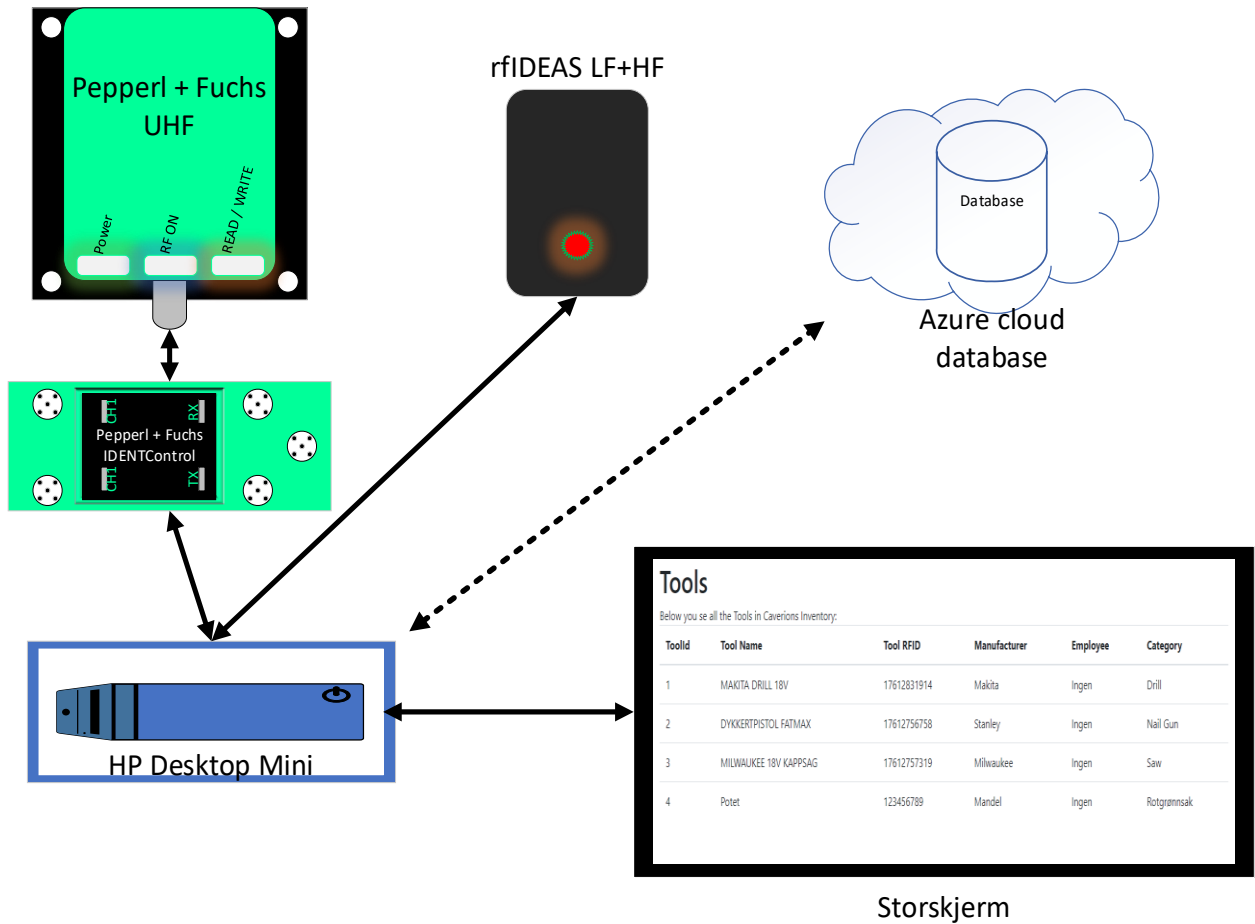


PRH612-1 Bacheloroppgave

# Implementering av automatisk utlånsystem for verktøylager



IA6-6-21

**Emne:** PRH612-1 Bacheloroppgave

**Tittel:** Implementering av automatisk utlånssystem for verktøylager

Denne rapporten utgjør en del av vurderingsgrunnlaget i emnet.

**Prosjektgruppe:** IA6-6-21

**Tilgjengelighet:** Åpne

**Gruppedeltakere:** Nickolas Helgeland  
Jan-Erik Lunde

**Veileder:** Hans-Petter Halvorsen

**Prosjektpartner:** Caverion

**Godkjent for arkivering:** \_\_\_\_\_

**Sammendrag:**

Caverion har en del kostbart verktøy og har tidligere prøvd å implementere flere ulike systemer for å registrere utlån/innleveringen av verktøyene. Den manuelle prosessen i disse systemene har vist seg å være et problem, og en automatisk løsning er derfor ønsket.

Målet med prosjektet var å lage et komplett system som håndterer utlån og innlevering av verktøy, visning av verktøy og ansatte, og funksjoner for å redigere verktøy og ansatte. For å oppnå dette målet ble det utviklet to applikasjoner. En desktop-applikasjon for registrering av verktøy og en web-applikasjon som håndterer visning og redigering av verktøy og ansatte.

For prosjektet ble det valgt å bruke RFID-sensorteknologi fra leverandøren Pepperl + Fuchs. RFID-leseren gjør det mulig å detektere RFID-brikker over lengre avstander, slik at registreringsprosessen blir automatisk.

Valg av metoder og verktøy brukt i dette prosjektet baserer seg på kunnskaper og erfaringer studentene har ervervet seg gjennom studieløpet.

**Course:** PRH612-1 Bacheloroppgave

**Title:** Implementation of a tool inventory management system

This report forms part of the basis for assessing the student's performance on the course.

**Project group:** IA6-6-21

**Availability:** Open

**Group participants:** Nickolas Helgeland  
Jan-Erik Lunde

**Supervisor:** Hans-Petter Halvorsen

**Project partner:** Caverion

**Approved for archiving:** \_\_\_\_\_

**Summary:**

Caverion has several expensive tools and has previously tried to implement several different systems for registering borrowing / returning of the tools. The manual process in these systems has proven to be a problem, and an automatic solution is therefore desired.

The aim of the project was to create a complete system that handles borrowing and returning of tools, display of tools and employees, and functions for editing tools and employees. To achieve this goal, the group developed two applications. A desktop application for registering tools and a web application that handles viewing and editing of tools and employees.

For the project, it was chosen to use RFID sensor technology from the supplier Pepperl + Fuchs. The RFID reader makes it possible to detect RFID tags over longer distances, so that the registration process is automatic.

The choice of methods and tools used in this project is based on knowledge and experience the students have acquired through the course of study.

# Forord

Denne rapporten er en del av bachelorprosjektet, hvor vi dokumenterer arbeidsprosessen for utviklingen av et lagerstyringssystem for Caverion.

Rapporten konkluderer en treårig bachelor fullført av studenter ved Universitet i Sørøst-Norge. Gruppen består av to studenter som går elektroingeniør linjen innenfor Informatikk og Automatisering A-vei ved fakultetet for teknologi, naturvitenskap og maritime fag på USN.

Forsidebildet er laget av studentene og gir en oversikt over systemet.

