er ansvarlig for hvordan man skal levere produktet som product owner har spurt om. (Essential Scrum, 2012, s.14-15). Hvem som har ScrumMaster rollen er noe vi har skiftet på i løpet av prosjektet, mens product owner har vært en fast person. I tillegg til scrum rollene har vi også hatt en sekretær for hvert møte. Sekretæren fikk ansvar for å hjelpe product owner med å fylle ut sprint backlog under møtet.

Videre vil det i Scrum være en rekke med aktiviteter; sprint, sprint planning, daglig scrum, sprint execution, sprint review, sprint retrospektiv og product backlog grooming. (Essential Scrum, 2012, s.17). I vårt prosjekt har vi lagt hver sprint til å vare én arbeidsuke, fra mandag til fredag. Før planleggingsmøter hadde vi først grooming der teamet hjalp product owner med prioritering av backloggen. I følge Kenneth S. Rubin skal 10% av tiden i sprinten skal settes av til grooming, for oss tilsvarer dette to av tjue timer i uken. (Essential Scrum, 2012, s.106). Dette var for mye vårt prosjekt og det var nok med ca. et kvarter før hvert planleggingsmøte. Resten av planleggingen av sprint backlog tok vanligvis én time. For dette møtet og de andre møtene satt vi gjerne av mer tid slik at vi kunne jobbe med oppgaver i fellesskap.

For å ha oversikt over product backlog og sprint backlog lagde vi tabeller i Google Drive. For product backlog lagde vi en tabell der vi la inn brukerhistorier i prioritert rekkefølge. Her inkluderte vi også tre kolonner med "sprint backlog", "in progress" og "completed". Vi strukturerte det slik for å ha oversikt over hver av oppgavene tilhørende hver brukerhistorie. For hver sprint lagde vi en tabell for ukens sprint backlog. Her la vi til de brukerhistoriene vi skulle gjøre i løpet av sprinten. For hver brukerhistorie la vi til detaljerte beskrivelser av oppgavene som måtte gjøres for at denne brukerhistorien skulle bli gjennomført. For hver av disse oppgavene la vi til to kolonner; en der vi skrev hvilke(t) medlem(er) av teamet som skulle gjøre denne oppgaven, og en der vi estimerte tidsbruk for denne oppgaven. På hvert møte hadde vi disse tabellene på storskjerm slik at vi lett kunne kommunisere ved bruk av tabellene.

Beskrivelse av oppstartsprosessen

I det første møtet for prosjektet var det nødvendigvis en del administrative saker vi måtte ta stilling til. Vi ble enige om å ha rullerende ScrumMaster for hver uke, slik at alle skulle få prøve seg. Vi lagde en prosjektplan (bilde 8) slik at vi hadde en oversikt over hvem som skulle være ScrumMaster når, og for å få en oversikt over tidsaspektet og prosjektets faser.

UKE	1	2	3	4	5	6	7	8	9	10	11	12	13
									(Påske)				
	Idemyldring, product backlog												
		Intervju, design, product backlog											
			sprint - iterasjoner (1 uke) sprint planning, daglig scrum, sprint review og retrospective + testing										
Scrum master	Eirik	Vilde	Celine	Abe	Ahmed	Kristian	Eirik	Vilde	Celine		Rapport og avsluttende testing		Presentasjon og ferdigstille rapport

Bilde 8

Eirik startet som ScrumMaster og Vilde fikk ansvar for å booke grupperom for hver uke samt være product owner. Celine fikk hovedansvaret for rapporten slik at vi vet at noen hele tiden har oversikt over hva som er gjort og hva som mangler. For å logge alt som blir sagt på møtene ble vi enige om å velge en sekretær for hvert møte, hvem som gjorde dette ble bestemt der og da.

Vi diskuterte når folk hadde tid til å møtes, og ble enige om at mandager og fredager passet best. Dermed ble det slik at vi la sprint planleggingsmøte hver mandag og sprint retrospective og sprint review hver fredag. På tirsdager møttes vi i gruppetimen. I tillegg ble vi enige om å sette av et kvarter til daglig standup møte hver onsdag, og etterhvert også tirsdager og torsdager. De ukene vi har hatt flere scrum møter har vi enten laget et sammendrag fra alle referatene og lagt til i rapporten, eller så har vi bare valgt referat fra det møtet som inneholdt det viktigste for uken.

Det første steget i prosjektet var å velge hvilket case vi ville ta i bruk. Første gang vi møttes diskuterte vi de ulike casene og gikk gjennom fordeler og ulemper, på neste møte tok vi en

felles avgjørelse på at det var case nummer én, været til sjøs, som vi ville jobbe med. I tillegg begynte vi å tenke løst på hvilke krav vi tenkte kunne være viktige.

Uken etter hadde alle med med forslag til funksjonelle krav, vi diskuterte disse og ble enige i gruppen om hvilke vi skulle ha med videre. Vi omgjorde disse kravene til brukerhistorier og satte de inn i en prioritert liste over hva vi burde bli ferdig med først.

Priority	Product Backlog
1	Som utvikler ønsker jeg å vite nåværende og fremtidige koordinater gitt nåværende posisjon og gjennomsnittlig tidsbruk til ønsket posisjon), slik at jeg kan gi informasjon som er relevant for bruker.
2	Som bruker ønsker jeg å oppgi CE-merking for min båt fordi jeg vil få informasjon om anbefalt kjøring for min båttype.
3	Som bruker ønsker jeg å få en visuell framstilling av været for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.
4	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest viktig for meg.
5	Som bruker ønsker jeg å få informasjon om bølgehøyde, vindstyrke og havstrømmer i retningen båten beveger seg fordi det er relevant informasjon og for at jeg kan finne en trygg rute.
6	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.
7	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.
8	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er for strekningen fremfor meg, fordi jeg ønsker å vite hvor trygg reisen er.
9	Som bruker ønsker jeg å få informasjon om tidevannet fordi jeg ønsker å vite om tidevannet vil påvirke min rute.
10	Som bruker ønsker jeg å kunne kontakte en vaktcentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.
11	Som bruker ønsker jeg mulighet til å se informasjon til en valgt dato/tid, fordi jeg vil kunne planlegge reisen.
12	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.

Sprint 1

Sprint planleggingsmøte

Denne uken ble vi enige om at alle skulle lage skisser med forslag til design i løpet av uken. Vi tenkte det var viktig at vi fikk laget visuelle representasjoner i form av tegninger slik at ideene våre ble mer konkrete og kunne gi oss en bedre forståelse. Skisser gjør det lettere å kommunisere og forklare ideene innad i gruppen. I slutten av uken gikk vi gjennom disse forslagene sammen.

Vi kom også frem til at vi trengte en datainnsamling for å skaffe oss mer informasjon om brukernes behov og krav. For dette tenkte vi at semi-strukturert intervju ville være den beste metoden. En slik struktur er godt egnet når man har en viss ide om hva man vil utforske, men samtidig vil ha mulighet for at temaer kan utforskes mer ved behov.

Sprint Backlog

Product backlog story	Sprint Task	Team Member	Initial Estimate of Effort (hours)
-	Presentere designforslag	Celine, Vilde, Kristian, Ahmed	1,5
	Intervjuguide	Vilde, Eirik, Ahmed, Kristian	2
	Samtykkeskjema	Vilde	0,25
	Intervju 1	Celine, Vilde, Eirik	1
	Intervju 2	Celine, Vilde, Ahmed	1
	Oppdatering av Product backlog	Celine, Vilde	1
	Sette opp disposisjon for rapporten	Celine	0,5
	Lage skisse for applikasjonen i fellesskap	Vilde, Eirik, Ahmed, Kristian, Abe	3

