

Forprosjekt

Kristian Nedrevold, Daria Danilina

March 22, 2005

UiT Norges arktiske universitet

Fakultet for Ingeniørvitenskap og Teknologi
Institutt for Datateknologi og Beregningsorientert Ingeniørfag

REST API kodeverk

Nedrevold, Kristian
Danilina, Daria

Forprosjektrapport, 2023-Jan-25

Bachelorppgave 2023 - Forprosjekt

Introduksjon

Helse Nord IKT er et eget foretak under Helse Nord paraplyen som i hovedsak leverer tjenester til andre foretak i Helse Nord. Seksjon for systemutvikling i avdeling for Tjenesteutvikling er delt opp i ulike team.

Kvalitetsregister

Denne oppgaven er på bestilling fra team for kvalitetsregister.

Helse Nord IKT leverer i dag følgende kvalitetsregister: * Norsk gynekologisk endoskopiregister * Norsk register for invasiv kardiologi * Norsk kvalitetsregister for behandling av spiseforstyrrelser * Norsk register for arvelige og medfødte nevromuskulære sykdommer * Register for Hidradenitis suppurativa * Norsk register for analinkontinens * Norsk register for gastrokirurgi * Norsk kvalitetsregister for endokarditt * Nasjonalt kvalitetsregister for ryggkirurgi

Et medisinsk kvalitetsregister er en registreringsløsning for medisinske data relatert til et spesifikt fagfelt, der data samles inn for å brukes til analyse og forskning. Et eksempel på et slikt register er Norsk Gynekologisk Endoskop Register(NGER).

Registeret samler inn data om(kvalitetsregistre, 2023[cited 2023Jan23]):

- Konvertering til laparoskopi (ut fra hysteroskopi)/ laparotomi (ut fra hysteroskopi, laparoskopi)
- Intraoperative komplikasjoner
- Postoperative komplikasjoner
- Reoperasjon for komplikasjoner innen 4 uker
- pasientens helsegevinst og
- tilfredshet med behandelende enhet

[ref]<https://www.kvalitetsregistre.no/register/gynekologi/norsk-gynekologisk-endoskopiregister>

Ved å samle inn data fra alle pasienter som blir endoskopisk operert for gynekologiske tilstander og sykdommer ved offentlige og private sykehus er det da mulig å utføre statistiske analyser for å identifisere positive og negative aspekter ved det enkelte behandlingssted og på tvers av behandlingssted (kvalitetsregistre, 2023[cited 2023Jan23]).

[ref]<https://www.kvalitetsregistre.no/register/gynekologi/norsk-gynekologisk-endoskopiregister>

Data samlet inn via slike medisinske kvalitetsregistre danner grunnlag for analyse og forskning som har som formål å gi økt behandlingskvalitet ved norske sykehus. Innregistrering av data til medisinske kvalitetsregistre blir ofte utført av helsepersonell, ofte i en hektisk verdag der de har mange andre arbeidsoppgaver. Det er av den grunn viktig at det er mulig å utføre registrering av data på en enkel måte med liten risiko for feil. Forbedring av kvalitet skjer ofte via kvalitetsforbedringsprosjekter. <https://www.kvalitetsregistre.no/kvalitetsforbedring>

Helse Nord IKT leverer en plattform for medisinske kvalitetsregister som er basert på OpenQreg. OpenQreg er en av tre godkjente plattformer for medisinske kvalitetsregister i Norge. Teamet jobber med å utvikle plattform for kvalitetsregister samt de spesifikke register som bruker plattformen. <https://www.nhn.no/tjenester/medisinske-kvalitetsregistre>. OpenQreg plattformen er opprinnelig utviklet av Uppsala Clinical Research Centre som en åpen basis plattform for kvalitetsregistre [kilde]. HN-IKT har nå sin egen “fork” av denne som vedlikeholdes og videreutvikles.

1.1 Bakgrunn for oppgaven

For registrering av medisinske kvalitetsparametre brukes det diverse kodeverk. Et praktisk eksempel på et kodeverk vi alle kjenner er postnummer. Et postnummer er en lenke til informasjon om et sted, den inneholder informasjon om hvilket fylke og by det referes til. Postnummer er et av kodeverkene applikasjonen skal omfavne. Den skal også omfavne medisinske kodeverk. De kobler en kode opp mot en diagnose eller tilstand.

Kodeverk som brukes er [Skriv ned alle]. I tillegg kreves det oppdatert informasjon om opplysninger som norske postnummer, kommuner, fylker osv [fyll på mer her]. Dette er i dag opplysninger som er hardkodet inn i applikasjonens database, og som lastes inn ved oppstart av applikasjonen.

Med dagens løsning kan hvert enkelt kvalitetsregister oppdage at diverse koder eller dokumenter ikke er oppdatert til nyeste versjon. Dette fører til at data ikke kan registreres korrekt av helsepersonellet som utfører registreringen. Når en sluttbruker oppdager dette vil den ta kontakt med utviklerne via felles epost portal, en utvikler vil så opprette SQL script for oppdatering til nye koder. Dette scriptet sendes til drifts-personell som så kjører scriptet i produksjonsdatabase. Prosessen gjentas for alle 13 kvalitetsregistre. Dette er en ugunstig prosess da den stjeler tid fra sluttbruker og utviklere.

Det er derfor ønskelig med en felles tjeneste som kan hente inn koder fra ulike kilder(API, filer osv.) sammenfatte og versjonsstyre kodene. For så å levere de til register applikasjonene via et REST API. Det er altså ønskelig automatisere oppdaterings prosessen i størst mulig grad, og på denne måten kunne tilby mest mulig oppdatert data. Dette vil være til gevinst for både sluttbruker og for utviklere.

Det er da også naturlig at det implementeres en klient til APIet i registrenes felles kode. Da er det naturlig å tenke at koder ikke lengre lagres i SQL databasen, men heller i en type in-memory database for raskere oppslag i applikasjonen. In-memory databasen settes opp slik at den oppdateres hver gang det kommer nye koder i REST APIet.

Det er også uttrykt et ønske om mulighet for laste versjonerte dokumenter inn i applikasjonen som kan lagres som binary blobs. De skal også tilgjengeliggjøres via APIet. Dette ønskes da hvert kvalitetsregister lagrer og tilgjengeligjør et antall dokumenter for ulike formål. Det lagres også maler for meldinger som skal sendes til inbyggere. Det er da ønskelig å ha en felles portal for å oppdatere og tilgjengeliggjøre disse dokumentene.

En siste årsak til ønske om en felles tjeneste for dokumenter og kodeverk er alderen og den akkumulerte tekniske gjelden i OpenQreg platformen. For å kunne vedlikeholde å videreutvikle platformen har team for kvalitetsregister begynt å migrere til en mer mikrotjeneste basert arkitektur, der funksjonelle deler av applikasjonen flyttes til egne tjenester. Dette gjøres for å på lang sikt kunne slutte å bruke OpenQreg platformen. Ved å opprette en API basert applikasjon for styring av kodeverk og dokumenter blir opprettelse av fremtidig ny plattform arkitektur lettere.