Link til web-applikasjonen: <https://caverionims.azurewebsites.net>

I dette prosjektet har følgende dataverktøy blitt brukt:

- MS Excel 2019
- MS Power Point 2019
- MS Project 2019
- MS Word 2019
- MS SQL Server Management Studio 2018
- Visual Studio 2019

Vi benytter anledningen til å takke vår veileder Hans-Petter Halvorsen som har bidratt med sine tekstbøker og blogginnlegg. Dette har vist seg å være til stor hjelp for utviklingen av web-applikasjonen.

Vi vil takke vår oppdragsgiver Caverion for samarbeidet. Spesielt vil vi takke våre kontaktpersoner i Caverion, Simen Vrålstad og Kim Roger Haugseter.

Vi vil også rette en stor takk til Pepperl + Fuchs AS som har lånt oss sitt utstyr gjennom hele prosjektperioden. Spesielt vil vi takke Robert Aasen Kverndalen som har kommet med gode tips og råd til løsningen og Agnar Sæland som stilte utstyret til disposisjon.

Det er en forutsetning at leseren har en generell kunnskap om programmering, Structured Query Language og RFID teknologi for å oppnå en komplett forståelse av rapporten.

Porsgrunn, 24.05.21

# Nomenklaturliste

ASCII	American Standard Code for Information Interchange
CRUD	Create, Read, Update, Delete
ER	Entity Relations
HF	High frequency (Høy frekvens)
HID	Human interface device
LF	Low frequency (lav frekvens)
kHz	Kilohertz
MHz	Megahertz
RDBMS	Relational database management system
RFID	Radio Frequency Identification
RS	Recommended Standard (Protocol)
SSMS	SQL Server Management Studio
SQL	Structured Query Language
USN	Universitetet i Sørøst-Norge
UHF	Ultra high frequency (Ultrahøy frekvens)

# Innholdsfortegnelse

<b>Forord .....</b>	<b>4</b>
<b>Nomenklaturliste .....</b>	<b>5</b>
<b>Innholdsfortegnelse .....</b>	<b>6</b>
<b>1 ..Innledning.....</b>	<b>8</b>
1.1 Beskrivelse av prosjektet .....	8
1.2 Prosjekt mål .....	8
1.3 Omfang .....	8
1.4 Rapport struktur .....	9
<b>2 ..Verktøy og metoder .....</b>	<b>10</b>
2.1 Visual Studio og C#.....	10
2.2 RFID .....	10
2.3 ASP.NET .....	10
2.4 Razor pages.....	11
2.5 ADO.NET .....	12
2.6 SQL.....	12
2.7 Code-first approach .....	12
<b>3 ..Systembeskrivelse og maskinvare .....</b>	<b>13</b>
3.1 Systembeskrivelse.....	13
3.2 Maskinvare .....	14
<b>4 ..RFID-applikasjon .....</b>	<b>17</b>
4.1 Krav til RFID-applikasjon .....	17
4.2 Registrering av adgangskort .....	18
4.3 Registrering av RFID-brikke .....	19
4.4 Transaksjon.....	20
<b>5 ..Web-applikasjonen .....</b>	<b>21</b>
5.1 Krav .....	21
5.2 Kodestruktur.....	25
<b>6 ..Database .....</b>	<b>30</b>
6.1 Databasekrav .....	30
6.2 Databasestruktur .....	31
<b>7 ..Kostnader og estimater .....</b>	<b>34</b>
7.1 Utstyrsliste.....	34
7.2 Kostnad.....	35
<b>8 ..Diskusjon .....</b>	<b>36</b>
8.1 Korona situasjonen .....	36
8.2 RFID-applikasjon .....	36
8.3 Web-applikasjon .....	37
8.4 Videre utvikling.....	38

<b>9 ..Oppsummering .....</b>	<b>39</b>
<b>Referanser.....</b>	<b>40</b>
<b>Vedlegg .....</b>	<b>41</b>

# 1 Innledning

I dette kapitlet skal det gis en beskrivelse av prosjektet, mål for prosjektet, det overordnede målet, gruppen og Caverions mål og forventninger til prosjektperioden.

Arbeidet i prosjektet er delt opp i flere aktiviteter som er vist i Vedlegg B. For å holde kontroll på prosjektfremgangen har det blitt utarbeidet en fremdriftsplan som har blitt oppdatert gjennom prosjektperioden, og er vist i Vedlegg C.

## 1.1 Beskrivelse av prosjektet

Caverion har i sitt lokale et lager hvor de har en del kostbare verktøy samt mindre kostbare verktøy. De har tidligere prøvd å implementere ulike systemer for å registrere lån og tilbakelevering av verktøy, som mer eller mindre har vært en manuell prosess. Disse systemene har på sikt sklidd ut over tid på grunn av den manuelle prosessens de ansatte må gjennom for å låne verktøyene. Caverion ønsker nå et mer automatisk system som vil gjøre det enklere for de ansatte å låne og lever tilbake verktøy, men også gi en mer detaljert oversikt til verktøyansvarlig. Prosjektet i sin helhet består av flere små oppgaver og danner grunnlaget til de tre større oppgavene, som til slutt skal sys sammen til en total løsning. Denne Rapporten er ment å bli lest av fagsjefer, oppdragsgiver og gruppemedlemmer. Den skal gi informasjon til beslutningstaker om totalløsningen. Se Vedlegg A for fullstendig prosjektbeskrivelse.

## 1.2 Prosjekt mål

Målet med prosjektet er å lage en totalløsning for automatisk registrering for utlån av verktøy i et verktøylager hos Caverion. For å lage dette blir det satt opp en plan for verktøylageret og en fysisk løsning. For å sørge for en god utførelse av prosjektet har gruppen utarbeidet en gruppeavtale, som definerer hvordan gruppemedlemmene skal opptre ovenfor hverandre og hva slags konsekvenser som må påberegnes ved brudd på gruppeavtalen. Se Vedlegg D.

## 1.3 Omfang

Oppgaven er begrenset til å omfatte utvikling av RFID-applikasjon, web-applikasjon og oppretting av en skybasert database for lagring av data. For prosjektet vil det også bli opprettet brukermanualer for web-applikasjonen, desktop-applikasjonen og koblingen av systemet. Det vil være nødvendig for administratoren å ha generell kunnskap om Azure SQL, slik at databasen enkelt kan vedlikeholdes.



## 1.4 Rapport struktur

Følgende liste vil gi innblikk i hva kapitlene i denne rapporten handler om, og gi en kort introduksjon.

- Kapittel 2 beskriver kort teoretisk hvilke verktøy og metoder som er brukt i dette prosjektet.
- Kapittel 3 gir en beskrivelse på hvordan systemet fungerer og maskinvaren som utgjør systemet.
- Kapittel 4 forklarer hvordan RFID-applikasjonen fungerer og hvilket krav som gir utgangspunktet for applikasjonen.
- Kapittel 5 omfatter web-applikasjonen og websidenes funksjonalitet.
- Kapittel 6 gir en oversikt over databasen og hvilke tabeller databasen består av.
- Kapittel 7 dekker det økonomiske aspektet for dette prosjektet.
- Kapittel 8 er diskusjonskapittelet som tar for seg problemer og begrunnelse for valgte løsninger.
- Kapittel 9 Konkluderer resultatet for prosjektet.

