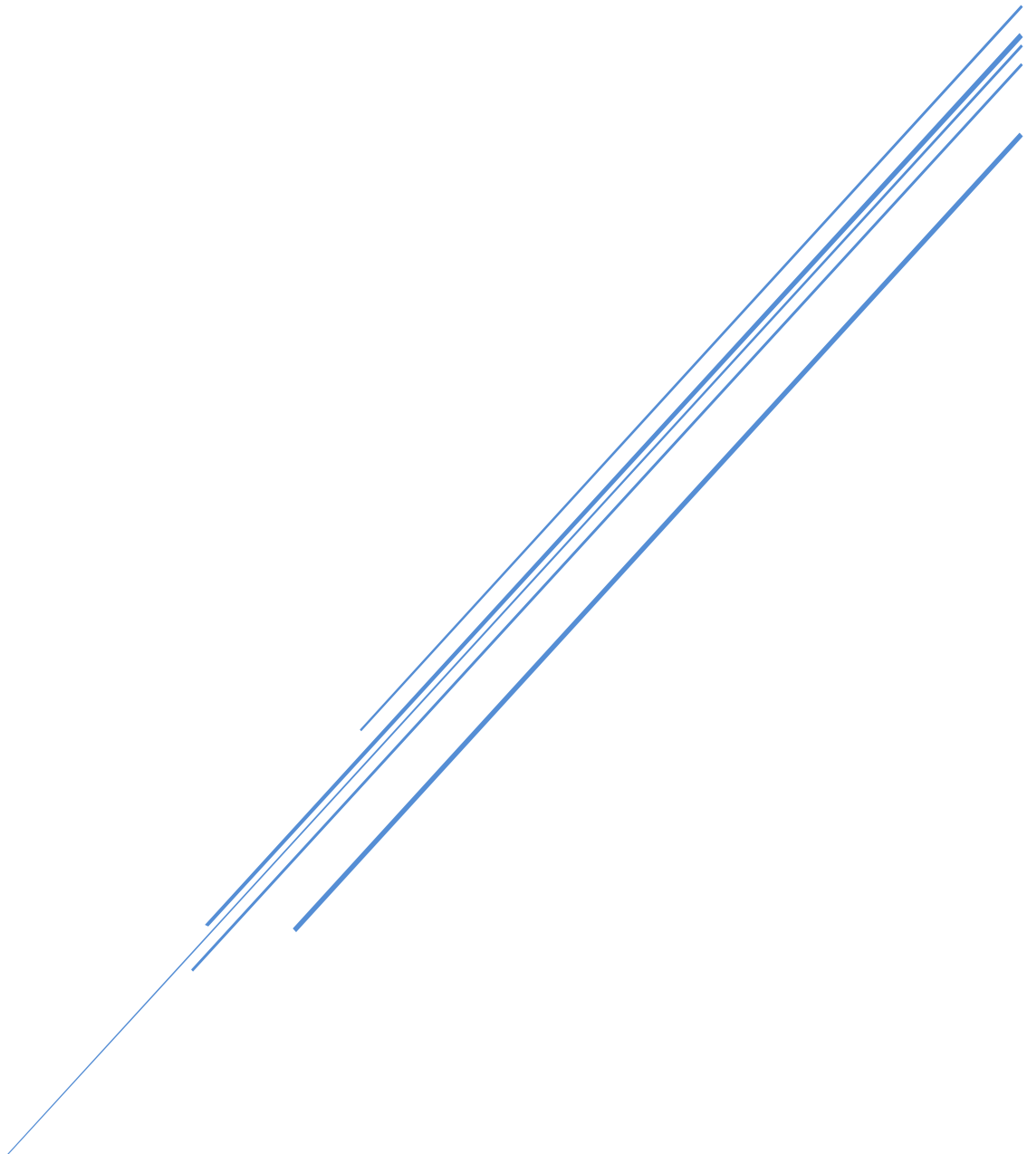


FORPROSJEKTRAPPORT

Gruppe 38 – Goodtech Transaction Manager



Innholdsfortegnelse

GoodTech	2
Gruppen vår	2
Presentasjon	3
Dagens situasjon	4
Mål og rammebetingelser	4
Løsninger / alternativer	5
Alternativ løsning	5
Teknisk	5
Arbeidsmetoder	6
Arkitektur	6
Risikoanalyse	7
Fremdriftsplan	9
Teknologier	10

