



UiT Norges arktiske universitet

Fakultet for Ingeniørvitenskap og Teknologi
Institutt for Datateknologi og Beregningsorientert Ingeniørfag

REST API kodeverk

Nedrevold, Kristian
Danilina, Daria

Forprosjektrapport, 2023-Feb-01

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter(e)
01/feb/23	1.1	Firste utkast.	Kristian Nedrevold, Daria Danilina
/feb/23	1.2	Andre utkast. Etter tilbakemeldinger fra veiledere.	Kristian Nedrevold, Daria Danilina

Innholdsfortegnelse

Revisjonshistorie	i
Innholdsfortegnelse	iii
1 Introduksjon	1
1.1 Bakgrunn for oppgaven	2
1.2 Prosjektbeskrivelse og analyse	3
1.3 Rammebetingelser	5
2 Brukere, brukermiljø og behov	6
2.1 Brukere	6
2.2 Brukermiljø	6
2.3 Sammendrag av brukerens behov	7
3 Produktegenskaper	8
3.1 Produktets funksjonelle egenskaper	8
3.2 Ikke funksjonelle egenskaper	8
4 Gjennomførbarhetsanalyse	10
4.1 Tilgjengelige ressurser	10
4.1.1 Utstyr	10
4.1.2 Kunnskaper	10
4.1.3 Litteratur	10
4.2 Risikoanalyse	11
4.2.1 Bakgrunn for risikoanalyse	11
4.2.2 Risikoanalyse	12
4.2.3 Risikomatrise	14
4.3 Drøfting og vurdering av ulike løsningsmetoder og analyse av forslag	15
4.3.1 Programmering verktøy / Utviklingsverktøy	15
4.3.2 Prosjektstyring verktøy	16
4.3.3 Utviklingsmetoder / metodikker	17
4.3.4 Dokumentasjon	18
4.4 Valg av løsning og utviklingsmetodikk	20
5 Kostnadsoverslag	21
6 Prosjektplan	22
6.1 Utviklingsmetode	22
6.2 Jira Software	23
7 Arbeidsavtale, rollefordeling	25
7.1 Samarbeidsavtale	25
7.2 Organisering av ansvar	25
7.2.1 Rollefordeling / Prosjektorganisering	25
7.2.2 Ansvarsfordeling	25
7.3 Signering av avtalen	25
8 Referanser	26
9 Liste over figurer	29

10	Liste over tabeller	30
11	Stikkordsliste	31
12	Vedlegg	32

1 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 er en av to godkjente leverandører av nasjonale medisinske kvalitetsregister. Teamet jobber med å utvikle plattform for kvalitetsregister samt de spesifikke register.

Kvalitetsregister som utvikles av HN-IKT:

- Norsk gynekologisk endoskopiregister[1]
- Norsk register for invasiv kardiologi[2]
- Norsk kvalitetsregister for behandling av spiseforstyrrelser[3]
- Norsk register for arvelige og medfødte nevromuskulære sykdommer[4]
- Norsk register for analinkontinens[5]
- Norsk register for gastrokirurgi[6]
- Norsk kvalitetsregister for endokarditt[7]
- Nasjonalt register for ablasjonsbehandling og elektrofysiologi i Norge (AblaNor)[2]
- Nasjonalt register for langtids mekanisk ventilasjon LTMV[8]
- Norsk kvalitetsregister for fedmekirurgi (SOReg-Norge)[9]
- Nasjonalt kvalitetsregister for ryggkirurgi (Paraply for tre register - Deformitet/Rygg/Nakke)[10]

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

Registeret samler inn data om [1]:

- 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 behandlende enhet

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 [1].

På denne måten er nasjonale kvalitetsregister et viktig verktøy for å sikre lik og trygg behandling for alle pasienter uavhengig av geografisk tilhørighet.

Data samlet inn via slike medisinske kvalitetsregistre danner grunnlag for analyse og forskning som har som formål å gi økt behandlings kvalitet ved norske sykehus. Innregistrering av data til medisinske kvalitetsregistre blir ofte utført av helsepersonell, ofte i en hektisk hverdag 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 kvalitets-forbedringsprosjekter. [11].

Helse Nord IKT leverer en plattform for medisinske kvalitetsregister som er basert på OpenQReg. OpenQReg er utviklet i Sverige av Uppsala Clinical Research Center.[12] 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.[13]

OpenQReg plattformen er opprinnelig utviklet av Uppsala Clinical Research Centre som en åpen basis plattform for kvalitetsregistre. 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 referer til. Postnummer er et av kode-verkene applikasjonen skal omfavne. Den skal også omfavne medisinske kodeverk. De kobler en kode opp mot en diagnose eller tilstand.

Kodeverk som brukes er:

- Postnummer
- ICD10 [14]
- NCMP [15]
- NCSP [16]

Dette er i dag opplysninger som er hard kodet inn i applikasjonens database, og som lastes inn ved oppstart av applikasjonen. Det er også andre kodeverk som er ønsket av registrene, men som ennå ikke er inkludert i 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 driftspersonell som så kjører scriptet i produktionsdatabasen. 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 versjons styre kodene. For så å levere de til register applikasjonene via et REST API. Det er altså ønskelig automatisere oppdatering 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 versjonsstyrte dokumenter inn i applikasjonen som kan lagres som binary blobs. De skal også tilgjengeliggjøres via APIet. Dette ønskes da hvert kvalitetsregister lagrer og gjør tilgjengelig et antall dokumenter for ulike formål. Det lagres også maler for meldinger som skal sendes til innbyggere. Det er da ønskelig å ha en felles portal for å oppdatere disse dokumentene.