Datainnsamling

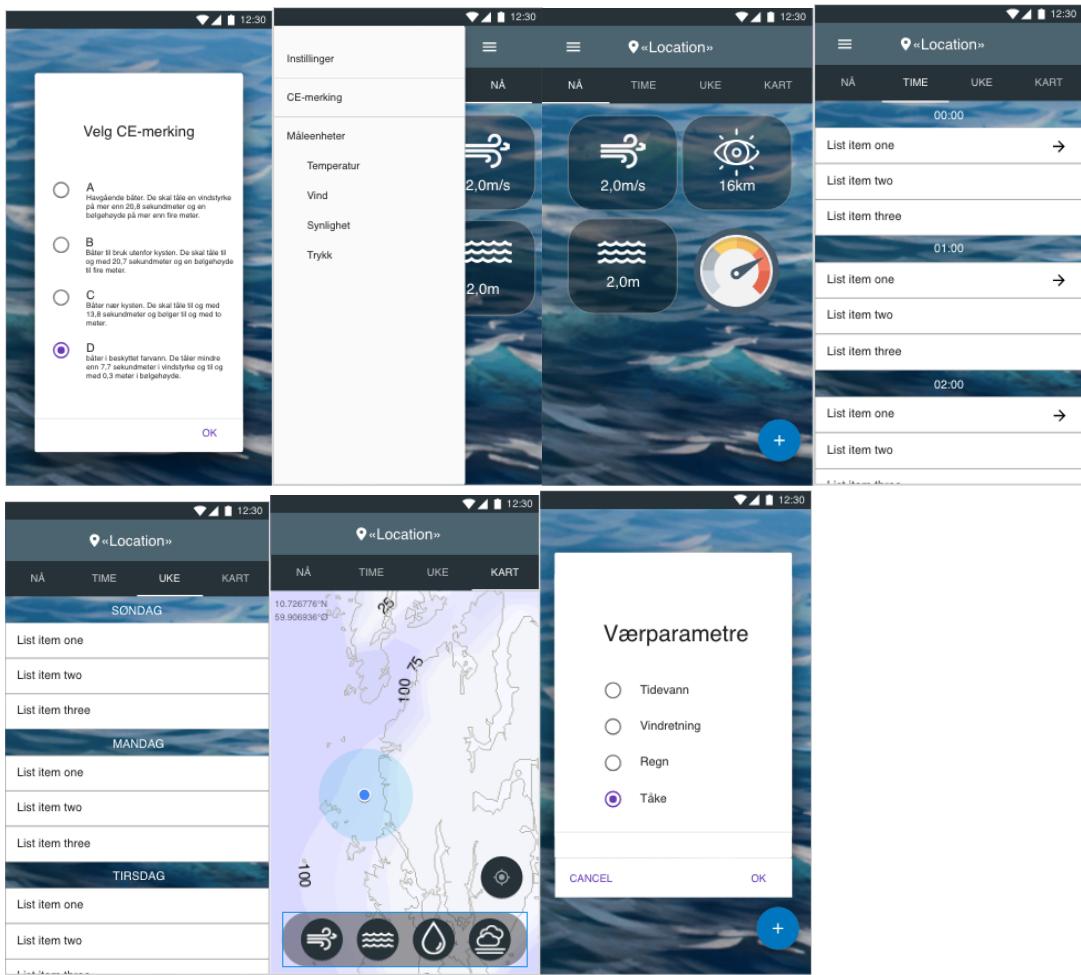
Vi lagde en intervjuguide (vedlegg 2) og et samtykkeskjema (vedlegg 3) og gjennomførte tre intervjuer i løpet av uken. Målet med disse intervjuene var å bekrefte eller avkrefte hvorvidt de kravene vi hadde satt opp stemte overens med hva brukere fra målgruppen mener, og også for å finne eventuelle nye punkt som vi selv ikke hadde tenkt på. De vi intervjuet var personer

fra målgruppen vår. Vi intervjuet to som bruker båt kun som fritidssyssel, der en har seilbåt og en har motorbåt, og en som har vært i marinens og er vant med en større båt. Vi gikk gjennom dataen og skrev ned viktige punkter. Ut fra dette oppdaterte vi product backloggen ved å endre og legge til nye krav. Vi fikk blant annet god informasjon om hvilke værparametere som ble oppfattet som de aller viktigste hos brukerne når man er ute på sjøen, disse presiserte vi i backloggen. Spesielt sikt/tåke var noe vi ikke selv hadde tenkt på før intervjuene. Vi fikk også inntrykk av at time-for-time funksjonen på yr var en viktig og god tjeneste som vi også burde ha med i vår app. Videre fikk vi inntrykk av at det ikke var så mange som visste hva CE-merking var, men når vi sa mer om beskrivelsen for hver kategori var det lett for de vi intervjuet å identifisere hvilken kategori båten deres hørte til. Derfor bestemte vi oss for at en beskrivelse må være med når bruker skal velge båttype. Vi valgte også å endre kategoriseringen til båttype i stedet for CE-merking.

Vi brukte også informasjon fra en omfattende brukerundersøkelse gjennomført av Meteorologisk Institutt. Denne undersøkelsen tok for seg maritime værtjenester, noe som var veldig passende for vårt case. Alle leste gjennom rapporten og noterte ned viktige poeng som igjen førte til endringer i backloggen. Blant annet ble det mye nevnt at det var ønske om mer bruk av kart. For eksempel var det ønskelig at posisjon bør vises når man velger et sted på et kart. Generelt var det også behov for mer detaljerte varsler, noe som gjorde at vi la til parametere som vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet. Her så vi også at time-for-time funksjonen var den mest brukte tjenesten på yr, noe som bekrefstet det vi fant ut i intervjuet. Både i intervjuene og rapporten ble det nevnt at det var behov for oppdateringer av varsling underveis, noe som gjorde at vi la inn et krav om dette.

Prototype

Siste dagen i sprinten satt vi av tid til å gå gjennom hvert enkelt krav og tegne opp forslag til design. Formålet med dette var at alle skulle få en lik grov ide om hvordan designet til applikasjonen skulle se ut og for at det skulle være enklere å starte på programmeringen. Vi gikk gjennom alle kravene fra backloggen én og én og utviklet samtidig en skisse av applikasjonen på tavlen i fellesskap. Deretter lagde vi en litt mer høyoppløselig prototype i adobe XD (bilde 9).



Bilde 9

Sprint Review

Alle oppgavene vi planla for sprinten var blitt gjort. Vi hadde ingen problemer med å gjennomføre oppgavene i sprinten. Vi gikk gjennom funn fra intervjuene og kravene som hadde blitt oppdatert for de som ikke hadde vært med på denne prosessen. I og med at vi ikke hadde startet på oppgaver som var i product backloggen var det ikke stort mer å ta for seg i denne sprint reviewen. Vi ble enige om at neste uke skulle vi ha som prioritet at alle på gruppa skulle sette seg mer inn i API og hvordan det fungerer, i og med at vi ser på det som en av de større utfordringene.

Sprint Retrospective

ScrumMaster hadde planlagt sprint retrospective der vi skulle reflektere over hvordan sprinten har gått. Det vi var fornøyde med var at vi fikk gjort alle oppgavene som vi hadde satt opp. Det vi ville forbedre oss på til neste uke var blant annet en bedre organisering og fordeling av oppgavene slik at alle vet hva de skal gjøre og at alle har noe å gjøre. Vi ble også enige om at alle må være flinke til å sjekke Google Drive og være oppdatert på hva som er gjort, spesielt dersom man går glipp av et møte.

Product Backlog etter sprint 1:

Priority	Product Backlog
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.
2	Som bruker ønsker jeg å oppgi eventuelt fremtidige koordinater, slik at jeg kan få informasjon om været dit jeg skal.
3	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.
4	Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.
5	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.
6	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.
7	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.
8	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.
9	Som bruker ønsker jeg mulighet til å se værinformasjon for et valgt sted, fordi jeg vil kunne planlegge reisen.
10	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.
11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.
12	Som bruker ønsker jeg å få informasjon om tidevannet fordi jeg ønsker å vite om tidevannet vil påvirke min rute.
13	Som bruker ønsker jeg å kunne kontakte en vaksentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.
14	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.

Sprint 2

Sprint planleggingsmøte

For å planlegge denne ukens sprint backlog tok vi utgangspunkt i product backloggen og valgte 4 av de øverste brukerhistoriene. Vi hoppet over brukerhistorien som omhandler fremtidige koordinater siden vi var litt usikre på hvordan vi skulle løse det, og hadde noen spørsmål vi måtte få oppklart først. Brukerhistoriene delte vi opp i ulike sprint tasks slik at alle hadde oversikt over hvilke oppgaver som måtte bli gjort for at brukerhistoriene skulle bli fullført. Deretter estimerte vi hvor langt tid hver oppgave kom til å ta, dette gjorde det enklere å fordele oppgaver, i og med at man da visste hvor mye tid man tok på seg. Vi ble enige om å sette opp et møte på onsdag etter daglig sprint der vi skulle gå gjennom det å hente informasjon fra API sammen. Dette er en viktig og vanskelig oppgave vi mente alle burde ha kontroll på.

For hver sprint fremover har vi satt av oppgaver som omhandler skriving i rapporten. Oppgavene som omhandler det å skrive referat fra planleggingsmøte, daglig scrum, review og retrospektive er oppgaver som gjelder for hver sprint fremover. Vi vil derfor ikke inkludere det i tabellen for resten av rapporten, siden dette blir unødvendig tekst. Det er verdt å nevne at disse oppgavene ble fordelt jevnt mellom teammedlemmene slik at alle fikk mulighet til å skrive referat i løpet av prosjektet.