## 2 Verktøy og metoder

I dette kapitlet skal de gis en kortfattet introduksjon til de ulike verktøyene og metodene som blir tatt i bruk i dette prosjektet. Gruppen har valgt å bruke disse virkemidlene Visual Studio, C#, RFID, ASP.NET, Razor Page, ADO.NET, SQL og Code-First approach. Begrunnelse for valg av virkemidler er beskrevet i hvert av underkapitlene.

### 2.1 Visual Studio og C#

For dette prosjektet har gruppen valgt å bruke Visual Studio som utviklingsmiljø. Valget falt på Visual Studio først og fremst fordi dette er utviklingsmiljøet som har blitt brukt gjennom hele studietiden, men også fordi Caverion ønsket en løsning som er minst mulig lisensbasert. Microsoft Visual Studio er et utviklingsmiljø laget av Microsoft for .NET plattformen. Visual studio gjør det enkelt for brukeren å kunne lage Windows Forms, websider eller mobilapper. C# (C sharp) er et objektorientert programmeringsspråk utviklet av Microsoft som en del av .NET rammeverket. C# er bygget på kombinasjonen mellom programmeringsspråkene C++ og Java.[1]

### 2.2 RFID

Tidlig i prosjektet ble det diskutert hva slags teknologi som ville bli brukt i systemet for å registrere verktøyene. Gruppen har valgt å bruke RFID (Radio-frequency identification) framfor annen teknologi på grunn av kravet om at utlånsprosessen skal være så automatisk som mulig. RFID (Radio-frequency identification) er en teknologi som muliggjør lesing og skrivning til radio mottakere ved hjelp av elektromagnetiske felt. Radio mottakerne kan enten være passive eller aktive. Passive mottakere får energien sin fra en RFID-leser som sender ut et elektromagnetisk felt. Aktive mottakere drives av en ekstern strømkilde som for eksempel et batteri, og vil kunne leses på mye større avstander enn en passiv mottaker.[2] Utover i denne rapporten vil radio mottakerne bli betegnet som RFID-brikker.

### 2.3 ASP.NET

ASP.NET er en webutviklings plattform utgitt av Microsoft. Som beskrevet tidligere i 2.1 har gruppen erfaring med Microsoft sin plattform fra studietiden. Derfor ble det naturlig å velge ASP.NET. Denne plattformen blir brukt til å lage nettsider og applikasjoner. ASP.NET støtter C# som er programmeringsspråket som brukes i dette prosjektet. I dette prosjektet blir ASP.NET Core brukt. ASP.NET Core er en ny versjon av ASP.NET. ASP.NET Core kan kjøres på alle databehandlingsplattformer som Windows, MacOS og Linux som gjør det mulig å kunne utvikle program uansett hvilken databehandlingsplattform gruppe-medlemmene eventuelt skulle bruke.[3] Høy hastighet, lav pris og enorm språkstøtte er de viktigste fordelene med ASP.NET.

## 2.4 Razor pages

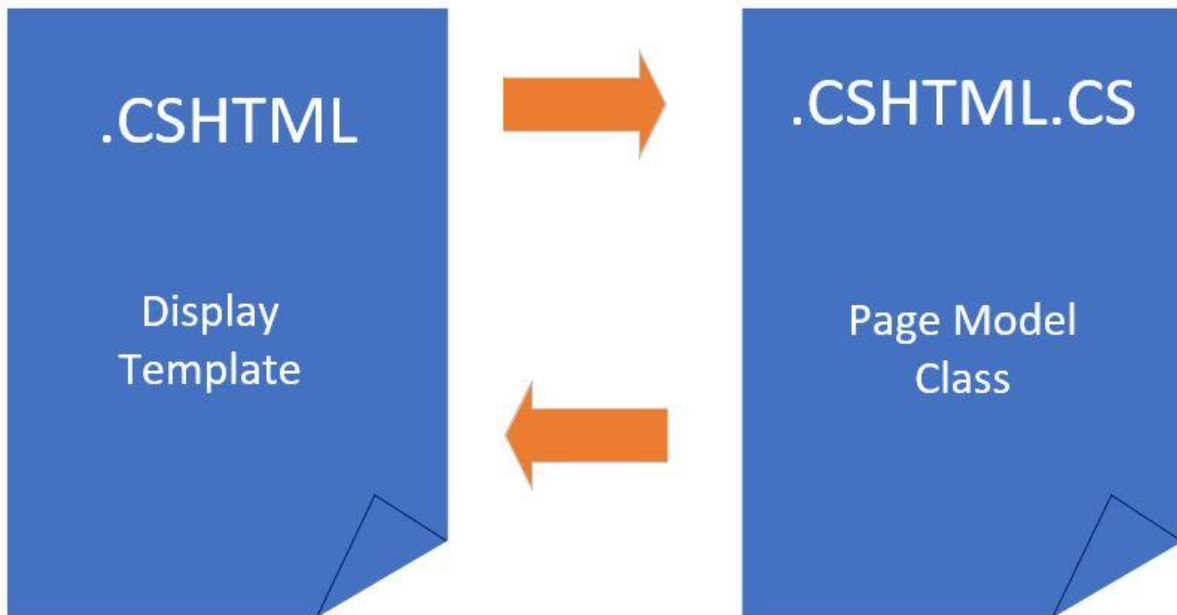
Gruppen valgte å bruke Razor Page fra anbefaling fra veileder. Razor Page er ny teknologi for å lage websider raskere og mer effektivt. Razor page ble introdusert i ASP.NET Core 2.0. Razor page gir utvikleren full kontroll over den ferdige HTML-websiden. Razor page har likheter til ASP.NET webform rammeverket. I ASP.NET brukes en aspx side og en kode-bakside. Aspx siden inneholder HTML og kontrollerer den visuelle delen av web-applikasjonen. Kode-baksiden inneholder sidens C# syntaks eller Visual Basic kode som behandler hendelsene som skjer på siden.

En Razor Page er et sett med filer. En Razor Page med navn Index, vil ha en fil med navn Index.cshtml som er den visuelle siden som inneholder HTML og Razor syntaksene for Razor Page siden. Den andre filen vil ha navnet Index.cshtml.cs som er sidemodellens klasse som inneholder sidens C# kode og behandler sidens hendelser og dataen som den visuelle siden trenger. Figur 2-1 beskriver oppbygningen til en Razor page.