1.2 Prosjektbeskrivelse og analyse

Utviklingen vil foregå i to ulike kodebaser. Selve tjenesten som skal hente data fra eksterne kilder og i den eksisterende register koden der HTTP klient og in-memory database skal etableres.

Alexandria: Applikasjonen vi skal utvikle har fått navnet Alexandria etter det store biblioteket i Alexandria i Egypt.

Eksisterende backend:

1. HTTP Klient som skal hente oppdaterte data fra Alexandria etter mottak om oppdaterte data
2. Storage service som lagrer og henter data fra in-memory database
3. In-memory database

Dataflyt i applikasjonen

Grønne ruter skal implementeres som en del av prosjektet.

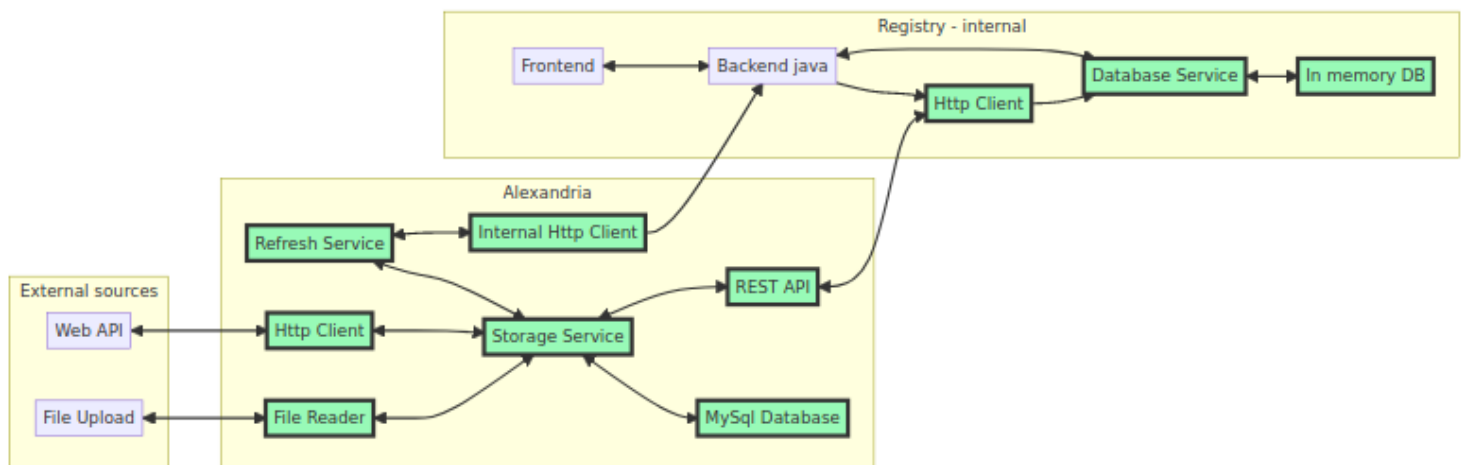


Fig 1. Dataflyt i applikasjonen. Generert med mermaid

Teknologier:

Kvalitetsregister teamet bruker JVM teknologier for sin utvikling, da spesifikt Java og Kotlin, der hovedkodebasen er en monolitt skrevet i Java som tar i bruk mange ulike teknologier. Ny kode skrives som regel i Kotlin. Tjenester som er ekstern til monolitten skrives i Kotlin. Med dette tatt i betraktning skal også denne tjenesten skrives i Kotlin. Alle nye tjenester kjører i kontainermiljø, der produksjonsmiljøet er driftet av Norsk Helse Nett(NHN). Hele kvalitetsregisterplattformen til HN-IKT er i en migreringsprosess der den skal over i Kubernetes. Det er derfor viktig at applikasjonen utvikles slik at den enkelt kan kjøre i et slik miljø.

Hovedteknologier som skal brukes for tjenesten * Kotlin * Ktor web server og http klient rammeverk * R2DBC for databasetilgang * Gradle som byggverktøy * Docker og docker-compose for kontainer kjøremiljø og oppsett * MySql database

1.4 Rammebetingelser

Utvikling kan skje fra egen PC. Anbefalt å bruke IntelliJ IDEA for utvikling av Kotlin kode. Team for kvalitetsregister må tilgjengeligjøre kildekode for utvikling av komponenter som er intern i register koden, eventuelt veilede i hvordan et eksternt lib kan opprettes slik det kan føyes inn i applikasjonen.

Brukere, brukermiljø og behov.

2.1 Brukere

Teknisk bruker av applikasjonen er da team for kvalitetsregister ved HN-IKT.

Sluttbrukerene er her følgende kvalitetsregister: * Norsk gynekologisk endoskopiregister * Norsk register for invasiv kardiologi * Norsk kvalitetsregister for behandling av spiseforstyrrelser * Norsk register for arvelige og medfødte nevromuskulære sykdommer * Register for Hidradenitis suppurativa * Norsk register for analinkontinens * Norsk register for gastrokirurgi * Norsk kvalitetsregister for endokarditt * Nasjonalt kvalitetsregister for ryggkirurgi

2.2 Brukermiljø

Applikasjonen blir en intern del av kvalitetsregister mikrotjeneste platformen, da den i hovedsak erstatter det som i dag er en tungvindt og manuell prosess. Dette vil igjen påvirke brukerne av kvalitetsregistrene som vil oppleve at kvalitetsregistrene stadig er oppdatert med nyeste versjoner av kodeverkene.

Sluttbrukerene vil ikke oppleve endringer på bakgrunn av dette arbeidet i daglig bruk av registeret. Den merkbare forskjellen er at det ikke vil ligge noen forsinkelse fra publisering av ny data til den er tilgjengelig i register-frontenden.

2.3 Sammendrag av brukerens behov

Behov	Prioritet	Påvirker	Dagens Løsning	Foreslått løsning
Automatisk datafangst	Høy	Alle brukere	Manuell innhenting	Hente fra eksterne kilder via web API
Versjonering	Høy	Alle brukere	Manuelt ved lagring i database	Automatisk ved lagring i database
Tilgjengeligjøre av oppdatert data	Høy	Alle brukere	Manuell prosess	Automatisk ved innhenting av ny data
Mellomlagring i registeret	Medium	Alle brukere	Lagres i MySql	Lagre in-memory. f.eks REDIS
Tilgjengeligjøre av dokumenter	Medium	Alle brukere	Lagres i ressurs mapper	Lagre versjonert i database

Produktegenskaper

3.1 Produktets funksjonelle egenskaper

- HTTP Klient som skal hente data fra eksterne kilder der det er mulig

- Fil leser som skal lese inn filer som inneholder data der datafangst via API ikke var mulig. Skal også kunne lese inn filer som lagres i binært format (f.eks PDF filer)
- Lagrings service som snakker med data inn og data ut tjenestene og lagrer/henter data i en MySql database
- Database for lagring av innhentet data
- REST API som serverer data til kvalitetsregister ved forespørsel
- Refresh service som poller storage service for endringer. Når det er registrert sendes det en melding til registeret om at det kan hente ny oppdatert data til sin in-memory database
- HTTP Klient som sender melding til kvalitetsregisteret om mulighet for oppdatering av data
- Distribueres som et container image for drift i Docker og Kubernetes miljø

3.2 Ikke funksjonelle egenskaper

Det er forventet at produktet er levert med høy kvalitet på kode, høy grad av test dekning, og godt beskrevet dokumentasjon. Det skal være lett for andre utviklere å videreutvikle produktet. Produktet skal ha et logisk konstruert API. Det må vere støtte for måling av metrics via f.eks. Prometheus.