Sprint Backlog

PBS	Sprint Task	TM	EST. H
Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	1. Google maps kart 2. Be om tillatelse til å bruke min posisjon 3. Min posisjon floatingActionButton 4. Lage header som skal vise lokasjon 5. ClickEvent på kart for å hente koordinater og oppdatere header som viser lokasjon 6. Begrense kartet til hva MET har informasjon om.	Ahmed Ahmed Abe Abe Ahmed Ahmed, Abe	2 1,5 0,5 0,5 3 2
Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om	1. AlertDialog første gang man åpner applikasjonen 2. Lage alternativer (A, B, C, D) og begrense til ett valg	Eirik, Abe Ahmed	3 0,5

anbefalt kjøring for min båttype.	3. Navigation Drawer for hurtiginnstillinger hvor man kan endre båttype	Ahmed	2,5
Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	1. Hente informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet fra API	Eirik, Ahmed, Celine, Vilde, Abe, Kristian	6
	2. Opprette menybar med “Nå, time for time, uke, kart” og opprett en ny activity for hver.	Ahmed, Abe	3
	3. Presentere vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet med passende ikoner på “Nå” siden.	Eirik, Kristian, Abe	6
	4. Sette vind, bølgehøyde og sikt til default	Eirik	2
Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	1. Legge til floatingActionButton (+) i activity “Nå”	Celine, Vilde	0,5
	2. AlertDialog når man trykker på (+), der man kan velge tidevann, temperatur, nedbør, fuktighet og skytetthet. (locationforecast)	Celine, Vilde	3
Rapport	1. Skrive referat fra intervju	Eirik	0,25
	2. Skrive “Forside” og “Presentasjon” delen	Celine	1
	3. Skrive om 1 og 2 sprint	Vilde	2

Daglig Scrum

Vi gikk først gjennom hva alle hadde gjort frem til nå av sine oppgaver. Deretter snakket vi om hvilke utfordringer vi har hatt. Et problem som gjaldt alle og som hadde preget store deler av sprinten var at emulatoren ikke funket optimalt. Vi jobbet med å fikse denne feilen sammen, og satt derfor av litt mer tid på møtet til dette. Et annet problem som hadde oppstått var at vi ikke fikk til å slide til aktiviteten “kart”, noe som gjorde at vi måtte snakke om endring av design. Vi ble enige om å gå gjennom det førstkomende fredag. Noen hadde også hatt problem med å gjøre sine oppgaver på grunn av at andre oppgave ikke var gjort, dette gjorde at oppgaven ble utsatt. Det gikk greit siden vi hadde andre oppgaver som kunne

gjøres i mellomtiden. Men vi ble enige om vi måtte finne en løsning slik at problemet ikke oppstår igjen. Noen andre oppgaver ble også utsatt på grunn av at vi ventet på en mail med svar fra MET, men disse oppgavene kunne vi utsette. På slutten oppsummerte vi kort hva hver enkelt skulle jobbe med til fredag.

Sprint Review

Vi gikk først gjennom backloggen og snakket om hvilke oppgaver som ikke er blitt gjort, hva som var "in progress" og hva som var ferdig. På grunn av diverse problemer beskrevet i daglig scrum fikk vi ikke tid til å bli ferdig med alle oppgavene som var satt opp. Oppgaven som gikk ut på å begrense kartet til hva MET har informasjon om ble vi ikke ferdig med fordi vi ikke hadde fått svar fra MET, og ble dermed utsatt. Brukerhistorien som omhandlet innhenting av informasjon fra APIet tok mer tid en hva vi hadde beregnet, og derfor ble vi enige om å prioritere denne uken etter. På daglig standup møte kom det frem at noen oppgaver måtte utsettes fordi de var avhengige av at andres oppgaver ble gjort først. For å unngå dette ble vi enige om at vi må se nøyere på avhengighet mellom oppgaver når vi fordeler oppgaver på planleggingsmøtene.

Vi opplevde ingen andre problemer en hva som er beskrevet i daglig scrum. Statusen til product backlog på dette punktet var at vi hadde kommet godt i gang, og at vi fortsatt føler vi ligger godt an tidsmessig.

Sprint Retrospective

Denne sprinten var det stor enighet om at vi har hatt en god arbeidsmengde og fordeling gjennom uken. Alle har jobbet godt og er engasjerte, noe som har ført til et veldig godt arbeidsmiljø innad i gruppen. Det ble også nevnt at vi samarbeider godt og er flinke til å hjelpe hverandre. Fordi vi alle har ulike timeplaner har det passet best at vi møter klokken 09 på fredager slik at vi har tre timer å disponere. På møtet kom det frem at flertallet på gruppa synes at møtetidene på fredager har vært for tidlige. For å løse dette ble vi enige om å sette av mer tid etter gruppetime på torsdager, slik at vi kan møtes en time senere på fredager.

Videre snakket vi om viktigheten rundt oppdateringer av Backloggen, og at alle må være flinkere til å skrive inn hvor de ligger an underveis i sprinten. Til slutt ble det nevnt at vi må være bedre på å si fra når man er ferdig med oppgavene sine slik at man kan hjelpe andre eller få en ny oppgave av product owner.

Product Backlog etter sprint 2:

P	Product Backlog	Sprint Backlog	In progress	Completed
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6,)	6	1, 2, 3, 4, 5

2	Som bruker ønsker jeg å oppgi eventuelt fremtidige koordinater, slik at jeg kan få informasjon om været dit jeg skal.			
3	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		1, 2, 3
4	Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4)	1,3, 4	2
5	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.			
6	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
7	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.			
8	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.			
9	Som bruker ønsker jeg mulighet til å se værinformasjon for et valgt sted, fordi jeg vil kunne planlegge reisen.			
10	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.			
11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.			
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen,			

	slik at jeg umiddelbart får informasjon om min posisjon.			
13	Som bruker ønsker jeg å få informasjon om tidevannet fordi jeg ønsker å vite om tidevannet vil påvirke min rute.			
14	Som bruker ønsker jeg å kunne kontakte en vaksentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.			
15	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.			

Sprint 3

Sprint planleggingsmøte

Under planleggingsmøtet ble vi enige om denne ukens sprint backlog. Som vi diskuterte i sprint review ble vi enige om å prioritere de oppgavene vi ikke ble ferdige med i forrige sprint. Vi ble enige om å sette av mye tid til brukerhistorien som gikk ut på å hente informasjon fra APIet. Vi ble også enige om å gjøre denne delen sammen, i og med at det er en så viktig del av prosjektet. Vi regnet med at denne oppgaven kom til å ta mye tid så vi la kun til en ekstra brukerhistorie.

Sprint Backlog

PBS	Sprint task	TM	EST.H
Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	1. Hente informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet fra API.	Eirik, Ahmed, Celine, Vilde, Abe, Kristian	12
Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	6. Begrense kartet til hva MET har informasjon om.	Ahmed	1

Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	1. RecycleView med Vind og bølge info i hovedoversikt. Resten av værparametre i dropdown meny: (værparametrene: vind, vindretning, bølgehøyde og retning, tåke, strøm)	Eirik, Abe	5
---	---	------------	---

Daglig Scrum

Daglig scrum denne sprinten hadde litt lite innhold i og med at vi hadde jobbet sammen hver dag. Vi startet hvert møte med å gå gjennom hva folk hadde gjort på egenhånd, om det var tilfellet. Problemer som oppstod ble løst i fellesskap. Et mindre problem vi møtte på var at to av APIene, "Tidalwater" og "Textforecast", ikke hadde Json format. Dette gjorde at vi måtte finne en annen løsning for disse APIene. Vi prøvde oss på retrofit først, men hadde litt problemer med denne løsningen. Et annet problem som hang igjen etter forrige sprint var at vi fortsatt ikke hadde fått svar på spørsmålene vi sendte til MET. Dette hindret oss i å fullføre noen deloppgaver. Vi sendte en ny mail og håpte på svar før neste sprint.

Sprint Review

Vi ble ikke ferdig med å bruke data fra API, men vi fikk gjort data tilgjengelig. Vi skjønte i løpet av sprinten at det hadde vært greit å delt opp disse oppgavene i to, siden oppgaven er så omfattende. Det gjenstod fortsatt noen småting i enkelte oppgaver som ikke var pusha til master. Disse oppgavene var det viktig at vi ble helt ferdig med før neste sprint. Vi ble ferdige med store deler av layout til applikasjonen, slik at alt var klart for neste sprint der vi kan putte data fra APIet rett inn.

Vi hadde litt problemer med posisjonen til kart-aktiviteten. Forrige sprint bestemte vi oss for å lage en ny bar der vi kunne plassere kartet, fordi den opprinnelige løsningen ikke var i tråd med androids design guidelines. Dette fikk vi ikke helt til, dermed bestemte vi oss for å gå tilbake til den forrige løsningen. Det var viktig for oss å ikke bruke for mye tid på detaljer i denne fasen av prosjektet.