Figur 2-1: Razor page

Som nevnt tidligere er Razor Page et sett med filer, .CSHTML og .CSHTML.CS, som bygger opp rammeverket, vist i Figur 2-2. Filen inneholder HTML koden og Page Model-Class filen inneholder sidens C# kode. All koden som blir skrevet på sidemodellklassen har en relevans til den visuelle side filen. Den visuelle side filen og sidemodellklassen er på ett sted og nært relaterte til hverandre dette gjør det til en effektiv måte å lage websider. Konklusjonen fra Microsoft er at en burde bruke Razor Pages når en skal bygge «web UI». [4]



Figur 2-2: Razor Page Design

## 2.5 ADO.NET

Gruppen bruker ADO.NET for kommunikasjon mot database. ADO.NET er et rammeverk for å kommunisere mellom C# og databasen. Ved å laste ned en NuGet pakke til Visual Studio prosjektet ble det mulig å kommunisere med databasen. Denne NuGet pakken er «System.Data.SqlClient».[5]

## 2.6 SQL

Structured Query Language er et kontrollgrensensnitt som blir brukt til å kjøre operasjoner i Databaser. I dette prosjektet har gruppen brukt databaseverktøyet Microsoft SQL Server Management Studio (SSMS). SSMS har blitt lokalt lagret for testing opp mot prototype programmet. SSMS har god kompatibilitet mot Microsoft Visual Studio og ASP.NET, som er grunnet til at valget falt på dette verktøyet. Microsoft er selskapet bak begge disse programmene som gjør at de jobber godt sammen. Databasen bruker tabeller for å utgjøre de forskjellige nødvendige prosedyrene i programmet.[6] Microsoft Azure, ofte referert til som Azure, er en sky tjeneste skapt av Microsoft for å bygge, teste, distribusjon og administrasjon av applikasjoner og tjenester gjennom Microsoft-administrerte datasentre.[7]

## 2.7 Code-first approach

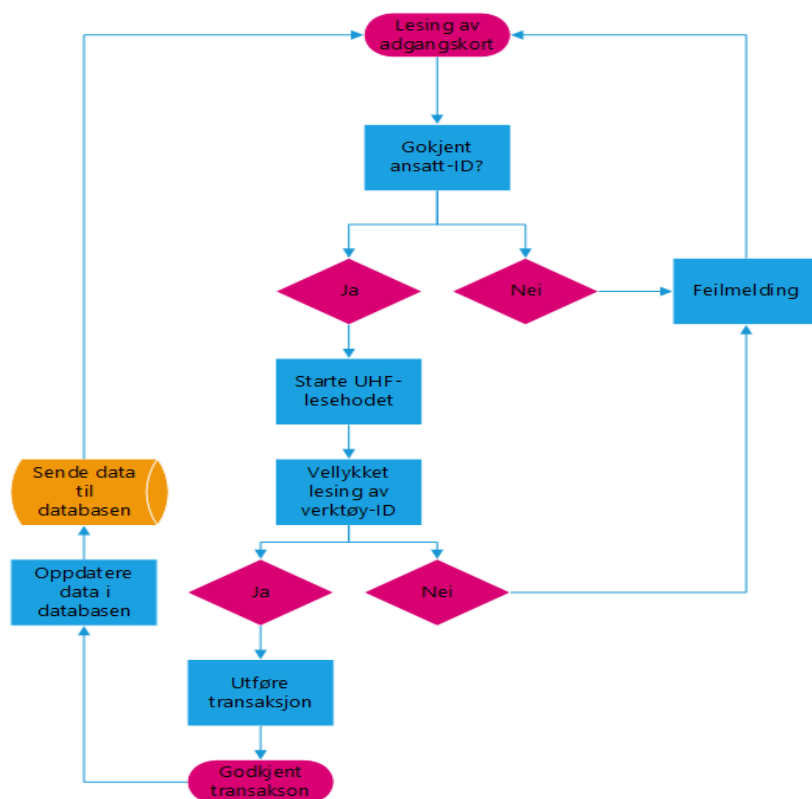
Gruppen har brukt Code-First approach ved utvikling av web-applikasjonen. Code-first er en metode som ble introdusert i Entity Framework 4.1 hvor det er fokus på å utvikle klassene til domenet til applikasjon før designet til databasen. Når klassene til domenet er utviklet så utvikles klassene for tilegning av informasjonen i databasedesignet.[8] Utviklingsflyten i Code-first er: Opprett domeneklassene, konfigurere disse klassene til å passe sammen med database designet ditt, opprett eller oppdatere databasetabellene ved hjelp av automatisk migrering eller kodebasert migrering.

### 3 Systembeskrivelse og maskinvare

I dette prosjektet skal det lages et lagerstyringssystem for å holde kontroll på Caverions verktøylager. Lagersystemets funksjon er å gjøre det enkelt for de ansatte å låne verktøy, men også gjøre det enklere for verktøyansvarlig å holde kontroll på hvilket verktøy som er på lager eller ikke. De ansatte låner verktøyet ved å først registrere sitt adgangskort og går ut med verktøyet de ønsker å låne. Lagerstyringssystemet slår sammen dataen fra adgangskortet og verktøybrikken for å lage en utlånsordre. Når utlånsordren er opprettet, vil verktøyet være registrert på den ansatte som hentet det ut.

#### 3.1 Systembeskrivelse

I dette underkapittelet skal det gis en kortfattet og enkel forklaring på hvordan applikasjonene fungerer. Systemet består av to applikasjoner, en desktop-applikasjon, en web-applikasjon og et RFID-system bestående av to lesere. Desktop-applikasjonen tar seg av innlevering og utlån av verktøy. Her vil det være to RFID-lesere, en som leser adgangskort og en som leser RFID-brikke på verktøyene. Desktop-applikasjonen vil enten gi en feilmelding ved en ugyldig lesing eller fortsette til transaksjonen, som vist i Figur 3-1. Når desktop-applikasjonen har sendt data til databasen, vil den kunne leses av på web-applikasjonen. I web-applikasjonen fremstilles dataen i form av to tabeller, verktøy-tabell som viser alle verktøyene og en ansatt-tabell, se kapittel 6 for mer detaljert beskrivelse. Web-applikasjonen har også muligheten til å legge til, slette og redigere verktøy og ansatte. Videre i denne rapporten vil desktop-applikasjonen bli omtalt som RFID-applikasjon.



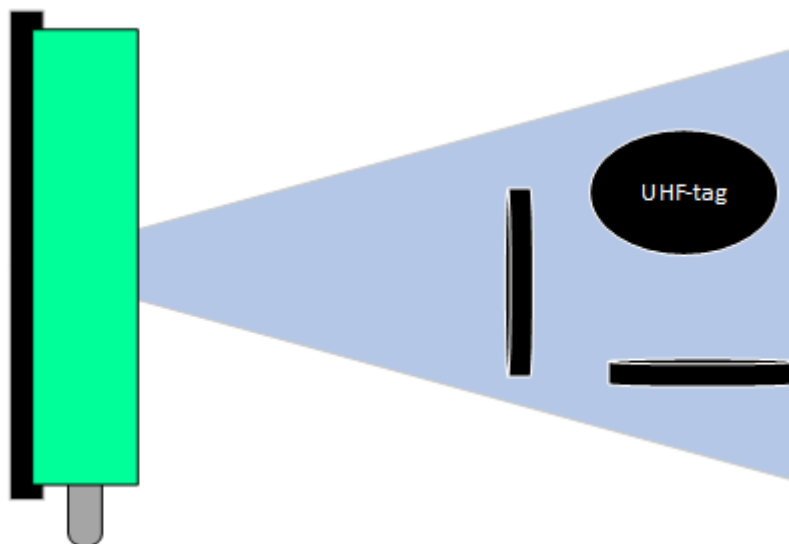
Figur 3-1: Flytskjema som viser hovedfunksjonene i systemet.

## 3.2 Maskinvare

Systemet kreve flere ulike komponenter for å fungere. Siden dette prosjektet omhandler RFID-teknologi, har gruppen valgt å benytte seg av UHF (Ultra high frequency) og LF (Low frequency) RFID-lesere. For å kjøre programmet brukes det en mini-PC og en storskjerm for å grafisk fremstille data. Det vil også være en skjerm inne på lageret som viser RFID-applikasjonen.