GoodTech

Goodtech er et teknologikonsern med kjernevirksomhet innen automasjon, kraft og industriell prosjektering. Vår visjon er å være førstevalget for industriell effektivitet. GoodTech leverer prosjekter, service og produkter til kunder innen prosess- og produksjonsindustrien, havbruk, olje & gass, kraft og vannrensing. Goodtech er notert på Oslo Børs, omsetter for ca. NOK 600 millioner og har ca. 300 medarbeidere i Norge, Sverige og på Åland.

Goodtech har lokaler i Arvika, Bergen, Fredrikstad, Førde, Karlstad, Mariehamn, Moss, Os, Oslo, Porsgrunn og Trondheim hvor hovedkontoret er i Oslo ved Helsfyr.

Gruppen vår

Midlertidig prosjektside

<https://bachelor.sagen.app>

Alexander Meisdalen Sagen

E-post: s326194@oslomet.no

Tlf: +47 480 73 448

Alexander er en IT-student på OsloMet som har vært interessert i kode og teknologi fra en alder av 10 år. Gjennom mange år med sterk interesse har Alexander vært innom mange teknologistacker. Alexander jobber i dag som utvikler hos OsloMet og Sense Data AS.

Marius Rostad

E-post: s326185@oslomet.no

Tlf: +47 968 50 702

Marius er en IT-student på OsloMet med en stor entusiasme for teknologi, kode, og sikkerhet. Marius har hittil konsultert Mesterbex AS, Fossheim bygg AS med sine IT-kompetanse, og er nå utvikler for Sense Data AS.