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

Figuren under viser hvordan arbeidsflyten til en bruker av registeret er og den underliggende prosessen bak dette.

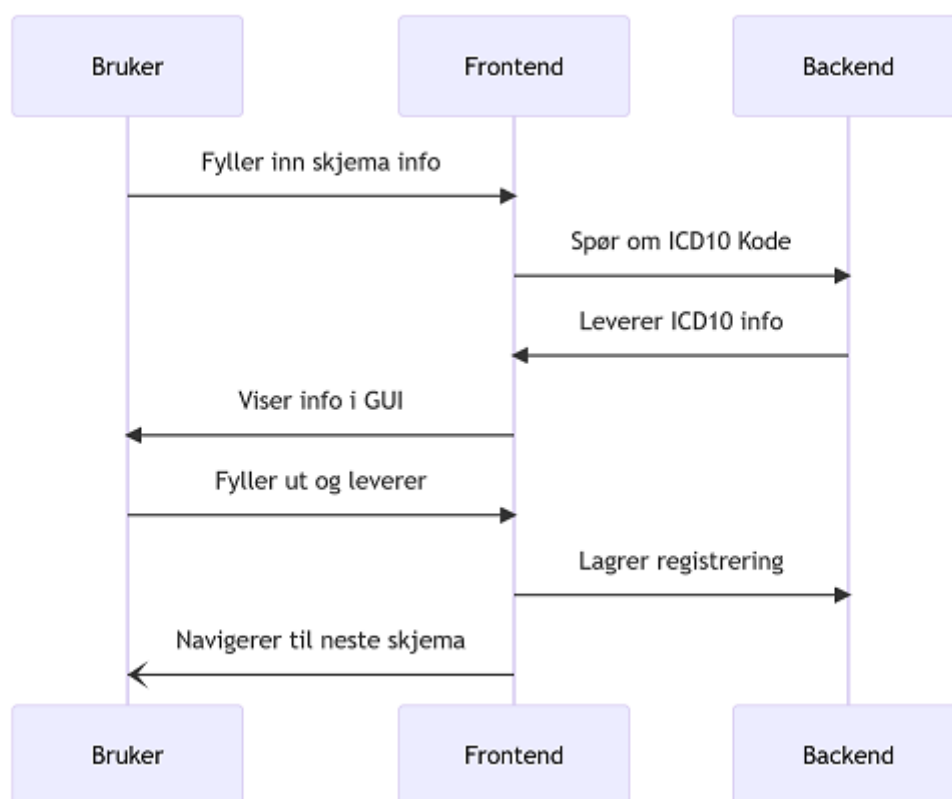


Figure 1: Sequence diagram. Generert med mermaid.

Utifra figuren kan vi se at hver gang et skjema som inneholder ICD10 koder eller andre eventuelle koder skal fylles ut vil prosessen stoppe tidlig og må ligge på vent til programvaren er oppdatert.

1.2 Prosjektbeskrivelse og analyse

Utviklingen vil foregå i to ulike kode baser. Selve tjenesten som skal hente data fra eksterne kilder og i den eksisterende register koden der HTTP klient og midlertidig lagring av koder skal etableres.

Alexandria:

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

I eksisterende register kodebase:

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

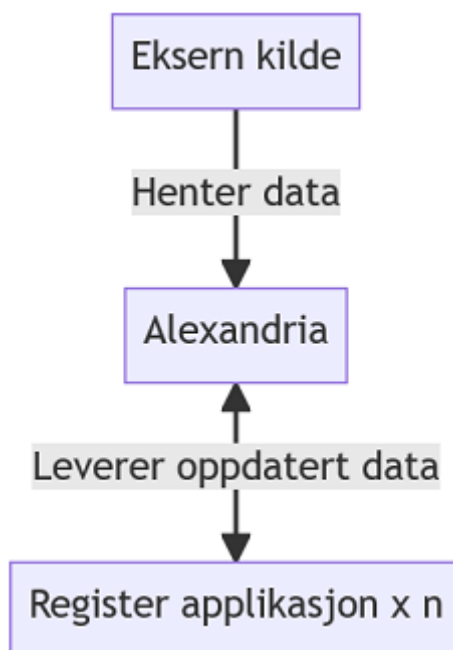


Figure 2: Overordnet dataflyt i løsningen. Generert med mermaid.

Teknologier:

Kvalitetsregister teamet bruker JVM teknologier for sin utvikling, da spesifikt Java og Kotlin, der hoved kode basen 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 migrering prosess 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

1.3 Rammebetingelser

Utvikling kan skje fra egen PC. Anbefalt å bruke IntelliJ IDEA for utvikling av Kotlin kode. Team for kvalitetsregister må gjøre tilgjengelig 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.

2 Brukere, brukermiljø og behov

2.1 Brukere

Teknisk bruker av applikasjonen er da team for kvalitetsregister ved HN-IKT. Sluttbrukere som får levert ut kodene er registrene nevnt i introduksjon.