4 Gjennomførbarhetsanalyse

4.1 Tilgjengelige ressurser

4.1.1 Utstyr Alle gruppemedlemmer har en egen PC som kjører Windows, Linux eller macOS. Det kreves ingen spesielt utstyr for gjennomføring av prosjektet.

4.1.2 Kunnskaper For å kunne effektivt planlegge og gjennomføre prosjektarbeid er det ønskelig at studenten har:

- generelle programmeringskunnskaper
- kjennskap til objektorientert programmeringsspråk, som Java eller Kotlin
- kjennskap til systemutvikling
- kjennskap til utvikling av tjenester etter REST prinsippet
- kjennskap til kvalitetssikring av programkode ved hjelp av enhetstester
- kjennskap til bruk av agile utviklingsmetodikker
- kjennskap til bruk av Jira Software
- kjennskap til bruk av Git og GitHub

4.1.3 Litteratur Det er planlagt å bruke ulike informasjonskilder slik som bøker, internettkilder, interne dokumenter og rapporter osv. For å kunne finne inspirasjon eller mer elegante løsninger kan det være aktuelt å bruke YouTube, Stackoverflow og informasjon fra forskjellige kurser/plattformer.

4.2 Riskanalyse

4.2.1 Bakgrunn for risikoanalyse I starten av hvert prosjekt er det viktig å vurdere mulige risikoer som kan oppstå underveis. Godt utført risikoanalyse gir prosjektgruppe en bedre oversikt over problemer, som kan oppstå under prosjektgjennomføringen. Dette gir mulighet til å iverksette risikoreduserende tiltak i tide og minimere risikoen for å ikke fullføre prosjektet (Brustad, 2022).

Det ble laget en risikotabell som viser en oversikt over mulige uønskede hendelser. Sannsynlighet og konsekvens vurderes for hver enkelt hendelse. Ved hjelp av en enkel formel beregnes det grad av risiko for gjennomføring av prosjektet. Ut fra resultater kan man vurdere behov for iverksetting av risikoreduserende tiltak (Brustad, 2022).

Begreper

Uønsket hendelse er en hendelse som kan føre til tap av verdier (Midtgaard, 2021).

Risiko er en kombinasjon av sannsynlighet og konsekvens for en uønsket hendelse (Midtgaard, 2021).

Sannsynlighet viser i hvilken grad er det trolig at en bestemt hendelse vil inntreffe (Midtgaard, 2021).

Konsekvens viser mulig følge av en bestemt hendelse (Midtgaard, 2021).

Sannsynlighet	Beskrivelse
1	veldig lite sannsynlig
2	mindre sannsynlig
3	sannsynlig
4	meget sannsynlig
5	svært sannsynlig

Konsekvens	Beskrivelse
1	ikke farlig
2	farlig
3	kritisk
4	meget kritisk
5	katastrofalt

Beregning av risikonivå

$\text{Risiko} = \text{Sannsynlighet} * \text{Konsekvens}$

(Brustad, 2022)

4.2.2 Risikoanalyse

Tabell Risikoanalyse representerer uønskede hendelser, grad av sannsynlighet, grad av konsekvens, beregnet risikonivå og mulige tiltak (Brustad, 2022).

Nr.	Beskrivelse av hendelsen	Påvirker	Sannsynlighet	Konsekvens	Risikonivå	Forebyggende tiltak	Korrigerende tiltak
1	Prosjektet ble ikke startet innen den gitte tidsrammen	Prosjekt	2	2	4	Oppstartsmøte. Velge kommunikasjonsverktøy. Etablere arbeidsplan og -rutiner.	Ved behov sette litt ekstra tid for å starte å jobbe med prosjektet i tide.
2	Prosjektet ble ikke fullført innen den gitte tidsrammen	Prosjekt	3	4	12	Aktivt bruk av smidige utviklingsmetoder i løpet av hele prosjektperioden. Ha kontroll over utførelsen av ulike aktiviteter. God kommunikasjon mellom gruppe-medlemmer og veiledere.	Ta kontakt med oppdragsgiver og veiledere for å justere arbeidsplan. Diskutere mulighet for gjennomføring av manglende aktiviteter i et annet prosjekt.
3	Sykdom eller skade i en gruppe	Person / Prosjekt	4	2	8	Unngå alvorlig sykdom og redusere aktiviteter, som kan føre til skade.	De andre gruppe-medlemmer hjelper ved behov

Nr.	Beskrivelse av hendelsen	Påvirker	Sannsynlighet	Konsekvens	Risikonivå	Forebyggende tiltak	Korrigerende tiltak
4	Dobbeltarbeid	Person / Pros- jekt	2	2	4	God kommunikasjon mellom gruppemedlemmer. Utarbeide og følge arbeidsplan. Aktivt bruk av Jira Software (Scrum).	Gjøre nødvendige justeringer i arbeidsplanen. Utføre de kritiske aktivitetene. Omprioritering av aktiviteter ved behov.
5	Dårlig arbeidsmiljø i en gruppe / Konflikter	Person / Pros- jekt	1	4	4	Fordeling av roller og ansvarsområder i en gruppe. God kommunikasjon for å forebygge mulige konfliktsituasjoner.	Ved oppstått misforståelse prøve å finne årsaken til dette. Etterpå finne en generell løsning til problemet. For eksempel, gjøre endringer i ansvarsområdet, justere arbeidsmengde osv.
6	Tap av rapport, kildekode, dokumentasjon ol.	Produkt / Pros- jekt	2	5	10	Dokumentasjon og kildekode skal lagres både lokalt hos alle gruppemedlemmer og på de valgte skytjenester som GitHub, Overleaf osv.	Prøve å gjenopprette tapt dokumentasjon og kildekode. Ved behov starte på nytt fra siste checkpoint.
7	Ønsket funksjonalitet blir ikke implementert	Produkt / Pros- jekt	2	4	8	Under planlegging av prosjektarbeid bør gruppe sette realistiske mål over funksjonalitet som skal bli implementert i applikasjonen.	Gjøre endringer i prosjektplanen. Prøve å finne alternative løsninger. Prioritering av oppgaver.

4.2.3 Risikomatrise Risiko kan fremstilles ved hjelp av en risikomatrise, der sannsynlighet er plassert langs x – akse og konsekvens langs y – akse. Risikomatrise er delt inn i grønt, gul og rødt farge, som representerer risikonivå for en bestemt hendelse i risikotabellen. Tabellen risikoakseptkriterier viser behov for igangsettelse av risikoreduserende tiltak i forhold til risikonivå.