### 3.2.1 Pepperl + Fuchs UHF RFID-leser

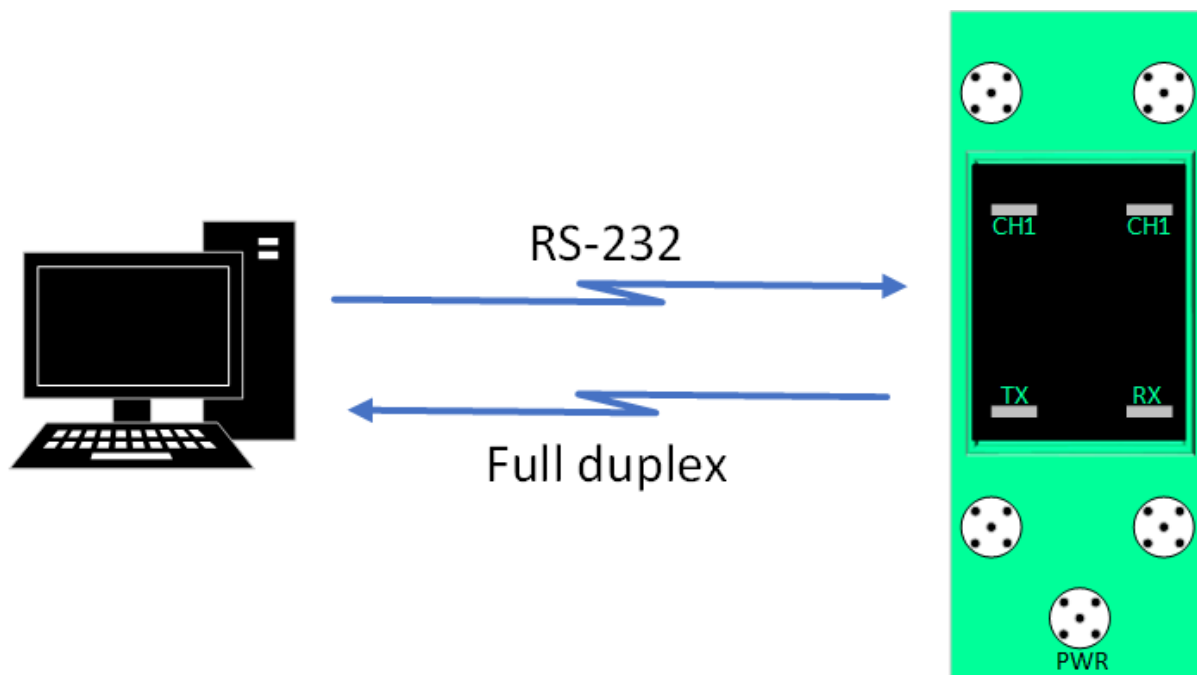
IUH-F190-V1-FR1-01 er et UHF RFID-leser som bruker ultrahøy frekvens for å lese og skrive til RFID-brikkene, og har en oppgitt leseavstand på inntil 2 meter. RFID-leseren kan bare lese ultrahøy frekvente brikker, som også løste det mulige problemet med avlesninger av adgangskort samtidig som verktøybrikkene. Siden leseren sender ut signaler i en konisk-form, vil det være lettere for RFID-leseren å oppdage brikkene uavhengig av orientering på brikkene.[9] Dette blir illustrert i Figur 3-2. I Dette systemet vil Pepperl + Fuchs RFID-leseren være den leseren som tar seg av lesing av verktøybrikkene. Siden den har en stor leseavstand, vil det kun være nødvendig å stå i nærheten for at brikken skal bli lest. UHF RFID-leseren fra Pepperl + Fuchs har mange avanserte innstillinger, som gjør det mulig å kunne justere leseavstanden og lesehyppigheten etter behov.



Figur 3-2: Kjegleformet elektromagnetisk felt som leser brikkene i ulike posisjoner.

### 3.2.2 Pepperl + Fuchs IDENTControl

IDENTControl er et kontrollgrensesnitt som muliggjør kommunikasjon mellom datamaskinen og RFID-leseren. I dette systemet vil kontrollgrensesnittet fungere som kjernen, da den er midtpunktet for kommunikasjon mellom RFID-leser og datamaskinen. Kontrollgrensesnittet bruker RS-232 (Recommended standard) som er en standard protokoll for kommunikasjon.[10] RS-232 er «full duplex», noe som betyr at kontrolleren og datamaskinene kan kommunisere samtidig uten noen forstyrrelser som vist i Figur 3-3.



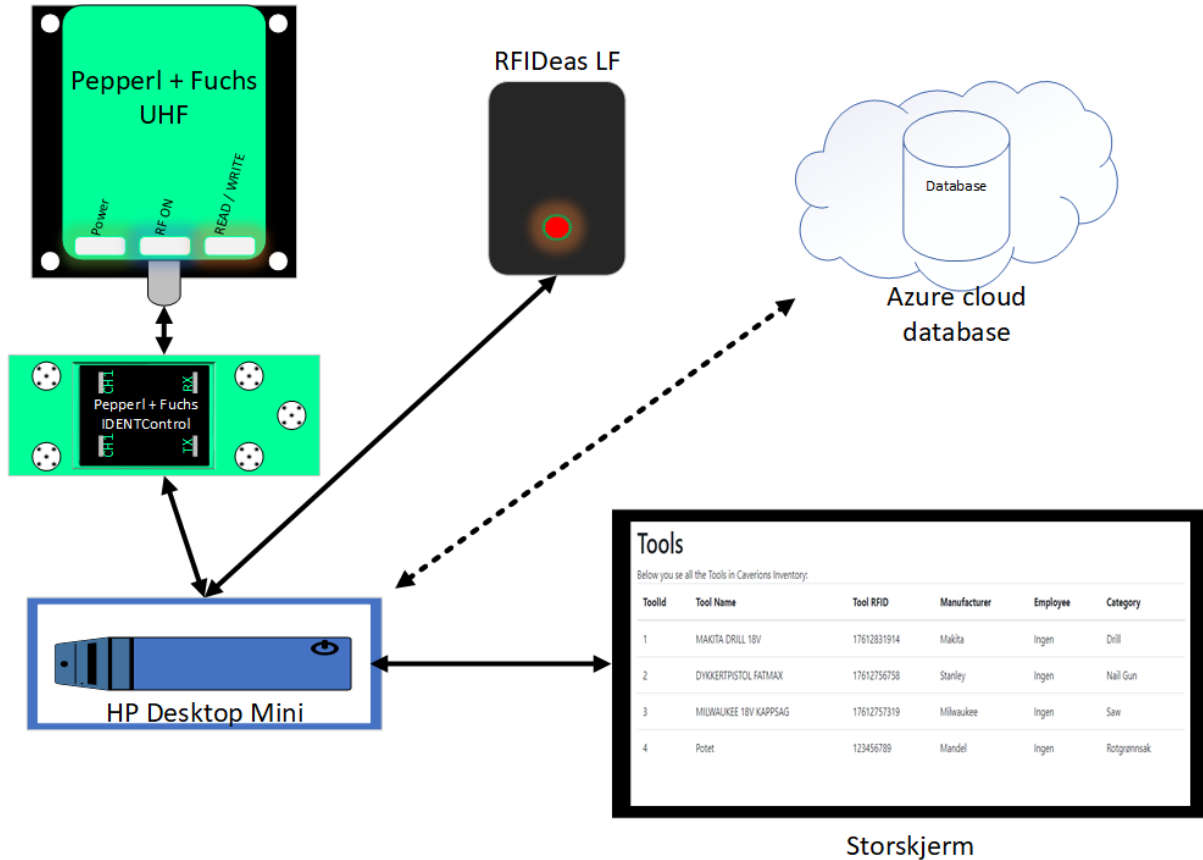
Figur 3-3: Kommunikasjon mellom datamaskin og IDENTControl med RS-232