2.2 Brukermiljø

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

Sluttbrukerne vil ikke oppleve endringer på bakgrunn av dette arbeidet i daglig bruk av registeret, i de tilfeller der kodeverk er oppdaterte.

Den merkbare forskjellen er at det ikke vil være 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
Versonstyring	Høy	Alle brukere	Manuelt ved lagring i database	Automatisk ved lagring i database
Tilgjengeligjøring 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
Tilgjengeligjøring av dokumenter	medium	Alle brukere	Lagres i ressurs mapper	Lagre versjonert i database

Table 1: Sammendrag av brukerens behov

3 Produktegenskaper

3.1 Produktets funksjonelle egenskaper

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

Hoved logikken i applikasjonen vil ligge i en Storage Service som har ansvar for å holde på data. Den vil via jevne mellomrom prøve å automatisk oppdatere data fra eksterne kilder, samt ha evne til å motta manuelt innlastede filer. Dette gjøres via respektive HTTP- og fil-klienter Den persisterer data i en database. Applikasjonen vil ha en tjeneste som gir beskjed til register applikasjonen når ny data kan hentes fra REST apiet. Beskjeden blir gitt via et HTTP kall fra en HTTP-klient.

Tjenesten som skal kjøre i applikasjonen for å utføre periodiske oppgaver vil kjøres via såkalte coroutines. En coroutine er en type virtuell tråd som kan suspenderes når den ikke jobber for så å vekkes igjen når den skal jobbe. De er et lettvekts konstrukt i Kotlin for å skrive applikasjoner som skal utføre samtidige prosesser. Med coroutines som verktøy kan vi oppnå samtidige langtids kjørende jobber som bruker minimalt med system ressurser.[17] Vi kan også benytte oss av ikke-muterende datastrukturer og metoder som kan abstrahere over de for å utføre parallelle prosesser på en trygg måte.

Http klient og lagring av koder i register kode-basen blir utviklet parallelt med applikasjonen, på denne måten kan vi vite at vi alltid kan gi og motta meldinger til registrene som skal bruke løsningen. Register kodebasen har i dag støtte for å bruke Ktor http-klient.

Figur 3 viser dataflyt i applikasjonen. Grønne ruter skal implementeres som en del av prosjektet.

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 være støtte for måling av metrics via f.eks. Prometheus. Applikasjonen må kunne settes opp med TLS for sikker kommunikasjon på helsenettet. Applikasjonen må kunne nå og kommunisere med de ulike registrene som kjører i produksjon. Det må i registeret da være mulig å sette hvilken url/ip-adresse tjenesten kjører på. Samtidig må tjenesten kunne nå de eksterne datakildene den skal snakke med for å hente data. Kommunikasjon med registre og eksterne tjenester vil skje med HTTP. Applikasjonen

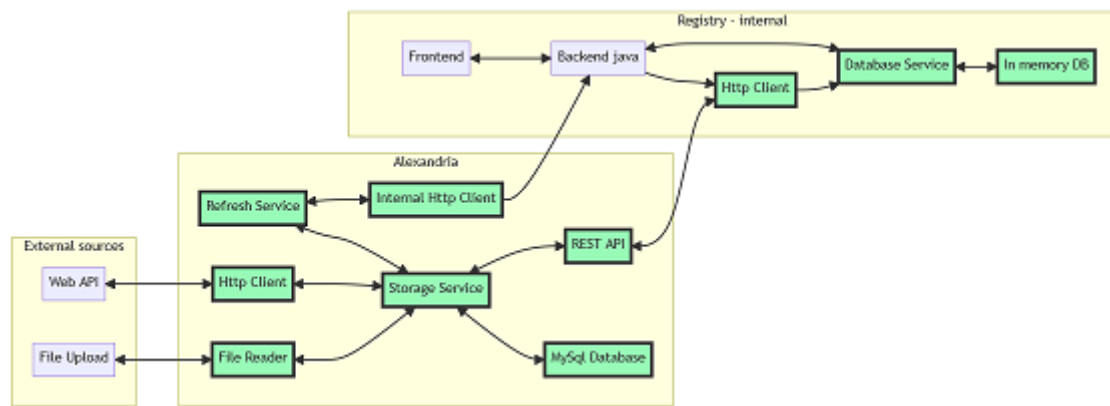


Figure 3: Detaljert dataflyt i applikasjonen. Generert med mermaid.

må da kunne tolke og behandle de ulike HTTP responsene, f.eks loggføring av Bad Request meldinger.

4 Gjennomførbarhetsanalyse

4.1 Tilgjengelige ressurser

4.1.1 Utstyr

Alle gruppe-medlemmer 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, Internett kilder, 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 Risikoanalyse

4.2.1 Bakgrunn for risikoanalyse

I starten av hvert prosjekt er det viktig å vurdere mulig risiko 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 risiko for å ikke fullføre prosjektet [18].

Det ble laget en risiko tabell 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 [18].

Begreper

Uønsket hendelse er en hendelse som kan føre til tap av verdier [19].

Risiko er en kombinasjon av sannsynlighet og konsekvens for en uønsket hendelse [19].

Sannsynlighet viser i hvilken grad er det trolig at en bestemt hendelse vil inntreffe [19].