Figur: Risikomatrise (Danilina, Daria, 2023)

Nivå	Tiltak
Lavt	Aksepteres.
Middels	Aksepteres. Kontinuerlig utførelse av risikoreduserende tiltak.
Høyt	Aksepteres ikke. Risikoreduserende tiltak skal utføres umiddelbart.

Tabell: Risikoakseptkriterier

		Sannsynlighet				
Konsekvens		1	2	3	4	5
	1				5	
	2		1,4		7	6
	3				2	
	4		3			
	5					

Figure 1: Scrum

4.3 Drøfting og vurdering av ulike løsningsmetoder og analyse av forslag

4.3.1 Programmeringsverktøy / Utviklingsverktøy Det finnes mange forskjellige programmeringsverktøy. Følgende er beskrivelse av teknologier som er aktuelt i vårt prosjekt.

Programmeringsspråk

Prosjektgruppe har valgt å bruke Kotlin som programmeringsspråk. Grunnen til dette er at applikasjonen skal lages ved hjelp av Ktor rammeverket. Og det er bare Kotlin som kan bli brukt til dette formålet. Dersom prosjektgruppen velger et annet rammeverk, for eksempel Spring Boot, kan Java bli et godt alternativ.

Kotlin er et moderne programmeringsspråk. Det er forbedret versjon av Java, som er enklere i bruk, raskere å kompilere og inneholder en blanding av funksjonell og objektorientert programmering. Det har også andre fordeler, som clean code, null safety, extension functions, coroutines support osv (Marianna Berga, 2021[cited 2023Jan23]).

Web rammeverk

Det finnes flere rammeverk som brukes til å skrive applikasjoner i Kotlin. Prosjektgruppe har bestemt seg for å utvikle applikasjonen som REST tjeneste med Ktor rammeverket av JetBrains.

Ktor er et asynkront mikrorammeverk som egner seg godt til utvikling av små applikasjoner i Kotlin. Både Kotlin, Ktor og IntelliJ IDEA er produkter laget av JetBrains. Dette gir et utmerket verktøystøtte for prosjekter. Det kreves ikke så mye for å lage Ktor applikasjon, legge til ønsket funksjonalitet, endre konfigurasjon osv (JetBrains, 2023[cited 2023Jan23]a).

Ktor er lett og fleksibelt rammeverk som lar man utvikle og strukturere applikasjonen slik en trenger. Ved behov kan man enkelt utvide applikasjonens funksjonalitet ved hjelp av plugins (JetBrains, 2023[cited 2023Jan23]b).

Kotlin er det eneste programmeringsspråk som kan brukes for utvikling av applikasjoner ved hjelp av Ktor rammeverket.

Spring Boot er fortsatt ledende rammeverk for utvikling av applikasjoner i Java. Den er godt dokumentert og er mye brukt blant utviklere. Dersom prosjektgruppe hadde valgt Java som programmeringsspråk, kunne Spring Boot være et godt alternativ til Ktor.

IDE (Integrated Development Environment)

Prosjektgruppen er godt kjent med IntelliJ IDEA siden den ble mye brukt i løpet av studiet. Samtidig skal applikasjonen utvikles i Kotlin, og oppdragsgiveren prioriterer dette utviklingsmiljø også. Derfor ble den valgt for prosjektet vårt.

IntelliJ IDEA (Community Edition) er open source IDE, som egner seg best til utvikling av Kotlin applikasjoner. IntelliJ IDEA er utviklet og vedlikeholdt av JetBrains, har flere funksjoner og er godt egnet for rask programvareutvikling. Den har også gode

muligheter for integrasjon med andre typer verktøy som Git og GitHub, Gradle og Docker Compose. Siden det er ikke mulig å utvikle Kotlin prosjekter i en annen IDE, finnes det ikke et bedre alternativ til IntelliJ IDEA (JetBrains, 2023[cited 2023Jan23]c).

Dev/deployment env

Docker Containere vs Virtual Machines

Prosjektgruppe har en god erfaring med bruk av Docker og Docker Compose, og derfor ble den valgt for prosjektet vårt.

Docker er et verktøy som brukes til utvikling, testing og kjøring av applikasjoner i Docker containere. En container inneholder alle nødvendige biblioteker, konfigurasjonsfiler og avhengigheter som kreves for å kunne kjøre applikasjonen (Arora, 2022[cited 2023Jan23]).

Docker Compose er et verktøy som brukes til å kjøre flere containere samtidig (kode24, 2021[cited 2023Jan23]).

Det er flere fordeler ved bruk av Docker i programvareutvikling. Docker kan installeres på alle moderne operative systemer, har kort oppstartstid, lav innvirkning på OS og bruker ikke så mye plass på harddisken (Docker, 2023[cited 2023Jan23]).

Docker containere har høy portabilitet og derfor kan bli enkelt flyttet og distribuert mellom ulike maskiner, og med dette forenkler deling av kode (Schmitt, 2022[cited 2023Jan23]).

Som et alternativ kan man vurdere bruk av virtuelle maskiner. I motsetning til Docker, det er flere ulemper enn fordeler med disse. Virtuelle maskiner har større innvirkning på OS, tregere og tar mye diskplass (Docker, 2023[cited 2023Jan23]).

Virtuelle maskiner innkapsler en hel maskin, mens Docker Containere innkapsler bare applikasjonen og utviklingsmiljø. Dette gjør deling, ombygning og distribusjon av kode utfordrende. Men virtuelle maskiner har mye bedre sikkerhet enn Docker Containere (Schmitt, 2022[cited 2023Jan23]).

I enkelte prosjekter kan det være lurt å kombinere bruk av virtuelle maskiner og Docker Containere (Schmitt, 2022[cited 2023Jan23]).

Relasjonsdatabase

Det finnes mange forskjellige typer databaser, både NoSQL og SQL. Oppdragsgiveren prioriterer bruk av MySQL database. Derfor ble den valgt for prosjektet vårt.

4.3.2 Prosjektstyringsverktøy Microsoft Teams

Alle gruppemedlemmer er godt kjent med Microsoft Teams. Derfor ble den valgt som kommunikasjonsverktøy for hele prosjektperioden. Det ble etablert egen Teams gruppe. Microsoft Teams støtter organisering av møter via chat, videochat, lagring av filer og integrasjon med enkelte applikasjoner (Microsoft, 2023[cited 2023Jan23]).

Jira Software vs Trello

Både Jira og Trello er populære agile prosjektstyringsverktøy. Begge to eies av Atlassian. Hver plattform har distinkte fordeler og ulemper og kan bli brukt i ulike tilfeller. Følgende er beskrivelse av hovedforskjeller ved disse to plattformer.

Jira er godt egnet til styring av prosjekter av type programvareutvikling. Den har flere smidige funksjoner, som kan brukes under prosjektarbeid. Blant disse er mulighet til å lage Sprints, Epics og User Stories, bruk av Markup language og labels, mulighet til å legge til filer og koble prosjektet til repositories (Atlassian, 2023[cited 2023Jan23]).