Status på product loggen er at vi fortsatt føler vi er der vi skal være tidsmessig. Det har oppstått litt mer problemer enn planlagt, men siden vi har et godt tempo har det fortsatt blitt gjort mye. Til neste sprint måtte vi prioritere de deloppgavene som ikke var blitt helt ferdig. Hovedprioriteten for neste sprint ble å bruke informasjonen som vi har gjort tilgjengelig fra APIet. Vi tenkte også at det ville være lurt å teste om de ikonene vi tar i bruk er intuitive for brukere.

Sprint Retrospective

Fordi vi jobbet sammen med denne sprintens oppgaver var alle hele tiden oppdatert gjennom uken på hva som var gjort og ikke. Alle var enige om at sprinten i hovedsak hadde gått veldig

bra og at vi måtte prøve å opprettholde dette tempoet videre også. Forrige sprint innførte vi kakestraff for å komme for sent, dette viste seg å være en veldig effektiv løsning. I ukene før ble det nevnt at møtene må bli mer effektive, dette ble igjen nevnt. Et forslag for å bedre dette var at alle fra nå av legger vekk pc-skjermene under møtene (bortsett fra sekretæren).

Product Backlog etter sprint 3:

P	Product Backlog	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6)		1, 2, 3, 4, 5, 6
2	Som bruker ønsker jeg å oppgi eventuelt fremtidige koordinater, slik at jeg kan få informasjon om været dit jeg skal.			
3	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		1, 2, 3
4	Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5)	1, 5	2,3,4
5	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.			
6	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
7	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.			
8	Som bruker vil jeg kunne se time for time varsle i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1)	1	
9	Som bruker vil jeg kunne se ukesvarsle for valgt område slik at jeg til en hver tid er oppdatert på været fremover.			
11	Som bruker ønsker jeg mulighet til å se værinformasjon for et valgt sted, fordi jeg vil kunne planlegge reisen.			

10	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.			
11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.			
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.			
13	Som bruker ønsker jeg å kunne kontakte en vaksentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.			
14	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.			

Sprint 4

Sprint planleggingsmøte

For brukerhistorien som omhandlet informasjon om APIet gjenstod det å legge til informasjonen riktig sted, så det la vi til som to deloppgaver for denne sprinten. Oppgaven om time for time varsel ble også med i denne sprinten fordi det ikke ble ferdig uken før. Brukerhistorien om "min posisjon" fikk to nye sprint tasks som vi ikke hadde tenkt på før, disse omhandler tillatelse fra bruker om bruk av posisjon med en gang applikasjonen åpnes, og hva som skal skje dersom bruker ikke godtar dette. I tillegg ble vi enige om å gjennomføre en brukertest med mål om å finne ut hvilke vær-ikoner som er mest forståelige blant brukere, dette la vi også til som en sprint task.

Sprint Backlog

PBS	Sprint Task	TM	EST. H
Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	5.1 Legge til informasjon fra API på "NÅ" siden 5.2 Legge til informasjon fra API på "Time for time" siden	Kristian Celine, Vilde	4 4
	1. Legge til vind og bølge info i	Eirik, Abe	1,5

Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	hovedoversikt.		
Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	2. Lage dropdown med alle værparametere 7. Be om tillatelse for posisjon når applikasjonen starter	Eirik, Abe	1,5
	8. Håndtere at bruker ikke godtar bruk av posisjon	Ahmed	2
Testing	Brukertest om forståelse av ikoner	Vilde, Celine	4

Daglig Scrum

Brukertesten om forståelse av ikoner var gjennomført med åtte brukere. Videre var dropdown menyene på “time for time” siden kommet på plass, og headeren i applikasjonen oppdaterer brukerens posisjon. At applikasjonen ber bruker om tillatelse av bruk av posisjon var også blitt gjort. Et av problemene som ble nevnt var at det oppstod en loop i koden der hvor posisjonen kontinuerlig ble oppdatert, men dette ble løst slik at posisjonen kun oppdateres hver gang applikasjonen åpnes.

Sprint Review

Vi kom ikke helt i mål med alle oppgavene som ble satt opp til denne ukens sprint. Vi ble ikke ferdig med “NÅ” siden, siden vi enda ikke har fått svar fra MET og er litt usikre på hvordan utregningen for værparametere “sikt” er. Vi ble heller ikke helt ferdig med time-for-time-siden siden vi sleit med å få til bølgeparameteren. På grunn av denne feilen er heller ikke innholdet i dropdown blitt gjort. At bruker ikke godtar posisjon var satt opp som en egen oppgave, men dette blir håndtert av Google siden vi bruker deres kart. Noen av APIene var vanskeligere å benytte enn andre, f.eks var OceanForecast litt mer trøblete.

Det vi har blitt ferdige med er layout for både “Nå” og “time for time”. Det gjenstår kun små detaljer for innholdet. Statusen på backloggen har vært ganske lik de to siste ukene, dette er på grunn av at de oppgavene under disse sprintene har vært spesielt utfordrende. Dette regnet vi med kom til å ta tid og derfor føler vi ikke at vi ligger noe lengre bak i den helhetlige planen.

Til neste uke måtte vi fokusere på å bli 100% ferdig med de oppgavene vi hadde startet på, om disse ikke ble ferdige i løpet av helgen. Når vi diskuterte veien videre var vi enige om at “uke” siden måtte være førsteprioritet. Andre oppgaver som kan være relevant å starte på er trygghetsskalaen og visualisering på kartet.

Sprint Retrospective

Denne uken var vi enige om at hadde fungert veldig fint gruppemessig. ScrumMaster var syk under daglig standup, men siden forrige ScrumMaster var tilstede tok han over rollen under dette møtet. Her erfarte vi en fordel ved at alle skal prøve seg som ScrumMaster. Vi var blitt flinkere til å legge vekk pc-skjermer, slik at møtene ble mer effektive.

Product Backlog etter sprint 4:

P	PB	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6, 7,)		1, 2, 3, 4, 5, 6, 7
2	Som bruker ønsker jeg å oppgi eventuelt fremtidige koordinater, slik at jeg kan få informasjon om været dit jeg skal.			
3	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		1, 2, 3
4	Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5)	5.2	1, 2,3,4, 5.1
5	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2)		1, 2
6	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.			
7	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
8	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.			
9	Som bruker vil jeg kunne se ukesvarsel for valgt område slik at jeg til en hver tid er oppdatert på været fremover.			

11	Som bruker ønsker jeg mulighet til å se værinformasjon for et valgt sted, fordi jeg vil kunne planlegge reisen.			
10	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.			
11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.			
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.			
13	Som bruker ønsker jeg å kunne kontakte en vaktcentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.			
14	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.			

Sprint 5

Sprint planleggingsmøte

Vi var nesten ferdig med brukerhistorien som omhandlet “time for time” og manglet bare å hente ut informasjon om tidevann for hver time. Vi la til tre andre brukerhistorier som omhandlet ukesvarsel, trygghetsindikator og oppdatering av applikasjonen. Etter brukertesten som ble gjennomført forrige sprint, måtte vi denne sprinten endre på de ikonene som vi kom fram til at var mest intuitive. Vi ble også enige om endre litt på farger og design av selve applikasjonen slik at den ser litt mer ryddig og strukturert ut. Vi innså også at to av brukerhistoriene i backloggen var veldig like og omhandlet det samme:

- Som bruker ønsker jeg å oppgi eventuelt fremtidige koordinater, slik at jeg kan få informasjon om været dit jeg skal.
- Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.

Vi ble enige om å erstatte førstnevnte med sistnevnte fordi denne var mer presis og mest riktig i forhold til hva vi hadde tenkt. Vi endret også litt på prioriteringen i backloggen.

Sprint Backlog

PBS	Sprint Task	TM	EST.H
Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	5.2 Legge til informasjon fra API på “Time for time” siden	Abe, Ahmed	1
Som bruker vil jeg kunne se ukesvarsel for valgt område slik at jeg til en hver tid er oppdatert på været fremover.	1. Lage et layout for “Uke” siden. 2. Hente all data fra API 3. Legge til informasjon fra API på “Uke” siden	Celine, Vilde Celine, Vilde Celine, Vilde	1 2 2
Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	1. Lage layout for trygghetsskala delt inn i grønn, gul og rød. Skala med 10 punkter 2. Regne ut trygghet fra bølgehøyde og vindstyrke. 3. Slå sammen 1 og 2	Kristian Eirik Eirik, Kristian	2 2 1
Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.	1. Oppdater brukers posisjon når applikasjonen åpnes	Ahmed	1
Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	1. Lage layout på “kart” siden (knapper for å visuelt fremstille været)	Ahmed	2
Rapport	Dokumentere brukertest fra forrige sprint	Vilde	0,5

Annet	Bytte ut ikoner med resultat fra brukertesten om ikoner	Vilde, Celine	1
-------	---	---------------	---

Daglig Scrum

Møtet denne sprinten ble observert av veilederen vår. Kartet i applikasjonen har blitt oppdatert, den viser mer informasjon når man trykker på en marker, og applikasjonen viser også værmelding for valgt posisjon. Layouten og implementasjonen til knappen som skal vise visuell fremstilling for vind, regn og bølger er ferdig. Layouten til trygghetskalaen er også ferdig. Riktig måleenheter i "NÅ"-siden har blitt gjennomført. Time for time er også implementert ferdig, med dropdown og all dataen. Applikasjonen bruker nå alle de ikonene vi kom fram til vi skal bruke etter brukertesten av forståelse av ikoner. Et av problemene som ble nevnt var at vi er usikre på om vi skal vise gjennomsnitt av dataen i uke-siden eller om vi skal velge et fast tidspunkt hver dag som yr gjør. Vi ble enige om å velge et fast tidspunkt.