Konsekvens viser mulig følge av en bestemt hendelse [19].

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

Table 2: Grad av sannsynlighet [18]

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

Table 3: Grad av konsekvens [18]

Beregning av risikonivå

$Risiko(h) = Sannsynlighet(h) * Konsekvens(h)$,
hvor h - hendelse [18]

4.2.2 Risikoanalyse

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	Oppstart møte. Velge kommunikasjons verktøy. Etablere arbeidssplan 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 gruppemedlemmer og veiledere.	Ta kontakt med oppdragsgiver og veiledere for å justere arbeidssplan. 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 gruppemedlemmer hjelper ved behov
4	Dobbeltarbeid	Person / Prosjekt	2	2	4	God kommunikasjon mellom gruppemedlemmer. Utarbeide og følge arbeidsplan. Aktivt bruk av Jira Software (Scrum).	Gjøre nødvendige justeringer i arbeidssplanen. Utføre de kritiske aktivitetene. Omprioritering av aktiviteter ved behov.

Table 4: Risikotabell (del 1) som viser mulige uønskede hendelser, grad av sannsynlighet, grad av konsekvens, beregnet risikonivå og mulige tiltak [18].

Nr.	Beskrivelse av hendelsen	Påvirker	Sannsynlighet	Konsekvens	Risikonivå	Forebyggende tiltak	Korrigerende tiltak
5	Dårlig arbeidsmiljø i en gruppe / Konflikter	Person / Prosjekt	1	4	4	Fordeling av roller og ansvarsområder i en gruppe. God kommunikasjon for å forebygge mulige konflikt-situasjoner.	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 o.l.	Produkt / Prosjekt	2	5	10	Dokumentasjon og kildekode skal lagres både lokalt hos alle gruppemedlemmer og på de valgte sky tjenester 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 / Prosjekt	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.

Table 5: Risiko tabell (del 2) som viser mulige uønskede hendelser, grad av sannsynlighet, grad av konsekvens, beregnet risikonivå og mulige tiltak [18].

4.2.3 Risikomatrise

Risiko kan fremstilles ved hjelp av en risiko matrise, der sannsynlighet er plassert langs x – aksen og konsekvens langs y – aksen. Risiko matrise er delt inn i grønt, gul og rødt farge som representerer henholdsvis lavt, middels og høyt risikonivå for en bestemt hendelse i risiko tabellen. Tabellen risiko akseptkriterier viser behov for igangsettelse av risikoreduserende tiltak i forhold til risikonivå.

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

Figure 4: Risikomatrise

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

Table 6: Risikoakseptkriterier

4.3 Drøfting og vurdering av ulike løsningsmetoder og analyse av forslag

4.3.1 Programmering verktøy / Utviklingsverktøy

Det finnes mange forskjellige programmering verktøy. 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. Kotlin er utviklet av JetBrains som mål å være et bedre alternativ til Java. Kotlin tilbyr en moderne syntaks, mer funksjonell paradigme i form av f.eks. funksjoner som verdier og funksjoner av høyere orden. Kotlin har null verdier som en del av type systemet, en verdi kan aldri være eksplisitt null uten håndtering. Kotlin byr på mye mer enn dette også, i sum gjør det Kotlin til et godt valg over Java [20].

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 mikro rammeverk som egner seg godt til utvikling av applikasjoner i Kotlin. Både Kotlin, Ktor og IntelliJ IDEA er produkter laget av JetBrains. Dette gir et utmerket verktøy støtte for prosjekter. Det kreves ikke så mye for å lage Ktor applikasjon, legge til ønsket funksjonalitet, endre konfigurasjon osv [21].

Ktor er lett og fleksibelt mikro rammeverk. Ved behov kan applikasjonens funksjonalitet enkelt utvides ved hjelp av plugins [22].

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 en god støtte for å utvikle Kotlin prosjekter i en annen IDE, finnes det ikke et bedre alternativ til IntelliJ IDEA [23].

Dev/deployment env

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. Docker image inneholder alle nødvendige biblioteker, konfigurasjons filer og avhengigheter som kreves for å kunne kjøre applikasjonen [24].

Docker Compose er et verktøy som brukes til å konfigurere og starte flere containere samtidig [25].

Det er flere fordeler ved bruk av Docker i programvareutvikling. Docker kan installeres på alle moderne operative systemer, har kort oppstart tid, lav innvirkning på OS og bruker ikke så mye plass på harddisken [26].

Docker containere er portable og derfor kan bli enkelt flyttet og distribuert mellom ulike maskiner. Dette gir et stabilt kjøre miljø for applikasjonen [27].

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 [26].

I motsetning til Docker containere, innkapsler virtuelle maskiner en hel maskin, som gjør deling, ombygning og distribusjon av kode utfordrende. Men virtuelle maskiner har bedre sikkerhet enn Docker containere [27].

I enkelte prosjekter kan det være lurt å kombinere bruk av virtuelle maskiner og Docker containere [27].

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 Prosjektstyring verktøy

Microsoft Teams

Alle gruppemedlemmer er godt kjent med Microsoft Teams. Derfor ble den valgt som kommunikasjons verktøy 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 [28].

Jira Software vs Trello

Både Jira og Trello er populære agile prosjektstyring verktøy. 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 Git repositorier [29].