Jira Software støtter både Scrum og Kanban prosjekter, og er det mest populære verktøyet for Scrum Teams. Scrum Board er delt inn i tre hoveddeler: TO DO, IN PROGRESS og DONE. Denne inndelingen gir godt oversikt over prosjektets arbeidsflyt og bidrar til bedre kommunikasjon mellom gruppemedlemmer og fordeling av arbeidsoppgaver (Atlassian, 2023[cited 2023Jan23]).

Trello er et enkelt verktøy som er basert på Kanban board (Trello., 2023[cited 2023Jan23]). Den kan bli brukt for å visualisere ulike arbeidsflyt, men er ikke nok til å støtte store kompliserte prosjekter av type programvareutvikling (Jakes, 2022[cited 2023Jan23]).

Git og GitHub vs BitBucket

Alle gruppemedlemmer har en god erfaring med bruk av Git og GitHub siden det ble mye brukt i løpet av studiet. Derfor ble Git valgt som versjonskontrollsystem og GitHub som hosting for prosjektet vårt. En av de andre alternativene til lagring av kildekode er Bitbucket.

GitHub er fortsatt den største hosting for Git repository, og betjener et stort antall programvareutviklere med åpen kildekode. GitHub egner seg godt for store prosjekter, og har mulighet til å lage både offentlige og private repositorier (manmeetjuneja5, 2022[cited 2023Jan23]).

GitHub støtter prosjekter som bruker Git versjonskontrollsystem, og lar utviklere å jobbe sammen om et prosjekt. Den inneholder et stort antall funksjoner som gjør prosjektarbeid enklere. For eksempel, Github støtter Markdown og kan brukes til skriving av rapporter og annen dokumentasjon. GitHub kan også integreres med Jira som er et viktig verktøy for styring av prosjektarbeid (manmeetjuneja5, 2022[cited 2023Jan23]).

Bitbucket er en annen hosting som blir mer og mer populært blant utviklere. I utgangspunktet Bitbucket er fokusert på små prosjekter og lukket kildekode. Den støtter Git og Mercurial. Bitbucket integreres godt med Jira og Trello siden begge produktene er fra Atlassian (manmeetjuneja5, 2022[cited 2023Jan23]).

4.3.3 Utviklingsmetoder / metodikker Det finnes to populære agile utviklingsmetoder: Scrum og Kanban. Begge to har lignende prinsipper og brukes til styring av prosjektarbeid.

Scrum

Scrum er en metode som egner seg godt for store utviklingsprosjekter. Den består av tre deler: roller, hendelser og artefakter. Prosjektgruppe er relativt stor og har rollefordeling: Project Owner, Scrum Master og Developers. Hver eneste person i prosjektgruppe har sitt eget ansvarsområde. Utviklingsprosess er delt inn i korte perioder med en fast lengde, sprinter. Hver Sprint består av Sprint Planning, Sprint med Daily Scrum, Sprint Review og Sprint Retrospective. Hver del av sprinten har sitt eget mål og utførelsesteknikk. I løpet av sprinten utarbeides det såkalte artefakter: Produkt Backlog, Sprint Backlog og Product Increment (Rehkopf, 2023[cited 2023Jan23]).

Kanban

Kanban er en annen utviklingsmetode. I motsetning til Scrum, egner den seg godt for mindre prosjektgrupper. Kanban har ingen rollefordeling. Den er ikke inndelt i sprinter med fast lengde, og derfor mer fleksibelt enn Scrum. Kanban brukes til å visualisere kontinuerlig arbeidsflyt (Rehkopf, 2023[cited 2023Jan23]).

Scrumban

Prosjektgruppe har bestemt seg til å bruke det beste av begge metodene og valgt Scrumban, som er en hybrid av Scrum og Kanban. I likhet med Scrum, har Scrumban iterasjoner, der prosjektgruppe jobber med prioriterte oppgaver fra Backlog. Scrumban er mye mer fleksibelt enn Scrum, og Backloggen kan utvides med nye elementer i løpet av iterasjonen. Alle møtene er «on demand» og holdes når prosjektgruppen ser behov for det. Scrumban krever ikke så mye dokumentasjon enn Scrum, og dermed sparer tid som kan bli brukt til utvikling av prosjektet (Dhiman, 2022[cited 2023Jan23]).

Prosjektgruppe bruker Kanban board for visualisering av arbeidsflyt og setter en grense for antall elementer som kan være i «in process» tilstand. Dette hjelper å finne «svakheter» i gruppens arbeid, øker effektiviteten av prosjektgruppe og reduserer sjansene for å ikke fullføre planlagt arbeid (Radigan, 2023[cited 2023Jan23]).

4.3.4 Dokumentasjon Alt prosjektarbeid skal bli godt dokumentert. Gjennom hele prosjektperioden skal det lages forskjellige rapporter, grafer og diagrammer.

Markdown vs Microsoft Word, Endnote

Det finnes flere måter å skrive en rapport på. Tradisjonelt ble Microsoft Word og Endnote brukt til dette formål i løpet av studiet. Prosjektgruppe har valgt å bruke Markdown til å skrive rapporter og andre dokumenter i forbindelse med gjennomføring av prosjektet. En av grunnene er at GitHub som brukes til lagring av kildekode, støtter skriving og formattering av tekst i Markdown.

Selv om bruk av Markdown kan se vanskeligere for nybegynnere, den er stort sett mye bedre enn Microsoft Word. Først gjelder det dokumentdeling, der flere medlemmer kan skrive rapport samtidig. Det er mulig å lage en ny .md dokument i GitHub og dele den med de andre deltakere.

Ved å kombinere vanlig tekst med Markup symboler kan man style og strukturere teksten, legge til bilder, referanser, lage lister og tabeller mm. Det kan ta litt tid til å bli kjent med disse kommandoer, men det er absolutt verdt det.

Det er også mulig å nevne en bestemt person eller en gruppe på GitHub ved å skrive @ foran navnet. Ved hjelp av Git kan man spore endringer i dokumentet eller gå til en tidligere versjon (Docs, 2023[cited 2023Jan23]).

Microsoft Word er mye enklere i bruk og har bedre sjekk for grammatiske- og stavefeil enn Markdown, GitHub. Derfor er det gunstig å sjekke teksten i Microsoft Word før den skal overføres til GitHub. I tillegg til Microsoft Word kan man bruke Endnote for å styre referanser (EndNote, 2023[cited 2023Jan23]).

Mermaid vs VisualParadigm

Prosjektgruppe skal hovedsakelig bruke Mermaid for å lage forskjellige grafer og diagrammer. Mermaid er et JavaScript-basert verktøy, som bruker enkle kommandoer for å lage kompliserte diagrammer i Markdown og endre dem dynamisk. Ved hjelp av Mermaid kan man lage flowchart, sequence diagram, class diagram, gantt diagram, pie chart diagram, entity relationship diagram osv. Mermaid er integrert i GitHub, og det er mulig å lage grafer i GitHub's README filer som kan bli aktuelle i vårt prosjekt. Det finnes også Mermaid plugins for mange andre tjenester, for eksempel Visual Studio Code og IntelliJ IDEA (Mermaid, 2023[cited 2023Jan23]).