Sprint Review

Vi ble ferdige med nesten alt fra ukens sprint, det som gjenstod var å slå sammen layout for trygghetskalaen sammen med koden som regner ut trygghet. Vi ble heller ikke ferdig med å presentere data for sikt fordi vi ikke helt fant ut hvordan vi skulle regne ut dette. Vi fikk svar på mail fra meteorologisk og fikk vite at tåke er definert som sikt under 1 km. De foreslo at vi setter en terskel på 25% og sier at sikten er under 1 km hvis tåkeprosenten overstiger det. Vi ble enige om å gjøre det slik. Et av problemene denne uken var at vi fant ut at OceanForecast kun har mål på bølgehøyde for tre dager frem i tid. Dette hadde vi ikke fått med oss og hadde derfor antatt at vi kunne finne bølgehøyde for en uke frem i tid. På siden for ukesvarsel vises dermed bølgehøyde kun for tre dager. Vi valgte å la dette være inntil videre fordi vi har sett i andre apper og værvarslings sider at det ofte forekommer tomme parametere der hvor det ikke finnes data, og at dette ikke trenger å være et problem.

Vi ble enige om at alle skulle gå gjennom sin egen kode neste uke og kommentere mer der det er behov, i tillegg til å dobbeltsjekke at koden følger de organisasjonskravene som ble satt i starten av prosjektet (statisk testing). Noen av brukerhistoriene innebærer at brukeren kan søke på steder og få opp værvarsel for disse, vi sendte mail til meteorologisk angående inndeling av havområder og hvilke koordinater som dekker disse. De skjønte ikke helt hva vi mente og vi fikk dermed ikke svar på det vi lurte på. Vi diskuterte derfor om vi skulle sende en ny mail eller prøve å løse det på en annen måte, vi ble enige om å snakke mer om det på neste planleggingsmøte.

Sprint Retrospective

Noe av det vi var fornøyde med for denne sprinten var at de som hadde gjort mest av skriving og andre praktiske oppgaver fikk mer kodeoppgaver denne uken, og omvendt. Videre var vi også fornøyd med at vi fortsatt ligger godt an i forhold til product backloggen og at vi enda ikke føler at vi må stresse. Noe av det som var mindre bra denne uken var at ukens retrospektiv og sprint review ble flyttet til mandag over helgen fordi mange på gruppa ikke

hadde mulighet til å møte på fredagen. Vi merket at det var vanskeligere å snakke om sprinten etter helgen fordi det er fort gjort å glemme detaljer fra uken før, dermed prøver vi å unngå å forskyve disse møtene i fremtiden.

Product Backlog etter sprint 5:

P	Product Backlog	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6, 7)		(1, 2, 3, 4, 5, 6, 7)
2	Som bruker ønsker jeg å få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for valgt posisjon, fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5)		(1, 2,3,4,5)
3	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2)		(1, 2)
4	Som bruker ønsker jeg mulighet til å søke opp et sted for å få værinformasjon for dette stedet, fordi jeg vil kunne planlegge reisen.			
5	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	(1 ...)		1
6	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		(1, 2)
7	Som bruker vil jeg kunne se ukesvarsel for min posisjon eller valgt område slik at jeg til en hver tid er oppdatert på været fremover.	(1,2,3)		(1, 2, 3)
8	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	(1, 2, 3)	3	(1, 2)
9	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		(1, 2, 3)
10	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.			

11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.			
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.	(1)		1
13	Som bruker ønsker jeg å kunne kontakte en vaktcentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.			
14	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.			

Sprint 6

Sprint planleggingsmøte

Vi fikk svar på mail fra MET forrige uke noe som gjorde at vi kunne gjøre endringer i tidligere brukerhistorier. Vi la til brukerhistoriene som omhandlet “nå” og “time for time” oversikten, fordi vi nå kunne legge til tidevann og sikt. Forrige sprint ble vi ikke helt ferdige med brukerhistorien som omhandlet “trygghetsskalaen”. Vi la til en siste oppgave her som gikk ut på å slå sammen layout- og utregnings-kode. Denne sprinten ble vi enige om at det var viktig å rydde opp i eksisterende kode, for at vi skulle kunne sette opp Model-View-Presenter. Alle fikk i oppgave å gå over det de selv har kodet, lage gode kommentarer slik at alt er forståelig for resten av gruppen, og gi gode variabelnavn. Ettersom vi har fått til mye funksjonalitet ble vi enige om å starte planleggingen av en brukertest i løpet av denne uken. I tillegg gikk vi over brukerhistoriene og endret formuleringen slik at de ble mer presise.

Sprint Backlog

PBS	Sprint Task	TM	EST. H
Som bruker ønsker jeg å kunne få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for min posisjon og for et valgt sted på kartet fordi det er nyttig informasjon	6. Fikse “tidevann” parameter. Vise tidevann for nærmeste havn innen 30 km. Om det ikke finnes, vises ingenting.	Ahmed	2

og for at jeg kan finne en trygg rute.	7. Fikse “sikt”. 25% fåke eller mer er det dårlig sikt. God sikt hvis ikke.	Abe	1
Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	3. Fikse “tidevann” parameter. Vise tidevann for nærmeste havn innen 30 km. Om det ikke finnes, vises ingenting.	Ahmed	2
	4. Fikse “sikt”. 25% fåke eller mer er det dårlig sikt. God sikt hvis ikke.	Abe	1
Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	3. Slå sammen layout og utregning av trygghet.	Eirik, Kristian	1
Som bruker ønsker jeg å få push-varslel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.	1. Sette opp firebase 2. Gi varsel når trygghetsskalaen går over “85”.	Eirik, Kristian	1 2
Annet	Rydde opp i koden	Vilde, Celine, Eirik, Ahmed, Abe, Kristian	2
	Planlegge testing	Celine, Vilde	2

Daglig Scrum

To på gruppa hadde til i dag fått gått gjennom sin kode og ryddet opp. Mål for “sikt” i applikasjonen hadde blitt fikset. Å hente ut data for tidevann og få dette presentert i applikasjonen var også blitt gjort. Det som skulle bli gjort til neste gang innebærte planlegging av testing, fortsette med å rydde opp i resten av koden, gjøre ferdig trygghetsskalaen og få til push-varsling. En utfordring vi merket oss var at vi må hindre applikasjonen i å sende push-varslinger når brukeren er på land og ikke på sjøen.

Sprint Review

Vi ble nesten ferdig med alt fra denne ukens sprint. Eneste som manglet var å få firebase til å fungere optimalt. Grunnen til at firebase fortsatt ikke var oppe og går var på grunn av at det er mye som er “deprecated” og derfor måtte vi forsøke å lete etter oppdaterte verktøy i forhold til den. Samt funket heller ikke SOS funksjonen men dette ble fikset rett etter møtet. Det gjenstår å få til at preferanser skal fungere over alle aktivitetene, samt gjøre ferdig visualisering og få mulighet for å søke opp sted.

Annet som ble diskutert som må blir gjort var brukertesting, slå sammen layout og utregning av trygghetsverdi, fikse tidevann parameter slik at det fungerer for havner innen 30 km. Samme som forrige sprint gjelder fortsatt oppgavene om å rydde opp koden og kommentere der hvor det trengs og følger kravene som ble satt i starten av prosjektet.

Sprint Retrospective

Det vi var fornøyde med denne uken var at alle hadde bidratt og vært produktive og vi har blitt ferdig med en god del. Men vi må fortsatt få bedre kontroll på hva som står igjen, både rapport og kode. Det vi har vært misfornøyde med er at denne sprinten har vært lite effektiv og at vi ikke har fått gjort det vi har planlagt. Siden det også er påskeferie neste uke og folk er borte har vi mye å planlegge etter påsken.