Gaute Kvalheim

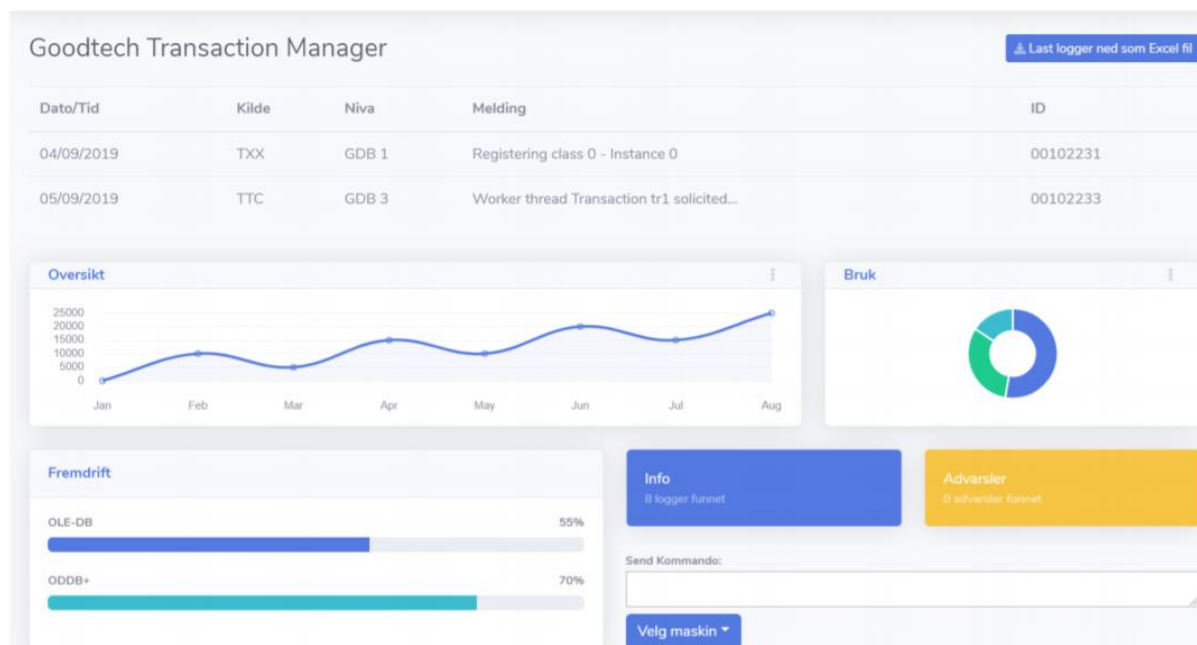
E-post: s326197@oslomet.no

Tlf: +47 913 61 611

Gaute studerer IT på OsloMet med fokus på programmering i Java, Spring og Android-utvikling. På siden av studiene jobber han som utvikler for Sense Data AS hvor han har bidratt i flere prosjekter, inkludert utvikling av nettside og app for firmaet.

Presentasjon

Bachelor oppgaven vår er å lage en egenutviklet løsning, som et potensielt erstatning/alternativ til deres teknologi stack. Goodtech bruker nå et program som heter Factorytalk som sin Transaction Manager, men Factorytalk har problemer. Programmet blir ikke lenger videreutviklet av eierne. Problemer Goodtech har i dag er: Factorytalk er veldig vanskelig og tidkrevende å sette opp for deres kunder. Det skjer ofte med feil på grunn av kompleksitet i programmet. Vår applikasjon skal lages som en potensiell erstatning til Transaction manageren de allerede har.



Kontaktperson fra Goodtech ASA

Kjartan Holmedal

E-post: kjartan.holmedal@goodtech.no

Tlf: +47 918 81 681

Intern veileder

Qian Meng

E-post: qianmeng@oslomet.no

Tlf: +47 672 38 009

Dagens situasjon

Goodtech er et teknologikonsern med kjernevirksomhet innen automasjon, kraft og industriell prosjektering. De leverer prosjekter, service og produkter til kunder innen prosess- og produksjonsindustrien, havbruk, olje & gass, kraft og vannrensing.

Hittil har Goodtech brukt et program som heter Factorytalk som sin Transaction Manager. Dessverre er programmet ikke modulert nok til deres forventninger, og derfor ønsker de å lage en egenutviklet applikasjon for å erstatte den. FactoryTalk er veldig tung og ressurskrevende å sette opp, og er vanskelig å utvide eller tilpasse til deres smak. Programvaren er også dyr og får ikke lenger oppdateringer.

GoodTech ønsker derfor å utvikle sin egen transaction manager som er lightweight, enkel å sette opp og tilpasset sine egne behov. Det skal være enkelt å legge til flere data koblinger til systemet og sette opp nye triggere.

GoodTech har tidligere god erfaring med å ta inn studenter fra OsloMet og har dette semesteret tatt inn 3 grupper. For oss studenter er dette en fantastisk måte å bli mer kjent med arbeidslivet, og man får brukt teknikker vi har lært i løpet av studiet.

På side 6 kan man se arkitekturen vi skal jobbe med. Det er en OPC-UA server som brukes for å kommunisere med PLSer nede på fabrikkgulvet. Disse PLSene sender og spør om informasjon hele tiden, og jobben til Transaksjon Manageren er å respondere til konfigurerte forespørsler som kommer over OPC-UA fra PLSene.

Mål og rammebetingelser

Målet med dette prosjektet er å lage en Transaction Manager i Spring Boot som kan kommunisere med PLSer over OPC-UA protokollen.