VisualParadigm er en alternativ type verktøy som kan bli brukt til å lage ulike diagrammer. Ved hjelp av VisualParadigm kan man lage flowchart, UML class diagrams, use case diagrams, entity relationship diagrams, grafisk fremstilling av arbeidsflyten osv. Den inneholder en stor samling av ulike maler, som kunne bli brukt i prosjektet. Men ikke alle funksjonene er gratis, og den er heller ikke koblet til GitHub. Derfor vil bruk av Mermaid prioriteres (VisualParadigm, 2023[cited 2023Jan23]).

Jira Software

Styring av prosjektarbeid skal gjennomføres ved hjelp av Jira Software. Prosjektgruppe skal lage et Kanban prosjekt, som bruker Roadmap og Kanban board for visualisering av arbeidsflyt. Jira Software skal også brukes til generering av diverse rapporter og diagrammer som dokumenterer prosjektarbeid (Atlassian, 2023[cited 2023Jan23]).

Microsoft Excel

Microsoft Excel skal brukes til å lage og føre projektdagbok som gir en oversikt over brukt tid for hele prosjektet og forskjellige delaktiviteter samt viser detaljert beskrivelse av utført arbeid. Excel kan også benyttes til grafisk representering, for eksempel i form av sektordiagram.

4.4 Valg av løsning og utviklingsmetodikk

Utvikling

- Programmeringsspråk: Kotlin
- IDE: IntelliJ IDEA
- Rammeverk: Ktor med Arrow
- Container: Docker / Docker Compose
- Relasjonsdatabase: MySQL

Prosjektstyring

- Kommunikasjonsverktøy: Microsoft Teams
- Prosjektstyringsverktøy: Jira Software
- Versjonskontrollverktøy: Git og Github

Dokumentasjon

- Rapportskriving: Markdown / Overleaf, LaTeX
- Grafer og diagrammer: Mermaid, Jira Software, Excel

Prosjektutvikling

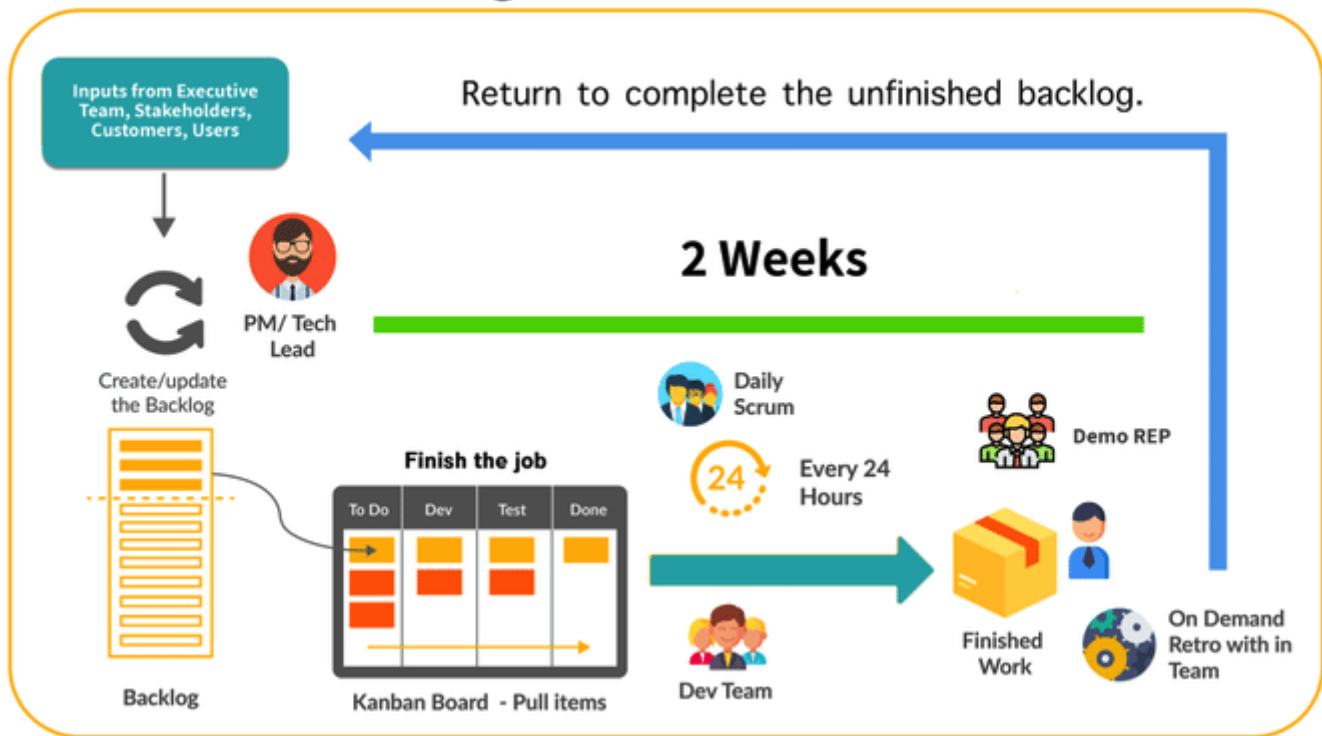
- Utviklingsmetode: Scrumban

6 Prosjektplan

6.1 Utviklingsmetode

Prosjektgruppe har valgt Scrumban som utviklingsmetode. Scrumban er en hybrid av Scrum og Kanban som benytter Backlog for planlegging, prioritering og fordeling av arbeidsoppgaver og Kanban board for visualisering av prosjektets framgang [6-scrumban-overview].

Agile Scrumban .



Figur: Beskrivelse [6-scrumban-overview].

Scrumban er iterativ prosess som kan deles inn i korte perioder. Lengde av hver periode kan variere fra en til fire uker og bestemmes av prosjektgruppen. Vanligvis tar en periode cirka 2 uker [6-scrum-vs-kanban-vs-scrumban].

Som en del av Scrumban prosess etableres det en Backlog. Den inneholder en liste over elementer som kan bli utviklet i løpet av prosjektperioden. Antall elementer i Backlogen varierer, og nye elementer kan bli lagt til når som helst [6-scrum-vs-kanban-vs-scrumban].

Elementene har prioritering og WIP begrensning. WIP står for «work-in-progress limit» og setter en grense for antall oppgaver som skal plasseres i «in progress» kolonne av Kanban board. Elementene med høyest prioritet skal utføres først. Estimering av elementer er valgfritt [6-scrum-vs-kanban-vs-scrumban].



Figur: Besrivelse [4-logrocket.com-scrumban].

Kanban board brukes til visualisering av kontinuerlig arbeidsflyt. Elementene fra Backloggen plasseres og tas av gruppemedlemmer etter pull prinsippet. Etter at man er ferdig med sin oppgave, tar en ny fra tavlen [6-scrum-vs-kanban-vs-scrumban].