Product Backlog etter sprint 6:

P	Product Backlog	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6, 7)		(1, 2, 3, 4, 5, 6,7)
2	Som bruker ønsker jeg å kunne få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for min posisjon og for et valgt sted på kartet fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5, 6, 7)		(1, 2, 3, 4, 5 ,6, 7)
3	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3, 4)		1, 2, 3, 4
4	Som bruker ønsker jeg mulighet til å søke opp et sted for å få værinformasjon for dette stedet, fordi jeg vil kunne planlegge reisen.			

5	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	(1 ...)		1
6	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
7	Som bruker vil jeg kunne se ukesvarsel for min posisjon eller valgt område slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3)		1, 2, 3
8	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	(1,2,3)		(1, 2, 3)
9	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		1, 2, 3
10	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.	(1, 2)	1,2	
11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.			
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.	(1)		1
13	Som bruker ønsker jeg å kunne kontakte en vaktcentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.			
14	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.			

Sprint 7

Sprint planleggingsmøte

Sprinten for denne uken ble litt annerledes i forhold til de andre sprintene fordi den gikk gjennom påskeferien. Flere på gruppa skulle reise bort i ferien i tillegg til at fire av seks måtte jobbe med hjemmeeksamen i et annet fag denne perioden. På grunn av dette ble vi enige om at vi forlenget sprinten med en uke, slik at folk fikk litt mer frihet til å disponere tiden i løpet av sprinten. Vi ble enige om å ta daglig standup over chat på torsdag i påskens i tillegg til vanlig standup møte uken etter.

Vi ble ikke ferdig med å sette opp firebase og sende ut pushvarsel så dette ble tatt med videre til denne spriten. Vi la til litt små oppgaver som å gjøre det mulig å konvertere mellom måleenhetene på nå-, time for time- og uke-siden. Siden ikke alle ble ferdig med å rydde opp i koden, la vi også til dette for ukens sprint.

Sprint Backlog

PBS	Sprint Task	TM	EST.H
Som bruker ønsker jeg å kunne kontakte en vaksentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.	1. Lage layout for SOS-knappen (slide-knapp)	Eirik	1
	2. Lage funksjonalitet for å sende melding til en tlf	Eirik, Ahmed	1
	3. Sende koordinatene til "min posisjon"	Ahmed	0,5
Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.	1. "Nå" - siden. - Endre vindretning til norsk - Mellomrom mellom tall og måleenhet - Gjør det mulig å endre fra km/t til m/s til default	Kristian	2
	2. "Time for time" - siden Gjøre det mulig å endre måleenhet for: - temperatur - vindstyrke - Mellomrom mellom tall og måleenhet	Abe	2
	3. "Uke" - siden - Gjør det mulig å endre måleenhet for vindstyrke - Mellomrom mellom tall og måleenhet	Celine	2
Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.	1. Sette opp firebase	Eirik	2
	2. Gi varsel når trygghetsskalaen går over "85".	Kristian	4
Rapport	Skrive om universell utforming	Celine	2
	Skrive om sikkerhet	Vilde	2
	Lese gjennom rapporten og endre/strukturene	Vilde	2

	på tekst.		
Annet	Rydde opp i koden	Eirik, Ahmed, Abe, Kristian	2-4

Daglig Scrum

Scrum møtet vi skulle ha over chat ble gjennomført som planlagt. Fordi de fleste på gruppa hadde vært opptatt med hjemmeeksamen og noen av oss var bortreiste hadde det ikke blitt gjort så mange oppgaver fra sprinten. Dette var noe vi var klar over at ville bli slik for denne uken. Det som dermed var blitt gjort var oppgavene som hørte til brukerhistorien om kontakt med vaksentral, det var også blitt gjort arbeid på rapporten.

I andre daglig scrum var det endel som var blitt gjort. Det eneste som gjenstod var å gjøre ferdig firebase og gi varsel når trygghetsskalaen går over 85. Denne oppgaven var litt utfordrende og vi ble enige om å samarbeide for å gjøre den ferdig fremfor at product owner skulle tildele flere oppgaver til de andre.

Sprint Review

Vi var blitt ferdig med alle de oppgavene som var satt opp for denne sprinten bortsett fra push-varsling. Denne oppgaven har hengt igjen lenge uten at det har fungert som det skal, firebasen har vi fått til, men ikke at push-varsling blir sendt automatisk. Vi ble enige om å la det være slik siden det har blitt jobbet mye med firebasen, men at vi ikke kan sette av enda mer tid til det. Brukerhistorien som omhandlet at bruker kan kontakte vaksentral var blitt gjort. Videre hadde vi noen oppgaver som omhandlet at bruker kan velge måleenhet for temperatur, vind, synlighet og trykk. Dette var blitt gjort for både "nå" siden, "uke" siden og "time for time" siden. For disse sidene var det også fikset at vindretning oppgis på norsk fremfor engelsk, og det var lagt til mellomrom mellom måleenhet og verdi slik at det er lettere å lese.

I backloggen var det hovedsakelig to brukerhistorier som stod igjen, disse omhandlet at bruker skal kunne søke på steder og få opp værvarsler og trygghetsskala for disse. Vi hadde enda ikke fått svar på mail fra meteorologisk om inndeling av havområder og hvilke koordinater som dekker disse, noe vi må vite for å starte på disse oppgavene. Hvis vi ikke får svar fra met over helgen, må vi heller prioritere arbeid med rapporten og applikasjonens design til neste sprint.

Sprint Retrospective

Fordi disse to ukene hadde vært hektiske for mange var vi enige om at det var fornuftig å la sprinten gå over to uker. Vi var fornøyde med at alle oppgavene som var satt opp i sprinten

var blitt ferdig. Siden vi har fått litt tid til å prioritere andre fag er vi nå veldig motiverte til å gjøre en god innsats i sluttspurten av dette prosjektet.

Product Backlog etter sprint 7:

P	Product Backlog	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6, 7)		1, 2, 3, 4, 5, 6, 7
2	Som bruker ønsker jeg å kunne få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetthet for min posisjon og for et valgt sted på kartet fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5, 6, 7)		1, 2, 3, 4, 5 ,6, 7
3	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3)		1, 2, 3
4	Som bruker ønsker jeg mulighet til å søke opp et sted for å få værinformasjon for dette stedet, fordi jeg vil kunne planlegge reisen.			
5	Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	(1)		1
6	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
7	Som bruker vil jeg kunne se ukesvarsel for min posisjon eller valgt område slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3)		1, 2, 3
8	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	(1,2,3)		1, 2, 3
9	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		1, 2, 3
10	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.	(1, 2)		1,2

11	Som bruker ønsker jeg å få en indikasjon på hvor trygt det er på et valgt sted, fordi jeg ønsker å vite hvor trygg reisen er.			
12	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.	(1)		1
13	Som bruker ønsker jeg å kunne kontakte en vaktcentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.	(1, 2, 3)		1,2,3
14	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.	(1, 2, 3)		1,2,3

Sprint 8

Sprint planleggingsmøte

Vi hadde fortsatt ikke fått noe svar fra MET som angikk de to siste brukerhistoriene. Derfor ble vi enige om at vi strøk disse to brukerhistoriene fra product backloggen og heller fokuserte på den koden som gjenstod og kvaliteten på denne, i tillegg til arbeid på rapporten. Det som gjenstod av krav etter å ha fjernet disse var brukerhistorien som omhandler visualisering på kartet. Denne startet vi såvidt på tidligere, så her gjenstår det å vise data på kartet. Andre nye oppgaver vi satt opp denne sprinten var å skrive delen om testing i rapporten. Vi ble også enige om å finjustere designet denne sprinten. En annen oppgave vi satte av tid til denne sprinten var å gå over all koden og sjekke at det følger design patternet Model View Presenter. Ahmed fikk hovedansvar for å sette seg inn i hva som må gjøres, slik at han kan deletere oppgaver til resten når man er ferdige med sine oppgaver. Ellers ble vi også enige om å ha en daglig scrum tirsdag, onsdag og torsdag.

Vi valgte å endre formuleringen av kravet om visualisering av kartet:

- Som bruker ønsker jeg å få en visuell framstilling av regn og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.

Dette var fordi vi fant ut at å ta i bruk data fra to ulike APIer samtidig krever mange API kall på en gang. Derfor måtte vi velge mellom visualisering av vind eller bølgehøyde og havstrøm. Vi valgte å fjerne bølgehøyde og havstrøm, siden vi anser vind som den viktigste parameteren. Vi valgte å ta med regn siden denne informasjonen finnes i samme API og vi så på det som nyttig informasjon.