Applikasjonen skal kunne:

- Sette opp/velge Control Connector. Dette er hvor vi får trigger signalet fra. Eksempel: OPC-UA, MQTT, andre protokoller osv. Vi skal bare fokusere på OPC-UA, men vi skal legge opp til at man skal kunne endre/legge til nye Control Connectors.
- Sette opp/velge Definere Enterprise Connector. Dette er hvor dataen skal sendes til, og der vi får resultat fra. Eks: SQL server med Procedures, Kafka kø, etc.
- Legge til/oppdatere Data points/data tag. Dette er konfigurasjonen på hvordan data skal lagres og håndteres. Eks: Navnet på verdien som lagres, og hvilken datatype det er.
- Legge til/oppdatere transaksjonsinformasjon. Hvordan man skal håndtere transaksjonen. Den har også informasjon om hvilken SQL Procedure som skal kjøres, eller hvilken Kafka Channel som skal brukes.
- Konfigurere trigger signal. Hvordan skal triggeren starte. Eks: Er det en med et jevnlig intervall, eller kommer den fra en annen kilde.

Løsninger / alternativer

Den foreslått løsningen er å lage et Transaction Manager i Spring Boot som er en OPC-UA Client som lytter til Tags. Henter og svarer med informasjon via OPC-UA. Konfigurasjonen vil da være: Hvilke tags lytter vi på, hvordan håndterer vi dataene. Man lagrer konfigurasjonen i databasen til applikasjonen vår istedenfor å bruke Tag Databasen Goodtech allerede har. Applikasjonen skal være modulær i form av konfigurasjon av triggere, og hvor vi sender/lagrer informasjon (eventuelt får svar fra å).

Alternativ løsning

Tar utgangspunktet i alternativet over, men vi velger å modularisere Control Connectors og vi velger å bruke Tag Databasen til Goodtech. Dette alternativet hadde vært den mest ideelle for Goodtech, men med dette alternativet tilfellet trækker vi over grensen det vi vil kunne oppnå i løpet av de arbeidstiden for prosjektet. Kravspesifikasjonen blir i det tilfellet så stor at vi risikerer å ikke bli ferdig med prosjektet.

Teknisk

Vi har blitt enige med GoodTech å jobbe 3 dager i uken; jobb hver tirsdag, onsdag og fredag. Da tilbyr GoodTech lokaler (pult og møterom) samt er så snille å gi oss lunsj de dagene vi jobber. Lokalene er i 4. etasje på Innspurten 15 på Helsefyr. Vi har fått hvert vårt tilgangskort som gir tilgang til bygget hver dag. Kortet krever kode før 08:00 og etter 17:00 som vi har fått tildelt.

For å finne feil i koden, logikken eller strukturen har vi blitt enige om at vi skal sende all ny kode til "code review" hos Goodtech. Vi har fått oppnevnt at Roger skal få hovedansvaret for å gå gjennom all ny kode vi skriver og gi oss feedback til eventuelle feil eller mulige feilkilder.

Utover ren kode har vi fått oppnevnt Kjartan som ansvarlig for funksjonalitet og overordnet arkitektur. Det er han som kjenner de omkringliggende systemene best og vet hvordan de forskjellige kommunikasjonsprotokollene virker. Det er også Kjartan som har mest erfaring med de nåværende systemene og hvilke krav som stilles til vårt produkt.

Vi har også avtalt med GoodTech å ha et statusmøte hver tirsdag klokken 14-15. Her vil vi fortelle om hva vi har fått gjort den siste uken og får mulighet til å stille dem spørsmål vi har fått den siste uken. Dette er også en god mulighet til å diskutere planene for den neste ukens arbeid og velge hvilke prosjekter fra backlog vi skal prioritere og hvordan disse skal implementeres.

Som en start har vi blitt enige med GoodTech om å opprette vår egen MsSQL database med egne testprosedyrer selv. Alexander har tilgang på en Linux server så han satte opp en rask MsSQL container på den med adressen `sagen.app:1433`. Denne har alle på gruppen fått fri tilgang til og vi bruker *SQL Server Management Studio* for å administrere denne. Videre har GoodTech satt opp en OPC-UA Ignition server som vi kan koble oss til. På denne har de lagt til simulasjoner som blant annet returnerer en sinusbølge. Dette gir pålitelig varierende data som gjør det enkelt å oppdage feil i triggere og utveksling av data.