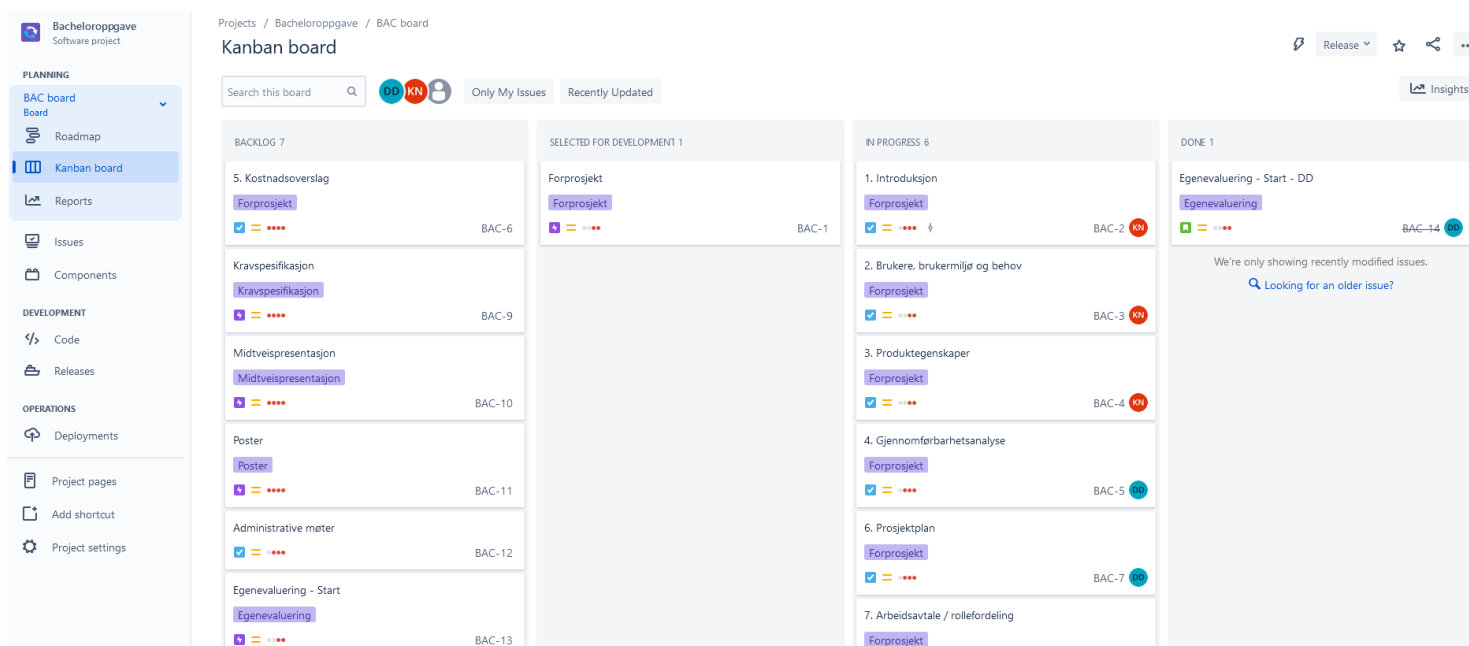
Prosjektgruppe skal møtes hver dag for korte standup møter. Målet med disse møtene er å gå gjennom gjennomført og pågående arbeid, se gjennom en liste med prioriterte oppgaver og finne løsninger til eventuelle problemer [4-logrocket.com-scrumban].

På slutten av iterasjonen, samles prosjektgruppe for å diskutere gjennomført arbeid, implementert funksjonalitet og eventuelle mangler, tilleggskrav fra kunden, utviklingsprosess og nødvendig forbedringstiltak som vil gjøre prosjektgruppe mer produktiv neste iterasjon. Etter det starter prosjektgruppe planlegging av en ny iterasjon, justering og prioritering av elementer i Backloggen [6-agilealliance-scrumban].

6.2 Jira Software

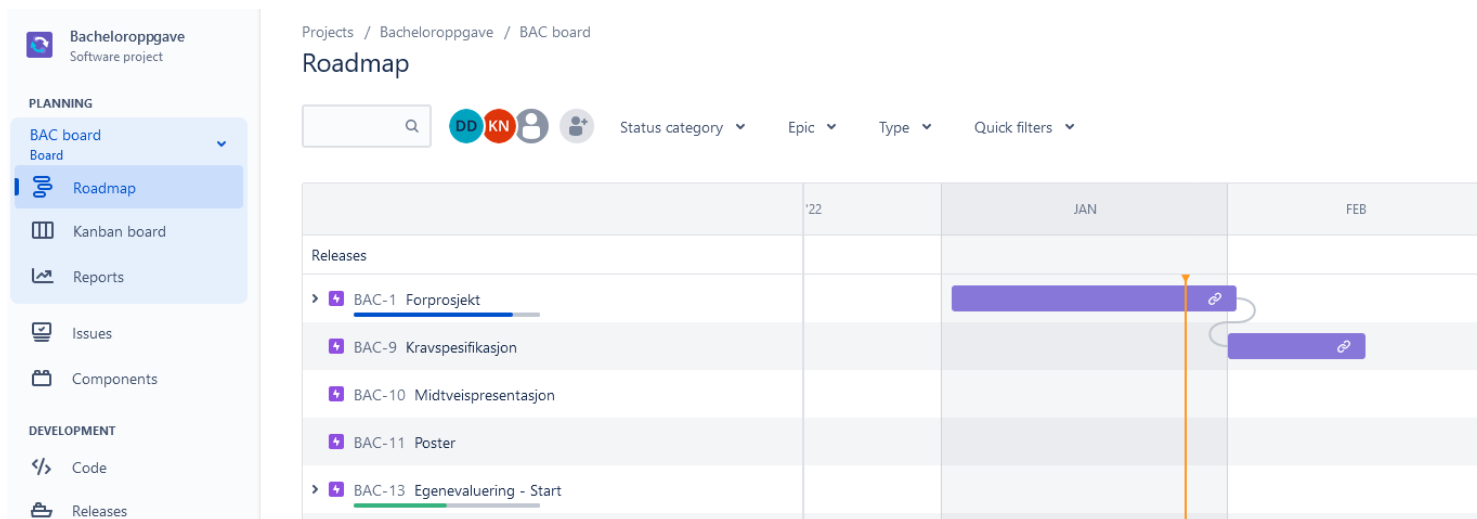
Jira Software er et prosjektstyringsverktøy som skal brukes til å støtte Scrumban prosess. Prosjektet lages ved hjelp av Kanban Project Template. Ved å velge Team-Managed Template, kan man gjøre styring av prosjektet mer fleksibel. Det finnes også mulighet til å koble GitHub repository til prosjektet (Atlassian, 2023[cited 2023Jan23]). Kanban board brukes til styring av prosjektarbeid. Den er delt inn i tre kolonner: TO DO, IN PROGRESS og DONE. Det er mulig å endre navn eller legge til ekstra kolonner, for eksempel BACKLOG (Atlassian, 2023[cited 2023Jan23]). Elementene fra Backloggen plasseres på tavla. Kanban prosjekt har tre hovedtyper elementer: epics, tasks og subtasks. Epics brukes til gruppering av elementer og kan bli koblet sammen ved hjelp av avhengigheter. Tasks er en primær type element og kan bli delt inn i mindre subtasks (Atlassian, 2023[cited 2023Jan23]).