Sprint Backlog

PBS	Sprint Task	TM	EST.H
Som bruker ønsker jeg å få en visuell framstilling av regn og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	2. Vise data i kartet	Ahmed	8
Rapport	Skrive om scrum som metode	Vilde	2
	Skrive om testing	Abe	4
Annet	Strukturere koden til MVP	Ahmed, <i>Eirik,</i> <i>Kristian,</i> <i>Vilde,</i> <i>Celine,</i> <i>Abe</i>	16
	Komme med forslag til endringer og endre design.	Vilde, Celine	5

Daglig Scrum

Alle hadde kommet godt i gang med oppgavene sine og så for seg å bli ferdige til fredagen. I tillegg til hovedoppgavene for denne sprinten ble det oppdaget noen feil i løpet av uken som også måtte fikses. Blant disse problemene var en feil i beregning for fahrenheit, unødvendig kategori "synlighet" i sidebaren, og engelsk beskrivelse av værparametre. Disse var fikset til dagens daglige scrum. Et problem en av oss nevnte under møtet var spørsmål om hvordan man får opp "tidevann" parameteren for nå siden. Dette ble løst i daglig scrum. De av oss som var ferdige med oppgavene våres, fikk oppgaver som omhandlet strukturering av kode fra Ahmed som var ansvarlig for MVP.

Sprint Review

For denne uken hadde vi blitt ferdige med alt som var satt opp i sprinten. Visualisering av regn og vind var lagt til på kartet. Vi hadde fått skrevet i rapporten om hvordan vi har testet applikasjonen og koden hittil i prosessen. Videre hadde vi fått gått gjennom all kode og sjekket om alt var strukturert i forhold til MVP. Her var det mest i orden, men koden for nå-siden og uke-siden måtte fikses litt, noe vi ble enige om å gjøre neste uke. Design endringene

som ble gjort handlet mest om farger, men vi la også til et lokasjons-ikon i headeren der hvor koordinatene til brukerens posisjon vises slik at det ble tydelig at koordinatene representerer posisjon. Problemene som oppstod denne uken handlet mye om emulatoren og at denne ofte ikke fungerer optimalt. Når vi snakket om prioriteringer for neste sprint bestemte vi at vi måtte fokusere på å ordne opp i små feil i applikasjonen som må fikses før vi kan si at alle brukerhistoriene i backloggen er 100% oppfylte. Hva alle disse feilene innebærer skulle vi ta opp på planleggingsmøtet for neste sprint. Vi var også enige om at en akseptansetest burde gjennomføres neste sprint.

Sprint Retrospective

Det ble ikke tatt opp noe spesielt angående hva vi var fornøyd og misfornøyd med. Samarbeidet fungerer fortsatt godt og vi var fornøyde med arbeidsinnsatsen og hvor langt vi var kommet.

Product Backlog etter sprint 8:

P	Product Backlog	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6, 7, 8)		1, 2, 3, 4, 5, 6, 7, 8
2	Som bruker ønsker jeg å kunne få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for min posisjon og for et valgt sted på kartet fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5, 6, 7)		1, 2, 3, 4, 5 ,6, 7
3	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3)		1, 2, 3
4	Som bruker ønsker jeg å få en visuell framstilling av regn og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	(1, 2)		1, 2
5	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
6	Som bruker vil jeg kunne se ukesvarsel for min posisjon eller valgt område slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3)		1, 2, 3

7	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	(1,2,3)		(1, 2, 3)
8	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3)		1, 2, 3
9	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.	(1, 2)		1, 2
10	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.	(1)		1
11	Som bruker ønsker jeg å kunne kontakte en vaktcentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.	(1, 2, 3)		1,2,3
12	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.	(1, 2, 3)		1, 2, 3

Sprint 9

Planleggingsmøte

Som vi snakket om på forrige ukes sprint review møte måtte vi denne uken prioritere å fikse alle feil og mangler. Fordi det ikke stod igjen noen brukerhistorier i backloggen bestemte vi oss for å legge til et par nye som omhandlet ting vi har snakket om før, men som vi ikke visste om vi fikk tid til. Vi la til en brukerhistorie som omhandler at man skal kunne bytte mellom norsk og engelsk i applikasjonen, og en som omhandler at man skal kunne få opp en visuell oversikt over alle norske havner. Sistnevnte skal være et alternativ sammen med visuell fremstilling av vind og regn på kartet. Oppgaver for dette ble lagt til i sprint-backloggen.

Sprint Backlog

PBS	Sprint Task	TM	EST.H
Rapport	Ferdigstille sekvensdiagram	Vilde, Celine	2

Annet	Legge til et loading-ikon til applikasjonen er ferdig loadet.	Ahmed	1
	Fikse MVP for time-for-time	Abe	4
	Fikse MVP for uke.	Vilde, Celine	4
	Oppdatere recyclerview slik at data fra ocean blir oppdatert.	Abe	3
Som bruker ønsker jeg å få en visuell oversikt over norske havner slik at jeg lettere kan orientere meg til sjøs.	1. Legge til riktig ikon for havner.	Vilde	0,5
Som bruker ønsker jeg å få en visuell framstilling av havstrøm, bølgehøyde og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	3. Endre design for visualisering av vind og regn på kartet	Ahmed	1
Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.	4. Oppdatere dataen når man endrer på måleenhetene 5. Fikse det å legge til og fjerne værpreferanser i recyclerviewen i nå-siden	Eirik Kristian	4 4
Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	4. Ha lengre beskrivelse av båttype, og endre layout slik at det blir mer avstand mellom beskrivelsen.	Celine	1
Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	4. Lage en if sjekk for at time-for-time startet på nåværende time, ikke forrige.	Abe	0,5
Som bruker ønsker jeg å kunne oversette applikasjonen til engelsk slik at jeg kan velge mitt foretrukne språk.	1. Oversette strings fra norsk til engelsk slik at applikasjonen både brukes på engelsk og norsk.	Ahmed	1

Daglig Scrum

Til møtet i dag var det gjort mange oppgaver fra sprint backloggen. Riktig ikon for å få opp havner, og det visuelle designet av regn og vind på kartet var blitt gjort. Loading-animasjonen når applikasjonen laster var også i orden. Videre var alle strengene blitt oversatt til engelsk slik at man kan endre språk i applikasjonen. MVP for time-siden og uke-siden var også på plass til i dag. De litt større oppgavene som omhandlet å oppdatere dataen når man endrer på måleenhetene og at recyclerviewen blir oppdatert når man legger til/fjerner værparametre i preferanser var blitt startet på, men disse måtte jobbes videre med over helgen.

Sprint Review

Vi ble ferdige med alle oppgavene som var satt av for denne sprinten. I løpet av sprinten oppdaget vi to feil som ikke var en del av sprinten, men disse ble vi ferdige med i løpet av den siste dagen. De største problemene vi har hatt er det at det har dukket opp nye feil i det vi skulle fikse andre problemer. Vi er veldig fornøyde med at vi er nå er så og si ferdig med product backloggen. Dette var siste sprint, så de tingene som sto igjen blir beskrevet i slutten av rapporten under "i hvilken grad ble kravspesifikasjonen oppfylt".

Sprint Retrospective

Teamet var veldig fornøyde med at vi har klart å prioritere dette emnet så godt denne uken at vi klarte å bli ferdige med backloggen. Vi har vært spesielt flinke til å hjelpe hverandre og det var en viktig faktor i denne innspurten. Ingen hadde noe de var misfornøyde med denne sprinten.

Product Backlog etter sprint 9:

P	Product Backlog	SB	IP	C
1	Som bruker ønsker jeg å kunne finne min posisjon på et kart, fordi jeg ønsker å se posisjonen min.	(1, 2, 3, 4, 5, 6, 7, 8)		1, 2, 3, 4, 5, 6, 7, 8
2	Som bruker ønsker jeg å kunne få informasjon om vind, vindretning, bølgehøyde, sikt, tidevann, temperatur, nedbør, fuktighet og skytetethet for min posisjon og for et valgt sted på kartet fordi det er nyttig informasjon og for at jeg kan finne en trygg rute.	(1,2, 3, 4, 5, 6, 7)		1, 2, 3, 4, 5,6, 7
3	Som bruker vil jeg kunne se time for time varsel i det området jeg befinner meg i slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3, 4)		1, 2, 3, 4
4	Som bruker ønsker jeg å få en visuell framstilling av regn og vind for nåværende posisjon slik at jeg kan få et direkte visuelt inntrykk av værforholdene.	(1, 2, 3)		1, 2, 3