### 3.2.3 rfIDEAS RFID-leser

Wave-ID Plus er en RFID-leser som har dobbel driftsfrekvens, som betyr at den leser frekvenser på 125 kHz (Kilo Hertz) og 13,56 MHz (Mega Hertz).[11] RFID-leseren kommuniserer over USB (Universal Serial Bus) og fungerer som et tastatur og blir betegnet som en HID (Human Interface Device). Den leser inn data fra brikker som en input, fremstiller det grafisk i et valgt skrivefelt som en output med et Enter-trykk. Ved bruk av en leser som fungerer som en HID fjernes behovet for å lese data via seriell kommunikasjon. For systemet vil det også være lettere å behandle, og sammenligne data siden det kommer ut som rene tekststrenger og ikke ASCII-strenger som må konverteres. I dette prosjektet vil RFID-leseren være ansvarlig for å lese inn data fra adgangskortene til de ansatte. Når RFID-leseren leser en gyldig ansatt-ID, vil den sende en bekreftelse til UHF RFID-leseren via kontrollgrensesnittet slik at det kan lese verktøybrikkene.

### 3.2.4 HP Desktop Mini

HP Desktop mini 400 G6 er en kraftig, kompakt datamaskin med rask prosessor og et integrert skjermkort.[12] I dette systemet vil datamaskinen bli brukt til datainnsamling fra Pepperl + Fuchs og rFIDEAS RFID-leserne, kjøre RFID-applikasjonen, fremstille web-applikasjonen til skjerm og være koblet opp mot den skybaserte databasen. Figur 3-4 viser hvordan datamaskinen samhandler med resten av maskinvaren i systemet.

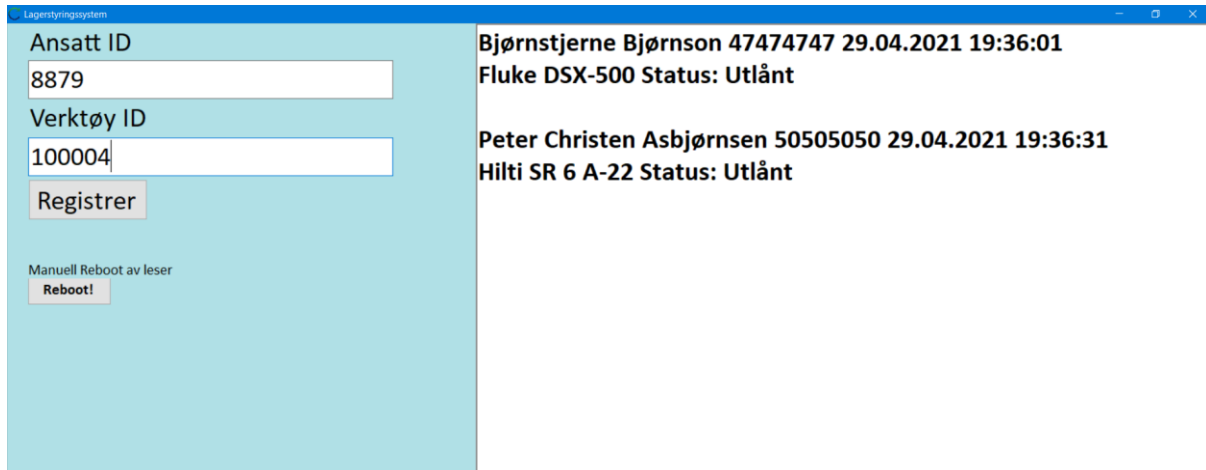


Figur 3-4: Viser samhandling mellom system og maskinvare



## 4 RFID-applikasjon

RFID-applikasjonen er ansvarlig for å lese adgangskort, registrere utlån, detektere RFID-brikke og fullføre transaksjoner. Applikasjonen vil også ha indirekte ansvar for å holde styr på verktøylageret, da den registrerer alle utlånte og leverte verktøy. Data utveksles mellom Azure databasen og applikasjonen, og fremstilles grafisk på en liten skjerm inne på verktøylageret. RFID-applikasjonen er utviklet i Visual Studio og blir kjørt på en kraftig mini desktop fra HP, som igjen er koblet opp mot en Azure database. Applikasjonen kan brytes ned til tre hovedkomponenter som blir nærmere beskrevet i underkapittel 4.2 til 4.4. Se Vedlegg F for brukermanual til RFID-applikasjonen



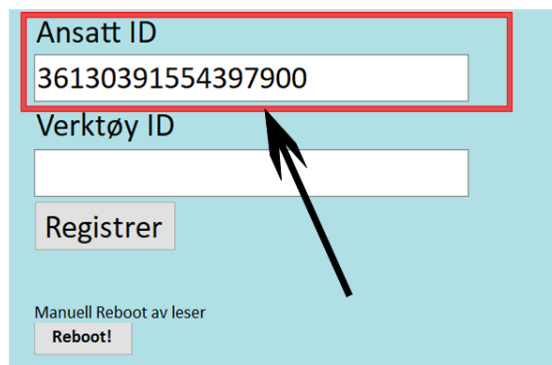
Figur 4-1: Brukergrensesnittet for RFID-applikasjon.

### 4.1 Krav til RFID-applikasjon

Fra Caverion sin side var det bestemt at applikasjonen skulle registrere verktøy som blir tatt inn og ut av lageret ved hjelp av sensortechnologi. Valget falt på RFID, da det har mange bruksområder og er forholdsvis enkelt å implementere. For å automatisere utlånsprosessen, var det et krav om at RFID-leseren måtte ha relativt stor leseavstand, slik at den kan detektere en brikke som passerer forbi. For å unngå feillesing må styrken til RFID-leseren nedjusteres, slik at leseavstanden reduseres fra to meter til én meter.

## 4.2 Registrering av adgangskort

For å lese adgangskortene bruker programmet en lavfrekvent RFID-leser som kommuniserer med datamaskinen via USB. Dette er nærmere forklart i kapittel 3.2.3. Når en ansatt registrerer kortet sitt ved RFID-leseren, vil det bli lagt inn i tekstboksen for ansatt-ID som vist i Figur 4-2. Tekststrengen fra adgangskortet, som gis av RFID-leseren vil bli skrevet inn i tekstboksen. Applikasjonene vil deretter sjekke ansatt-iden opp mot databasen for å se om den eksisterer, hvis den ikke eksisterer kommer det opp en feilhåndtering i form av en feilmelding som må anerkjennes av en ansatt. Når applikasjonen godkjenner ansatt-iden, vil tekstboksen bli satt i «Read-only» modus som gjør at det ikke kan skrives til, og med det unngå uønskede inntastinger. Etter inntastingen legger RFID-leseren inn et ENTER-trykk som vil fungere som en bekreftelse på at alt er godkjent. Applikasjonen vil starte en nye «Thread», som er uavhengig streng med kode som kan kjøre ved siden av den kjørende koden. Den nye Threaden håndterer skriving til og lesing fra den ultrahøy frekvente RFID-leseren via seriell kommunikasjon.