Prosjektgruppe styrer sitt arbeid ved å ta og flytte arbeidsoppgaver på forskjellige steder av Kanban board. Oppgavene skal utføres etter høyest prioritet.



Figur: Kanban Board, Jira Software (Atlassian, 2023[cited 2023Jan23]).

Kanban roadmap gir en oversikt over alle Epics og viser hvor lang tid en bestemt Epic tar i løpet av utviklingsprosessen (Atlassian, 2023[cited 2023Jan23]).



Figur: Kanban Roadmap, Jira Software (Atlassian, 2023[cited 2023Jan23]).

Jira Software gir også mulighet til å lage diverse rapporter og grafer (Average Cycle Time), som blir til en stor hjelp ved ferdigstilling av nødvendig dokumentasjon (Atlassian, 2023[cited 2023Jan23]).

7 Arbeidsavtale / Rollefordeling

7.1 Samarbeidsavtale

Samarbeidsavtale (Vedlegg 1).

7.2 Organisering av ansvar

7.2.1 Rollefordeling / Prosjektorganisering Prosjektgruppe skal bruke Scrumban utviklingsmetode. Oppdragsgiver er Helse Nord IKT. Torkil Grindstein er en veileder fra Helse Nord IKT og er et viktig bindeledd mellom oppdragsgiveren og prosjektgruppe. Gruppen består av Kristian Nedrevold og Daria Danilina som skal jobbe sammen for å levere best mulig produkt til oppdragsgiver og utarbeide all nødvendig dokumentasjon.

7.2.2 Ansvarsfordeling Prosjektgruppe består av to grupped medlemmer. Begge to skal aktivt delta i planlegging av prosjektarbeid, utviklingen av selve systemet og ferdigstilling av nødvendig dokumentasjon.

Kristian Nedrevold er prosjektleder og kontaktperson for gruppen. Han er ansvarlig for å ta kontakt med eksterne veiledere (Helse Nord IKT) og styring av møter. Han har også fått en rolle som utviklingsleder, og er ansvarlig for styring av utviklingsprosessen og fordeling av arbeidsoppgaver.

Daria Danilina er dokumentansvarlig, og skal sikre at all nødvendig dokumentasjon er oppdatert og organisert på en riktig måte. Hun har også fått en rolle som sekretær, og er ansvarlig for å ta møtereferater.

Alle grupped medlemmer skal fungere som utviklere gjennom hele prosjektperioden. De skal også sørge for at arbeidet med prosjektet går etter plan og finne mulige løsninger til oppståtte problemer. De skal jobbe sammen med ferdigstilling av nødvendig dokumentasjon, rapporter og diagrammer.

Alle grupped medlemmer skal delta i interne møter som skal avholdes via Microsoft Teams. Det er satt minst ett møte i uka. I tillegg skal det gjennomføres korte daglige møter for å kontrollere at arbeidet går som planlagt samt løse eventuelle problemer.

7.3 Signering av avtalen

Samarbeidsavtalen er signert [Dato].

Arora, S. (2022[cited 2023Jan23]) 'Docker vs. Virtual machines: Differences you should know [internet].' Available at: <https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know/>.

Atlassian (2023[cited 2023Jan23]) 'Jira software [internet].' Available at: <https://www.atlassian.com/software/jira>.

Brustad, T.F. (2022) *Risikoanalyse*. UiT Norges arktiske universitet.

Dhiman, M. (2022[cited 2023Jan23]) 'What is the scrumban methodology and how to implement it? [internet].' Available at: <https://blog.logrocket.com/product-management/what-is-scrumban-methodology-how-to-implement/#core-principles-of-scrumban/>.

Docker (2023[cited 2023Jan23]) 'Use containers to build, share and run your applications [internet].' Available at: <https://www.docker.com/resources/what-container/>.

Docs, G. (2023[cited 2023Jan23]) 'Quickstart for writing on GitHub [internet].' Available at: <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/quickstart-for-writing-on-github/>.

EndNote (2023[cited 2023Jan23]) 'Accelerate your research [internet].' Available at: <https://endnote.com/>.

Jakes, D. (2022[cited 2023Jan23]) 'Trello vs. Jira: Top agile project management tool. [internet].' Available at: https://technologyadvice.com/blog/information-technology/trello-vs-jira-choosing-an-agile-project-management-tool/#trello_vs_jira_which_is_right_for_your_business/.

JetBrains (2023[cited 2023Jan23]a) 'IntelliJ IDEA ultimate support [internet].' Available at: <https://ktor.io/idea/>.

JetBrains (2023[cited 2023Jan23]b) 'Simple and fun. [internet].' Available at: <https://ktor.io/>.

JetBrains (2023[cited 2023Jan23]c) 'What is IntelliJ IDEA? [internet].' Available at: <https://www.jetbrains.com/idea/features/>.

kode24 (2021[cited 2023Jan23]) 'Hva er greia med docker? [internet].' Available at: <https://www.kode24.no/artikkel/hva-er-greia-med-docker/74020891>.

kvalitetsregistre, N. servicemiljø for medisinske (2023[cited 2023Jan23]) 'Norsk gynekologisk endoskopiregister [internet].' Available at: <https://www.kvalitetsregistre.no/register/gynekologi/norsk-gynekologisk-endoskopiregister/>.

manmeetjuneja5 (2022[cited 2023Jan23]) 'Difference between bitbucket and GitHub [internet].' Available at: <https://www.geeksforgeeks.org/difference-between-bitbucket-and-github/>.

Marianna Berga, R.F. (2021[cited 2023Jan23]) 'Kotlin vs java: The 12 differences you should know. [internet].' Available at: <https://www.imaginarycloud.com/blog/kotlin-vs-java/>.

Mermaid (2023[cited 2023Jan23]) 'Mermaid diagramming and charting tool [internet].' Available at: <https://mermaid.js.org/#/>.

Microsoft (2023[cited 2023Jan23]) 'Microsoft teams [internet].' Available at: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software/>.

Midtgaard, A.K. (2021) *NS 5814:2021. Krav til risikovurderinger*.

Radigan, D. (2023[cited 2023Jan23]) 'What are WIP limits? [internet].' Available at: <https://www.atlassian.com/agile/kanban/wip-limits/>.

Rehkopf, M. (2023[cited 2023Jan23]) 'Kanban vs. Scrum: Which agile are you? [internet].' Available at: <https://www.atlassian.com/agile/kanban/kanban-vs-scrum/>.

Schmitt, J. (2022[cited 2023Jan23]) 'Containers vs virtual machines: What is the difference? [internet].' Available at: <https://circleci.com/blog/containers-vs-virtual-machines/>.

Trello., Atlassian. (2023[cited 2023Jan23]) 'Trello brings all your tasks, teammates, and tools together [internet].' Available at: <https://trello.com/home>.

VisualParadigm (2023[cited 2023Jan23]) 'Infographic maker [internet].' Available at: <https://online.visual-paradigm.com/>.