Arbeidsmetoder

Vi kommer til å følge smidige prinsipper under arbeidet med oppgaven. Dette innebærer å holde daglige standups for å oppdatere hverandre om hva vi har gjort og hva som skal gjøres, og å fordele oppgaver for dagen. I tillegg ønsker vi å bruke Trello for å definere og fordele oppgavene på en enkel måte, og for å holde oversikt over hva som er gjort og hva som må gjøres. På denne måten vil vi minimere konflikter i koden ved at to personer ikke jobber med samme oppgave samtidig uten å være klar over det, og at alle er oppdatert på hva som skjer/har skjedd.

Daglige standups vil skje på de dagene vi jobber på kontorene til Goodtech. Det er nå satt opp at vi skal jobbe på kontorene på tirsdager, onsdager og fredager. Vi kommer til å bytte på rollen som Scrum Master i løpet av prosjektet. Slik får vi alle sammen erfaring med rollen, i tillegg til at vi alltid har en person som kan styre standups for å sørge for at det er effektivt. Hver dag vil vi føre en prosjektdagbok for å ha oversikt over hva som er gjort hver dag, hvilke problemer som har oppstått og hvordan vi eventuelt løste dem. Hver torsdag kl. 09:00 skal vi ha møte med veileder og andre grupper på OsloMet. På disse møtene vil vi gi en oppsummering av ukens arbeid, basert på prosjektdagboken.

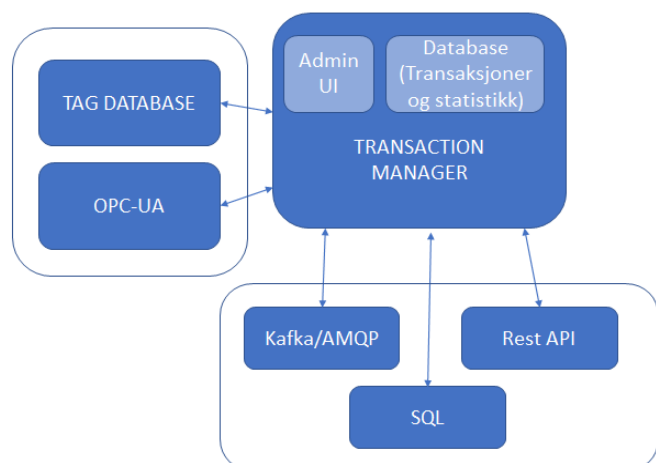
Arkitektur

Goodtech utvikler og setter opp programvare og systemer for produksjonslinjer. Eksempler på produksjonslinjer er fiskefor fabrikker, jern og titandioksid slaggb fabrikker, og aluminiumsverk.

På fabrikkgulvet blir alle roboter, maskiner og sensorer styrt av forskjellige PLSer (Programmerbar Logisk Styring). Disse PLSene er av forskjellige merker og har forskjellige protokoller for å kommunisere. Derfor har man opprettet en kommunikasjonsstandard som tillater PLSene å snakke med hverandre. Denne standarden kalles OPC-UA (Open Platform Communications - Unified Architecture). Denne OPC-UA standarden baserer seg som regel på å ha en UA server for hele produksjonslinjen.

Goodtech bruker oppå dette en Tag Database som fungerer som et abstraksjonslag for å snakke med forskjellige konnektorer nedover (OPC-UA, SQL)

For å koble produksjonslinjen til eksterne systemer som økonomi, administrasjon og overvåking trenger goodtech et system som henter ut verdier fra PLSene og distribuerer disse til høyere systemer.