Ansatt ID  
36130391554397900

Verktøy ID

Registrer

Manuell Reboot av leser  
Reboot!

Figur 4-2: Tekstboks med ansatt-ID fylt inn

### 4.3 Registrering av RFID-brikke

Når en ansatt-ID har blitt godkjent og registrert, vil den nye Threaden starte lesingen av verktøybrikkene. Selve lesingen ble i starten av utviklingsfasen kjørt på samme Thread som resten av programmet, men på grunn av at programmet brukte for lang tid på å lese, falt valget på å kjøre lesingen på en separat Thread. Når systemet skal lese en brikke, må det først sendes en kommando til RFID-leseren om at den skal lese. To av hovedkommandoene for lesing er SR – Single Read og ER – Enhanced Read. Single Read vil være avhengig av at brikken er innenfor rekkevidden til RFID-leseren, da det kun sender ut en enkel puls for å lese brikken, mens Enhanced Read leser kontinuerlig til den detekterer en brikke. Valget falt på Enhanced Read funksjonen da den gir den mest pålitelige lesingen. Dette forklares nærmere i kapittel 3.2.1. Når RFID-leseren mottar kommandoen om å lese, vil den lese til en brikke har passert området som leses. RFID-leseren detekterer brikken og sender en respons tilbake til systemet i form av en tekststreng bestående av segmenter med bits. Disse fire segmentene utgjør all dataen brikken sender tilbake til RFID-leseren. Responsen starter med et segment bestående av et BIT som er statusen på om lesingen var vellykket eller ikke, der 0 definerer en vellykket lesing og 5 definerer en mislykket lesing. Neste segment består av et BIT som definerer hvilken kanal som brukes, her brukes kanalen som er satt fra fabrikkinnstillingene til 0. Tredje segmentet er selve dataen du ønsker å lese fra brikken, og den kan være så stor som lagringsplassen på brikken tillater. Lagringsplassen på brikkene vil variere ut ifra type og behov. For dette prosjektet er det valgt en seksifret tallkombinasjon som starter på 100001, som blir vist i Figur 4-3

Ansatt ID  
36130391554397900

Verktøy ID  
100004

Registrer

Manuell Reboot av leser  
Reboot!

Figur 4-3: Tekstboks med ferdig innlest ansatt-ID og Verktøy-ID

Det er nevnt i underkapittel 4.1 at innlesningen av verktøy skal være automatisk som mulig, noe som gjør at Figur 4-2 og Figur 4-3 kan virke misvisende siden begge viser en knapp for registrering. Denne knappen er kun for manuell registrering av verktøy som f.eks. det ikke har blitt registret utlån eller innlevering på.

## 4.4 Transaksjon

Når ansatt-ID og verktøy-ID har blitt godkjent, starter registreringen av utlånet. I databasen har hvert verktøy en variabel som definerer om verktøyet er utlånt eller levert. For et verktøy vil det i Borrowed-raden til verktøy-tabellen stå enten 0 eller 1. Borrowed-raden er av datatypen BIT som vil si at 0 er False mens 1 er True. Programmet vil gjøre en sjekk om verdien er satt til 0 eller 1, det vil si at et verktøy som blir levert vil gå fra å ha 1 til 0 som verdi, og omvendt ved utlån. Transaksjonen blir vist i en tekstboks som kun kan leses fra, og oppdaterer seg ved å vise siste transaksjonen i bunn. Når transaksjon er ferdig utført vil UHF RFID-leseren gå fra å lese kontinuerlig til en standby-modus.



Ansatt ID	Jan-Erik Lunde 09.05.2021 21:04:49
<input type="text"/>	Hilti Te 6-A22 Status: Utlånt
Verktøy ID	Nickolas Helgeland 09.05.2021 21:05:21
<input type="text"/>	Elpress LV1300B Status: Utlånt
Registrer	Jan-Erik Lunde 09.05.2021 21:26:40
Manuell Reboot av leser	Hilti Te 6-A22 Status: Levert
Reboot!	Simen Vrålstad 09.05.2021 21:27:06
	Phoenix Contact CUTFOX 101 Status: Utlånt
	Kim Roger Haugseter 09.05.2021 21:27:38
	Phoenix Contact CUTFOX 100 Status: Utlånt

Figur 4-4: Utførte transaksjoner for levering og utlån

## 5 Web-applikasjonen

Dette kapitlet vil ta for seg web-applikasjonen som er utviklet i samhold med prosjektet. Applikasjonen har en startside som inneholder listen over verktøy, en side med liste over ansatte, en side for opprettelse av verktøy og en side for opprettelse av ansatte. For brukerveiledning av web-applikasjon se Vedlegg E.

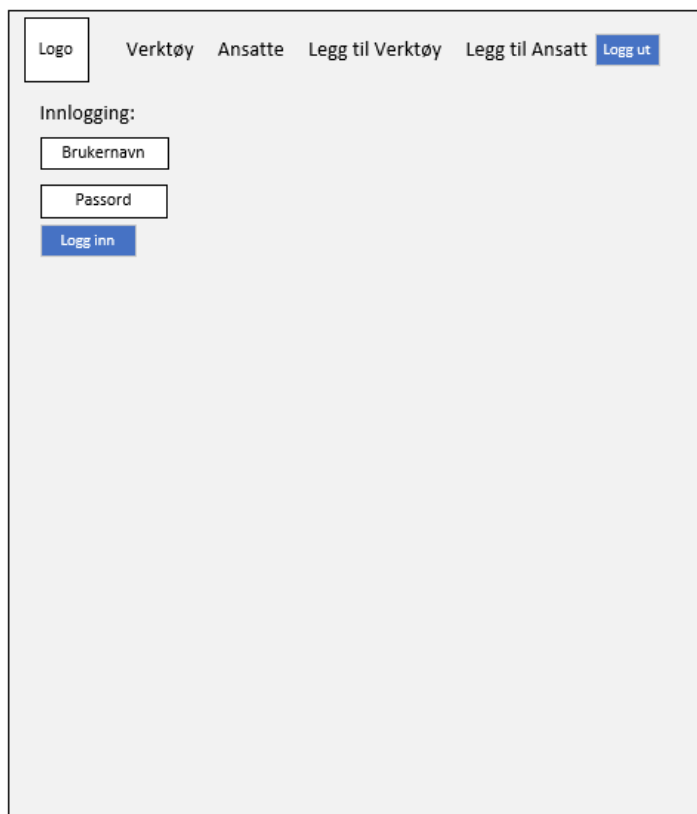
### 5.1 Krav

I dette underkapitlet skal leseren få en forståelse av brukerens behov og kravene for web-applikasjonens funksjoner. Kravene skal gjøre brukeropplevelsen så god som mulig.