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, bidrar til bedre kommunikasjon mellom gruppemedlemmer og fordeling av arbeidsoppgaver [29].

Trello er et enkelt verktøy som er basert på Kanban board [30]. Den kan bli brukt for å visualisere ulike arbeidsflyt, men er ikke nok til å støtte store kompliserte prosjekter av type programvareutvikling [31].

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 versjonskontroll system 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 programvare utviklere med åpen kildekode. GitHub egner seg godt for store prosjekter, og har mulighet til å lage både offentlige og private repositorier [32].

GitHub støtter prosjekter som bruker Git versjonskontroll system, 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 [32].

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 [32].

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. Scrum Team 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 [33].

Kanban

Kanban er en annen utviklingsmetode. I motsetning til Scrum, egner den seg godt for mindre prosjekter. Kanban har ingen rollefordeling. Den er ikke inndelt i sprinter, og derfor mer fleksibelt enn Scrum. Kanban brukes til å visualisere kontinuerlig arbeidsflyt [33].

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 produktutvikling [34].

Prosjektgruppe bruker Kanban board for visualisering av arbeidsflyt og setter en grense for antall elementer som kan være i «in progress» tilstand. Dette hjelper med å finne «svakheter» i gruppens arbeid, øker effektiviteten og reduserer sjansene for å ikke fullføre planlagt arbeid [35].

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 formatering 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 dokument deling, 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 kommandoene, 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 [36].

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 [37].

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.md fil som kan bli aktuelt i vårt prosjekt. Det finnes også Mermaid plugins for mange andre tjenester, for eksempel Visual Studio Code og IntelliJ IDEA [38].

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 [39].

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 [29].

Microsoft Excel

Microsoft Excel skal brukes til å lage og føre projektdagbok som gir en oversikt over brukt tid for hele prosjektet og forskjellige del aktiviteter samt viser detaljert beskrivelse av utført arbeid. Excel kan også benyttes til grafisk representering av prosjektarbeid, for eksempel gantt-diagram, sektor diagram osv.

4.4 Valg av løsning og utviklingsmetodikk

Programmering

- Programmeringsspråk: Kotlin
- IDE: IntelliJ IDEA
- Rammeverk: Ktor med Arrow
- Docker containers / Docker Compose
- Relasjondatabase: MySQL

I seksjon 4.3 er verktøyene og alternative verktøy beskrevet og satt opp mot hverandre. Den kvalitative årsaken til valgt teknologi er at det er den teknologien team for kvalitetsregister bruker til tjeneste per dags dato, og er dermed teknologi teamet kan overta, vedlikeholde og videreutvikle etter endt oppgave.

Prosjektstyring

- Kommunikasjon verktøy: Microsoft Teams
- Prosjektstyring verktøy: Jira Software
- Versjonskontrollverktøy: Git og Github

Dette er verktøy vi er komfortabel med å bruke til prosjektarbeid, og er derfor et naturlig valg til dette prosjektet. Kristian jobber med Jira i sin jobb og har erfaring med det. Teams er standard løsning for kommunikasjon levert av UiT for studenter. Git er en akseptert bransjestandard for versjonskontroll av kode.

Dokumentasjon

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

Latex gir stor kontroll for skriving av dokumenter, samtidig som den støtter bibtex filer som gjør at det er meget lett og holde styr på referanser, og eventuelt endre referanse stil ved ønske/behov. Det gjør det også lett å overføre referanser til eventuelt andre dokumenter. Mermaidjs tilbyr en enkel deklarativ måte å definere diagrammer på som støtter versjonskontroll via Git.

Prosjektutvikling

- Utviklingsmetode: Scrumban

5 Kostnadsoverslag

Det er ingen kostnader tilknyttet dette prosjektet per nå. Studentene bruker egen for utvikling av applikasjonen.

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 [40].

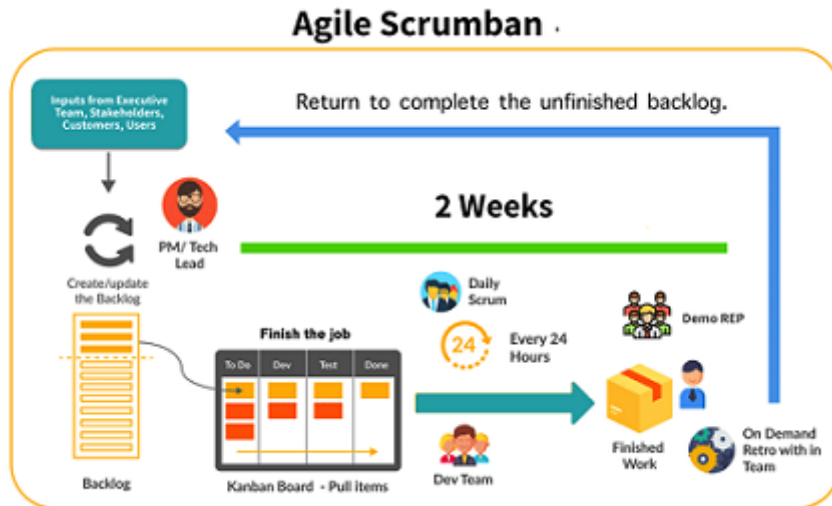


Figure 5: Scrumban prosessen [40]

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 [41].