Risikoanalyse

Hva kan gå galt?	Sannsynlighet	Konsekvens	Forebygging	Tiltak om det skjer
Kort fravær	Høy	Lav Personen som er fraværende kan kanskje ikke møte opp på kontoret til avtalte tider og kan kanskje ikke fullføre sine oppgaver innen tidspunktet som er satt.	Ved at alle har god forståelse av oppgaven kan vi sørge for at andre kan ta over arbeidsoppgavene til personen som er syk dersom det trengs. Man kan også unngå større konsekvenser ved å beregne god tid på oppgaver.	Andre personer på gruppen kan ta over viktige oppgaver. I noen tilfeller vil det kanskje være mulig for personen som er syk å jobbe hjemmefra.
Tidsberegning	Middels	Middels Ved å feilberegne tiden det vil ta å fullføre oppgaven risikerer vi å ende opp med en dårlig eller ikke fullført løsning.	Vi må være nøye på å følge smidige prinsipper og begrense prosjektet tidlig om nødvendig.	Vi bør se hvor mye vi kan øke arbeidsinnsatsen. Dersom dette ikke er nok må vi snakke med oppdragsgiver om å begrense prosjektet på en måte som går minst mulig ut over funksjonaliteten til sluttproduktet.
Endring av kravspesifikasjon	Lav	Middels Dersom oppdragsgiver etter hvert ønsker å endre på kravene til oppgaven kan det skape problemer da vi har brukt mye tid og må muligens gjøre ting på nytt.	Sørge for å følge prinsippet om loose coupling og high cohesion for å enklere kunne endre en del av systemet uten å måtte endre hele prosjektet. Ved å følge smidige prinsipper vil vi være bedre forberedt på eventuelle endringer i kravspesifikasjonen.	Gå gjennom hvilke deler av prosjektet dette påvirker og planlegge enklest mulig måte å tilpasse disse endringene.

Feiltolkning av oppgaven	Lav	Høy Dersom vi feiltolker oppgaven kan det ende med at oppdragsgiver ikke er fornøyd eller at man etter hvert finner ut at vi ikke kan løse oppgaven slik vi trodde.	Kontinuerlig holde oppdragsgiveren vår oppdatert med hvordan det går med applikasjonen, og stille spørsmål til oppgaven å fremtidig funksjonalitet.	Vi må være mer kritisk til det oppdragsgiver prøver å formidler om oppgaven. Vi må også be om konkrete eksempler/mer informasjon om oppgaven.
Dårlig kommunikasjon	Middels	Middels Dårlig kommunikasjon innad i gruppen kan føre til tapt tid, unødvendig arbeid og feil i løsningen. Dårlig kommunikasjon mellom gruppen og oppdragsgiver kan føre til feiltolkning av oppgaven eller mangel på informasjon som igjen fører til tap av tid og en dårligere løsning enn ønsket.	Ha møte ukentlig med oppdragsgiver for å diskutere hva som er gjort og hva som skal gjøres. Da får arbeidsgiver en mulighet til å gi innspill og korrigere feil så tidlig som mulig.	Ta det opp på Daily standup. Det er viktig å sørge for at alle er på samme bølgelengde når vi skal utvikle applikasjonen sammen.
Tap av data	Lav	Høy Tap av data vil føre til mye tap av tid og kompetanse.	Bruke VCS og holde en oppdatert kopi på alles maskin.	Siden vi bruker Subversion har alle en kopi av prosjektet på sin PC.
Lengre fravær	Lav	Høy Om en person må bli borte fra prosjektet over lengre tid vil det være en fare for at vi ikke rekker å fullføre prosjektet.	Ha daglige og ukentlige møter for å vite hva alle andre gjør. Ha tydelig definerte oppgaver og følge SCRUM.	Fordele arbeidsoppgaver på resten av gruppen. Skulle det være nødvendig kan man avtale med oppdragsgiver å begrense prosjektet.