5	Som bruker ønsker jeg å filtrere informasjon, slik at jeg kan velge hvilken informasjon som er mest relevant for meg.	(1, 2)		1, 2
6	Som bruker vil jeg kunne se ukesvarsel for min posisjon eller valgt område slik at jeg til en hver tid er oppdatert på været fremover.	(1, 2, 3)		1, 2, 3
7	Som bruker ønsker jeg å vite hvor trygt det er på min nåværende plassering fordi jeg vil vite om plasseringen min er trygg.	(1,2,3)		(1, 2, 3)
8	Som bruker ønsker jeg å oppgi hvilken type båt jeg har basert på en oppgitt beskrivelse, fordi jeg vil få informasjon om anbefalt kjøring for min båttype.	(1, 2, 3, 4)		1, 2, 3, 4
9	Som bruker ønsker jeg å få push-varsel om truende endringer i værforholdet fordi jeg vil vite om det er umiddelbar grunn til å snu.	(1, 2)	2	(1)
10	Som bruker ønsker jeg at min posisjon oppdateres når jeg åpner applikasjonen, slik at jeg umiddelbart får informasjon om min posisjon.	(1)		1
11	Som bruker ønsker jeg å kunne kontakte en vaksentral, og raskt oppgi min posisjon slik at jeg kan få hjelp dersom det oppstår en nødsituasjon.	(1, 2, 3)		(1,2,3)
12	Som bruker ønsker jeg å kunne velge måleenhet for temperatur, vind, synlighet og trykk, slik at jeg får oppgitt informasjon på en foretrukket måte.	(1, 2, 3, 4, 5)		(1, 2, 3, 4, 5)
13	Som bruker ønsker jeg å få en visuell oversikt over norske havner slik at jeg lettere kan orientere meg til sjøs.	1		1
14	Som bruker ønsker jeg å kunne oversette applikasjonen til engelsk slik at jeg kan velge mitt foretrukne språk.	1		1

I hvilken grad ble kravspesifikasjonene oppfylt?

Alle kravene i kravspesifikasjonen ble gjennomført eller startet på. Som vi ser i siste oppdatering av product backlog (tabellen over) var det kun én oppgave vi ikke ble helt ferdig med, nemlig kravet som omhandler push-varsel. Som beskrevet i prosessdokumentasjonen fikk vi kun til å sende push-varsel manuelt og ikke automatisk. Dette er noe vi hadde prioritert dersom vi skulle jobbet videre med applikasjonen.

Mot slutten av prosjektet fjernet vi et krav som vi egentlig så på som en av hovedfunksjonalitetene i appen. Dette kravet omhandlet søking av steder på havet. Her

trengte vi informasjon fra MET som vi dessverre ikke fikk. Denne funksjonen er noe vi gjerne ville fått til om vi skulle jobbet videre med prosjektet.

Videre var det noen feil som vi oppdaget for sent til å løse. En av disse er at når man endrer båttype eller måleenheter i innstillinger vil posisjonen automatisk endre seg til min posisjon. Denne feilen ser vi ikke på som veldig stor, i og med at å endre båttype og måleenhet ikke er ment å gjøre ofte.

En annen ting vi ville jobbet mer med var design av applikasjonen. Tidlig fant vi ut at bølgehøyde og vind var de viktigste parametrene for målgruppen vår, derfor ville vi ha dette i fokus på alle værmelding-sidene. Da vi skulle hente data for bølgehøyde til uke-siden fant vi ut at APIet kun hadde data for tre dager frem i tid og ikke én uke slik vi hadde tenkt. Dette gjorde at fremstilling av bølgedata på uke-siden ser litt uferdig ut. Vi diskuterte om vi skulle gjøre noe med det men ble enige om å ikke prioritere dette.

Andre endringer vi ville jobbet videre med er funnene fra akseptansetesten. Her fant vi ut at det ikke var intuitivt for alle at man kan få opp en dropdown med mer informasjon på timesiden. Videre var det også blandede meninger om man burde benytte langvarige trykk på kartet når man skal endre posisjon eller ikke.

Avsluttende refleksjon

Prosjektarbeidet har vært svært spennende og lærerikt, men også utfordrende. Vi har erfart viktigheten av kommunikasjon spesielt i en prosess der vi har kodet og jobbet på samme plattform. Kommunikasjon innad i gruppen har også vært en av utfordringene i prosjektet. Folk har hatt forskjellige timeplaner med jobb og annet skolearbeid, og derfor har vi hatt flere møter der ikke alle har vært tilstede. Dette har vi løst ved å være flinke til å loggføre hvert eneste møte, slik at de som ikke er tilstede ikke går glipp av noe. Noen har likt å jobbe tidlig på dagen mens andre har likt å jobbe senere, og dette har gjort det litt vanskeligere å planlegge arbeid sammen. Utover i prosjektet ble timeplanene til hver enkelt tilpasset mer og mer til prosjektets faste tider, slik at møtene ble mer effektive.

Vi er enige om at vi har vært flinke med å fordele arbeidet jevnt utover prosjektet. Men det var spesifikt en periode hvor vi slet med å følge tidsplanen. Dette var når fire på gruppen hadde hjemmeeksamen i et annet emne. I tillegg til hjemmeeksamen var det også påske i denne perioden. Dette gjorde at prosjektet ikke hadde så stor fremgang i denne perioden. Når denne perioden var over ble gruppen enig om å prioritere dette emnet, slik at vi tok igjen tapt tid. Dette gjorde at vi ble à jour med tidsplanen.

En annen utfordring vi har hatt i prosjektet er at Android Studio og emulator har vært krevende for datamaskinene. Dette har gjort at implementasjonen har gått saktere enn forventet. Noen datamaskiner slet så mye at det var bedre at de ikke fortsatte å kode på sin pc og at de heller begynte med parprogrammering. Derfor ble det også skeivfordeling av commitsene i GitHub.

Prosjektet har vært svært jobbrelevant og vi har fått mer erfaring med å jobbe i team. Det har vært spennende å lære et nytt språk, IDE og jobbe med API. Vi syns det er spesielt spennende at vi har fått mulighet til å lage en applikasjon som har en nytteverdi for andre enn oss. Vi har lært å sette opp en plan over en lang periode og effektivisere arbeidet etter tiden vi har, samt tilpasse planen etter utfordringene som kommer frem. Om noen var ferdige med sine oppgaver eller hadde tid til overs, har de vært flinke til å hjelpe de som har hatt problemer. En annen viktig men undervurdert læring, var det å klare å legge fra seg en oppgave om det ikke lot seg gjøre.

Alt i alt er vi alle enige om at prosjektet har vært svært vellykket og ikke minst lærerikt. Vi er også stolte av resultatet som har kommet fra hardt arbeid og godt teamarbeid.

Kilder

Ankush Grover. (2018). Basics of MVP—The Android Way
[<https://hackernoon.com/basics-of-mvp-the-android-way-f75da407019d>] Tilgjengelig fra 2018-02-27 [01.04.2019]

Antonio Leiva. (2019) MVP for Android: how to organize the presentation layer
[<https://antonioleiva.com/mvp-android/>] [01.04.2019]

Jinn Kinn. (2018). Getting Started with MVP (Model View Presenter) on Android
[<https://www.raywenderlich.com/7026-getting-started-with-mvp-model-view-presenter-on-android>] Tilgjengelig fra 2018-12-19 [01.04.2019]

Jøsang, A. (2019). Software Engineering og prosjektarbeid, sikker systemutvikling [<https://www.uio.no/studier/emner/matnat/ifi/IN2000/v19/forelesninger/in2000.2019.02.07.sikkersystemutvikling.pdf>]. Universitetet i Oslo.

Google (2019) [<https://material.io/design/usability/accessibility.html#composition>]. [08.03.2019]

Guru99 (2019) Enhetstesting <https://www.guru99.com/unit-testing-guide.html#1> [05.04.2019]

Guru99 (2019) Integrasonstesting <https://www.guru99.com/integration-testing.html> [05.04.2019]

Gjesdal, K, Goa, K, Olsen, A.M, Olsen, T. (2015) “Været til sjøs. Brukerundersøkelse om maritime værtjenester [https://www.met.no/publikasjoner/met-info/met-info-2015/_attachment/download/be90e558-3358-4008-acc7-d6ee658531f4:6ad1dd5c77926999b7e30decdadd698c8e43c1c2/MET-info-15-2015.pdf]. Tilgjengelig fra 2015-02-05. [06.03.2019]

Rakshit Soral. (2018). Architectural Guidelines to follow for MVP pattern in Android [<https://android.jlelse.eu/architectural-guidelines-to-follow-for-mvp-pattern-in-android-2374848a0157>] [01.04.2019]

Rubin, K.S. (2012) Essential Scrum, A practical guide to the most popular agile process. United States: Pearson Education.

Wikipedia (2018) [https://no.wikipedia.org/wiki/Beauforts_skala] [16.05.19]

Y. Lindsjørn, (2018). Mer om objektorientering og UML. Universitetet i Oslo. Universitetet i Oslo.

Vedlegg

Vedlegg 1 "Klassediagram"

Vedlegg 2 "Intervjuguide"

Vedlegg 3 "Samtykkeskjema"

Vedlegg 4 "Brukertest"

Vedlegg 5 "Akseptansetest"