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 [41].

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 [41]. Scrumban Board er presentert i figur 6.

Kanban board brukes til visualisering av kontinuerlig arbeidsflyt. Elementene fra Backlogen plasseres og tas av gruppemedlemmer etter pull prinsippet. Etter at man er ferdig med sin oppgave, tar en ny fra tavlen [41].

Prosjektgruppen skal møtes hver dag for korte stand-up 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 [34].

På slutten av iterasjonen, samles prosjektgruppe for å diskutere gjennomført arbeid, implementert funksjonalitet, eventuelle mangler, tilleggskrav fra kunden, utviklingsprosess og nødvendig forbedringstiltak som vil gjøre prosjektgruppe mer produktiv neste iterasjon [42].

Etter det starter prosjektgruppe planlegging av en ny iterasjon, justering og prioritering av elementer i Backlogen [42].



Figure 6: Scrumban Board. WIP begrensning i "in progress" kolonne. [34]

6.2 Jira Software

Jira Software er et prosjektstyring verktøy som skal brukes til å støtte Scrumban prosessen. Prosjektet lages ved hjelp av Kanban Project Template. Ved å velge Team-Managed Template, kan man gjøre styring av prosjektet mer fleksibelt. Det finnes også mulighet til å koble GitHub repository til prosjektet [29].

Kanban board bukes 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 [29].

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 [29].

Prosjektgruppe styrer sitt arbeid ved å ta og flytte arbeidsoppgaver på forskjellige steder av Kanban board. Oppgavene skal utføres etter høyest prioritet. Kanban board er presentert i figur 7.

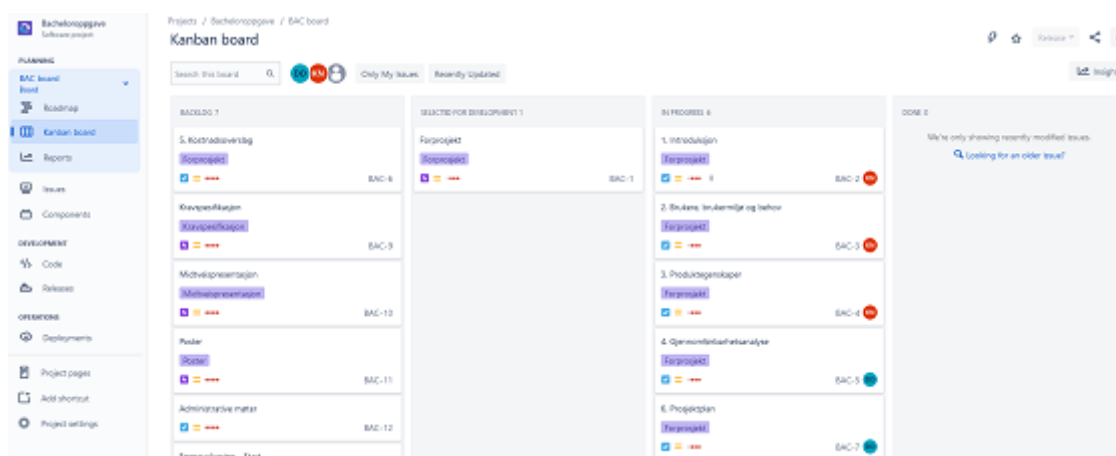


Figure 7: Kanban Board. Generert med Jira Software.

Kanban roadmap som er presentert i figur 8 gir en oversikt over alle Epics og viser hvor lang tid en bestemt Epic tar i løpet av utviklingsprosessen [29].

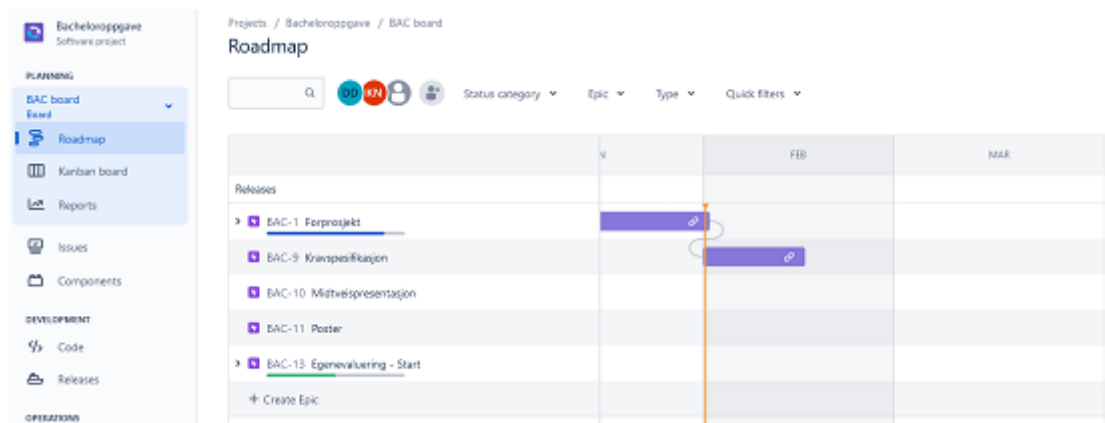


Figure 8: Kanban Roadmap. Generert med Jira Software.