Fremdriftsplan

Fase	Varighet	Beskrivelse
Oppstartsfase	02.12.19 – 09.01.20	I oppstartsfasen har vi møter med Goodtech for å få en oversikt over prosjektet, skrive en projektskisse og komme frem til en avtale studentene, OsloMet og Goodtech er enige i.
Forprosjekt	09.01.20 – 17.01.20	Forprosjektfasen handler om å få en bedre oversikt over hva som må gjøres, utforme en fremdriftsplan og risikoanalyse, og skrive det om til en forprosjektrapport. Forprosjektrapporten skal også inneholde mål og rammebetingelser, beskrivelse av løsningen og alternative løsninger og teknologiene som skal brukes. I denne fasen har vi møter for å definere oppgaven mer spesifikt sammen med Goodtech.
Startfase	18.01.20 – 31.01.20	I startfasen skal vi begynne å sette opp utviklingsmiljøet, få tilgang til repository, OPC-UA test server og bli bedre kjent med teknologiene vi skal bruke. Vi skal bygge en full forståelse for oppgaven og dele opp prosjektet i mindre deler som blir lagt på backlog.
Utviklingsfase	01.02.20 – 30.04.20	I utviklingsfasen vil vi bruke Scrum sprinter for å utvikle programvaren. Vi vil først utvikle en minimal løsning som vil bli videreutviklet til å utfylle kravspesifikasjonen. I løpet av fasen vil vi dokumentere arbeidet som blir gjort og alle problemer og løsninger vi kommer over. I slutten av denne fasen vil vi ha et produkt som størst mulig grad oppfyller kravspesifikasjonen og er klar for grundig integrasjonstesting.
Testfase	01.05.20 – 10.05.20	Her setter vi opp hele systemet i et større testmiljø med alle tjenester og programmer vår system skal kunne samhandle med. Her vil vi i samarbeid med Goodtech sørge for at alle funksjoner virker som de skal.
Sluttfase	11.05.20 – 25.05.20	I sluttfasen fullfører vi all dokumentasjon og rydder opp i koden. Her vil også sluttrapporten fullføres, og vi begynner å forberede presentasjonen.

Teknologier

En del av teknologiene vi har diskutert å bruke er listet under med en liten grov beskrivelse og lenke til mer informasjon.

Spring boot

Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

[Spring Boot](#)

Vaadin

Vaadin is an open-source platform for web application development. The Vaadin Platform includes a set of web components, a Java web framework, and a set of tools and application starters.

[Vaadin - The best Java framework for Progressive Web Apps](#)

OPC-UA

OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation. Distinguishing characteristics are: Focus on communicating with industrial equipment and systems for data collection and control.

[Unified Architecture](#)

Kafka

Apache Kafka is an open-source stream-processing software platform developed by LinkedIn and donated to the Apache Software Foundation, written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

[Apache Kafka](#)

SQL (Microsoft SQL)

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network.

[SQL Server 2019](#)

Rest API

Representational state transfer is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet.

[Representational state transfer](#)

MQTT / AMQP

The Advanced Message Queuing Protocol is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing, reliability and security.

[AMQP: Home](#)

Java 8

Java 8 is a revolutionary release of the world's #1 development platform. It includes a huge upgrade to the Java programming model and a coordinated evolution of the JVM, Java language, and libraries. ... Welcome to the latest iteration of the largest, open, standards-based, community-driven platform.

[What's New in JDK 8](#)

Maven

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

[Maven – Welcome to Apache Maven](#)

Subversion (Apache)

Apache Subversion is a software versioning and revision control system distributed as open source under the Apache License. Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.

[Apache Subversion](#)

Jenkins

Jenkins is a free and open source automation server. Jenkins helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

[Jenkins](#)

JUnit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.

[JUnit 5](#)

IntelliJ IDEA

IntelliJ IDEA is an integrated development environment written in Java for developing computer software. It is developed by JetBrains, and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development.

[IntelliJ IDEA: The Java IDE for Professional Developers](#)

Javadoc

Javadoc is a documentation generator created by Sun Microsystems for the Java language for generating API documentation in HTML format from Java source code. The HTML format is used for adding the convenience of being able to hyperlink related documents together.

[javadoc](#)

SQL Server Management Studio

SQL Server Management Studio is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. It's the successor to the Enterprise Manager in SQL 2000 or before.

[Download SQL Server Management Studio \(SSMS\) - SQL Server Management Studio \(SSMS\)](#)