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 [29].

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 prosjektgruppen. 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 gruppemedlemmer. Begge to skal aktivt delta i planlegging av prosjektarbeid, utviklingen av selve systemet og ferdigstillelse 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 dokument ansvarlig, 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 møtetreferater.

Alle gruppemedlemmer 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 ferdigstillelse av nødvendig dokumentasjon, rapporter og diagrammer.

Alle gruppemedlemmer 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

Kristian Nedrevold, Daria Danilina

Samarbeidsavtalen er signert 31. januar 2023.

8 Referanser

- [1] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Norsk gynekologisk endoskopiregister [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/gynekologi/norsk-gynekologisk-endoskopiregister/>.
- [2] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Nasjonalt register for ablasjonsbehandling og elektrofysiologi i norge (ablanor) [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/hjerte-og-karsykdommer/nasjonalt-register-ablasjonsbehandling-og-elektrofysiologi-i-norge>.
- [3] Nordlandssykehuset HF. Norsk kvalitetsregister for behandling av spiseforstyrrelser - norspis [internett]. 2023 [cited2023Jan23]. <https://nordlandssykehuset.no/forskning-og-innovasjon/kvalitetsregistre/norsk-kvalitetsregister-for-behandling-av-spiseforstyrrelser-norspis>.
- [4] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Norsk register for arvelige og medfødte nevromuskulære sykdommer [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/nervesystemet/norsk-register-arvelige-og-medfodte-nevromuskulaere-sykdommer>.
- [5] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Norsk register for analinkontinens [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/mage-og-tarm/norsk-register-analinkontinens>.
- [6] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Norsk register for gastrokirurgi (norgast) [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/mage-og-tarm/norsk-register-gastrokirurgi-norgast>.
- [7] Universitetssykehuset Nord-Norge HF. Nordnorsk kvalitetsregister for endokarditt [internett]. 2023 [cited2023Jan23]. Available from: <https://unn.no/fag-og-forskning/medisinske-kvalitetsregistre/nordnorsk-kvalitetsregister-for-endokarditt>.
- [8] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Nasjonalt register for langtids mekanisk ventilasjon [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/luftveier/nasjonalt-register-langtids-mekanisk-ventilasjon>.
- [9] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Norsk kvalitetsregister for fedmekirurgi (soreg-norge) [internett]. 2023 [cited2023Jan23]. Available from: <https://helsebergen.no/avdelinger/laboratorieklinikken/medisinsk-biokjemi-og-farmakologi/norsk-kvalitetsregister-for-fedmekirurgi-soreg-norge>.
- [10] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Nasjonalt kvalitetsregister for ryggkirurgi [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/register/muskel-og-skjelett/nasjonalt-kvalitetsregister-ryggkirurgi>.
- [11] Nasjonalt servicemiljø for medisinske kvalitetsregistre. Kvalitetsforbedring [internett]. 2023 [cited2023Jan23]. Available from: <https://www.kvalitetsregistre.no/kvalitetsforbedring/>.

- [12] Tomas Jernberg, Mona F Attebring, Kristina Hambræus, Torbjorn Ivert, Stefan James, Anders Jeppsson, Bo Lagerqvist, Bertil Lindahl, Ulf Stenestrand, and Lars Wallentin. The swedish web-system for enhancement and development of evidence-based care in heart disease evaluated according to recommended therapies (swedeheart). *Heart*, 96(20):1617–1621, 2010.
- [13] Norsk helsenett. Medisinske kvalitetsregistre [internett]. 2023 [cited2023Jan23]. Available from: <https://www.nhn.no/tjenester/medisinske-kvalitetsregistre/>.
- [14] Geir Sverre Braut (Høgskulen på Vestlandet) Ulrik Malt (Universitetet i Oslo). Store medisinske leksikon - icd-10 [internett]. 2023 [cited2023Jan23]. Available from: <https://sml.snl.no/ICD-10>.
- [15] Direktoratet for e helse. Ncmp: Kodeverk for medisinske prosedyrer. 2023 [cited2023Jan23]. Available from: <https://www.ehelse.no/standardisering/standarder/ncmp-kodeverk-for-medisinske-prosedyrer>.
- [16] Direktoratet for e helse. Ncsp: Kodeverk for kirurgiske prosedyrer. 2023 [cited2023Jan23]. Available from: <https://www.ehelse.no/standardisering/standarder/ncsp-kodeverk-for-kirurgiske-prosedyrer>.
- [17] JetBrains. Coroutines. [internett]. 2023 [cited2023Jan23]. Available from: <https://kotlinlang.org/docs/coroutines-overview.html>.
- [18] Tanita Fossli Brustad. Risikoanalyse. 2022.
- [19] Ann Karin Midtgaard. Ns 5814:2021. krav til risikovurderinger. [internett]. 2021.
- [20] Rute Figueiredo Marianna Berga. Kotlin vs java: the 12 differences you should know. [internett]. 2021 [cited2023Jan23]. Available from: <https://www.imaginarycloud.com/blog/kotlin-vs-java/>.
- [21] JetBrains. IntelliJ idea ultimate support [internett]. 2023 [cited2023Jan23]. Available from: <https://ktor.io/idea/>.
- [22] JetBrains. Simple and fun. [internett]. 2023 [cited2023Jan23]. Available from: <https://ktor.io/>.
- [23] JetBrains. What is intellij idea? [internett]. 2023 [cited2023Jan23]. Available from: <https://www.jetbrains.com/idea/features/>.
- [24] Simran Arora. Docker vs. virtual machines: Differences you should know [internett]. 2022 [cited2023Jan23]. Available from: <https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know/>.
- [25] kode24. Hva er greia med docker? [internett]. 2021 [cited2023Jan23]. Available from: <https://www.kode24.no/artikkel/hva-er-greia-med-docker/74020891>.
- [26] Docker. Use containers to build, share and run your applications [internett]. 2023 [cited2023Jan23]. Available from: <https://www.docker.com/resources/what-container/>.
- [27] Jacob Schmitt. Containers vs virtual machines: what is the difference? [internett]. 2022 [cited2023Jan23]. Available from: <https://circleci.com/blog/containers-vs-virtual-machines/>.
- [28] Microsoft. Microsoft teams [internett]. 2023 [cited2023Jan23]. Available from: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software/>.

- [29] Atlassian. Jira software [internet]. 2023 [cited2023Jan23]. Available from: <https://www.atlassian.com/software/jira>.
- [30] Atlassian. Trello. Trello brings all your tasks, teammates, and tools together [internet]. 2023 [cited2023Jan23]. Available from: <https://trello.com/home>.
- [31] Dale Jakes. Trello vs. jira: Top agile project management tool. [internet]. 2022 [cited2023Jan23]. Available from: <https://technologyadvice.com/blog/information-technology/trello-vs-jira-choosing-an-agile-project-management-tool/>.
- [32] manmeetjuneja5. Difference between bitbucket and github [internet]. 2022 [cited2023Jan23]. Available from: <https://www.geeksforgeeks.org/difference-between-bitbucket-and-github/>.
- [33] Max Rehkopf. Kanban vs. scrum: which agile are you? [internet]. 2023 [cited2023Jan23]. Available from: <https://www.atlassian.com/agile/kanban/kanban-vs-scrum/>.
- [34] Monica Dhiman. What is the scrumban methodology and how to implement it? [internet]. 2022 [cited2023Jan23]. Available from: <https://blog.logrocket.com/product-management/what-is-scrumban-methodology-how-to-implement/core-principles-of-scrumban/>.
- [35] Dan Radigan. What are wip limits? [internet]. 2023 [cited2023Jan23]. Available from: <https://www.atlassian.com/agile/kanban/wip-limits/>.
- [36] GitHub Docs. Quickstart for writing on github [internet]. 2023 [cited2023Jan23]. Available from: <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/quickstart-for-writing-on-github/>.
- [37] EndNote. Accelerate your research [internet]. 2023 [cited2023Jan23]. Available from: <https://endnote.com/>.
- [38] Mermaid. Mermaid diagramming and charting tool [internet]. 2023 [cited2023Jan23]. Available from: <https://mermaid.js.org/>.
- [39] VisualParadigm. Infographic maker [internet]. 2023 [cited2023Jan23]. Available from: <https://online.visual-paradigm.com/>.
- [40] Anuj Seth. Scrumban: An overview [internet]. 2022 [cited2023Jan23]. Available from: <https://pmtips.xyz/2022/08/11/scrumban-an-overview/>.
- [41] Teamhood. Kanban vs scrum vs scrumban - how do they compare? [internet]. 2022 [cited2023Jan23]. Available from: <https://teamhood.com/kanban/scrum-vs-kanban-vs-scrumban-how-do-they-compare/>.
- [42] Corey Ladas. Scrumban [internet]. 2022 [cited2023Jan23]. Available from: <https://www.agilealliance.org/scrumban/>.

9 Liste over figurer

1	Sequence diagram. Generert med mermaid.	3
2	Overordnet dataflyt i løsningen. Generert med mermaid.	4
3	Detaljert dataflyt i applikasjonen. Generert med mermaid.	9
4	Risikomatrise	14
5	Scrumban prosessen [40]	22
6	Scrumban Board. WIP begrensning i "in progress" kolonne. [34]	23
7	Kanban Board. Generert med Jira Software.	23
8	Kanban Roadmap. Generert med Jira Software.	24

10 Liste over tabeller

1	Sammendrag av brukerens behov	7
2	Grad av sannsynlighet [18]	11
3	Grad av konsekvens [18]	11
4	Risikotabell (del 1) som viser mulige uønskede hendelser, grad av sannsynlighet, grad av konsekvens, beregnet risikonivå og mulige tiltak [18].	12
5	Risiko tabell (del 2) som viser mulige uønskede hendelser, grad av sannsynlighet, grad av konsekvens, beregnet risikonivå og mulige tiltak [18].	13
6	Risikoakseptkriterier	14

11 Stikkordsliste

API	Application Programming Interface
HN	Helse Norge
IKT	Informasjons- og kommunikasjonsteknologi
ICD10	International Classification of Diseases and Related Health Problems 10th Revision
NCMP	Kodeverk for medisinske prosedyrer
NCSP	Kodeverk for kirurgiske prosedyrer
NHN	Norsk Helse Nett
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
JVM	Java Virtual Machine

12 Vedlegg

Vedlegg 1: Samarbeidsavtale