Caverion ønsket en web-applikasjon med en oversiktlig liste av sine verktøy. Videre ble det diskutert funksjoner for web-applikasjonen for å gjøre den mer brukervennlig. Listen skulle gi en oversikt over om verktøyet var utlånt, hvilken person som lånte verktøyet og muligheten til å få opp mer informasjon. Det var ønsket at det var mulig å legge til filtrering etter modell, kategori og hvilken ansatt som lånte verktøyet. Det var også et ønske om søkefunksjon for å kunne søke etter verktøykategori, modell navn og utlåner.

#### 5.1.1 Innloggingsside

Ved planleggingen for web-applikasjonen var det et ønske fra Caverion at det skulle lages en innloggings funksjon for å begrense utvending tilgang til applikasjonen. Forutsetningen for designet ble et enkelt brukergrensesnitt for innlogging. Brukernavn- og passordfelt og knapp til innlogging. Her var det også viktig å legge til et skille mellom sluttbruker og administratorbruker. Hvor sluttbrukeren har begrenset tilgang til sidene og administratorbrukeren har full tilgang til sidene. Figur 5-1, er en planskisse for designet av innloggingssiden til web-applikasjonen.

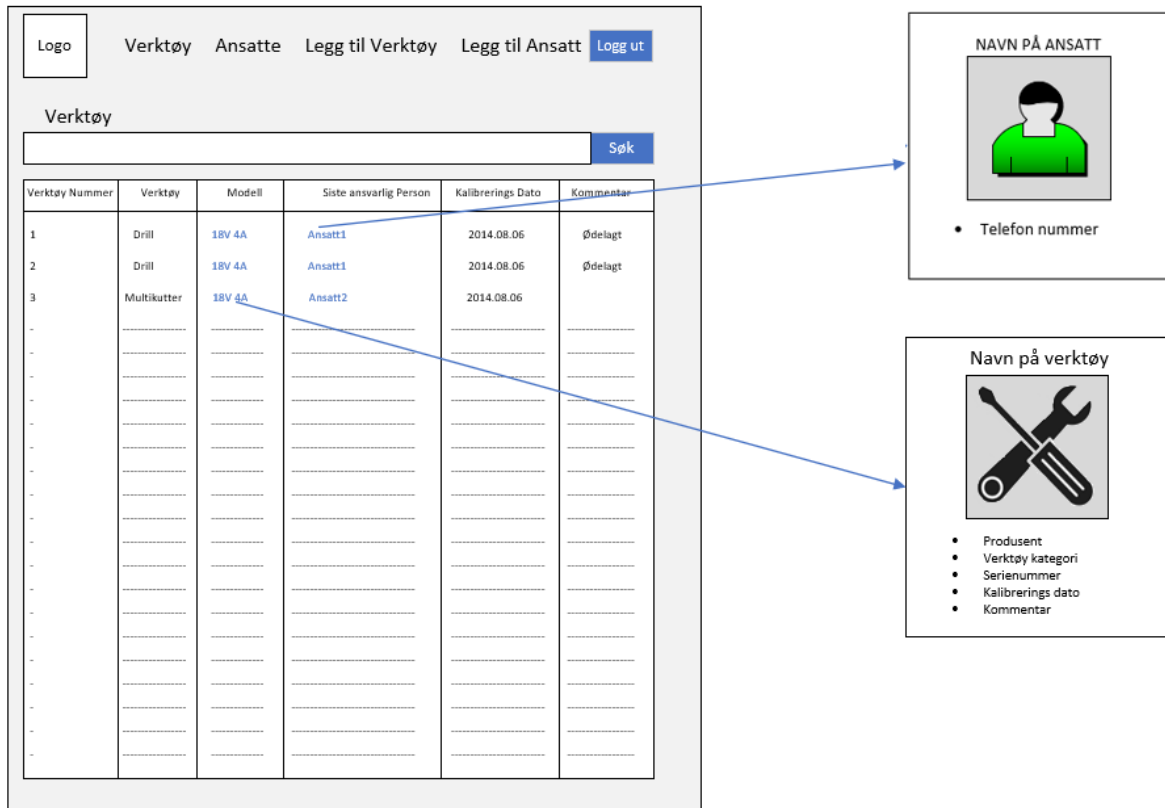


Figur 5-1: Planskisse innloggingside

### 5.1.2 Verktøyside

Ved planleggingen for web-applikasjonen var det et ønske fra Caverion at det skulle lages en oversiktlig liste av alle verktøyene, med mulighet til å se hvem som låner de enkelte verktøyene. Forutsetningen for dette designet ble en liste med den nødvendige informasjonen til verktøyene. Videre fra denne listen skal det være mulig å kunne klikke på ansatt eller verktøy for å få mer informasjon. Figur 5-2, er en planskisse for designet av verktøysiden til web-applikasjonen.

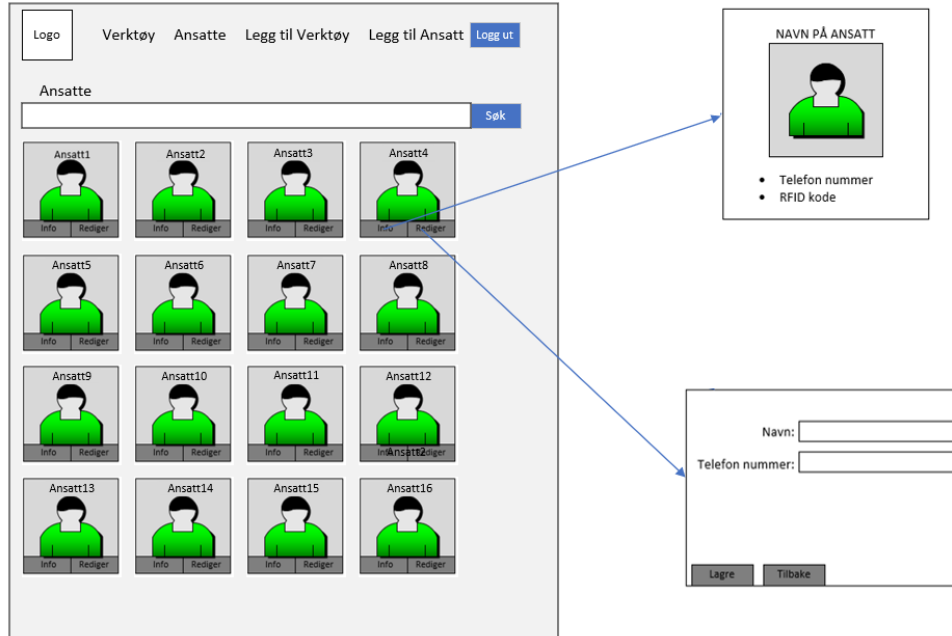
# Web-applikasjonen



Figur 5-2: Planskisse Verktøyside

### 5.1.3 Ansattside

Det ble diskutert om en ansattliste kunne være nyttig på web-applikasjonen. Ansattlisten måtte gi en god oversikt over de ansatte, med mulighet til å kunne få mer informasjon av den enkelte ansatte. Figur 5-3, er planskissen for designet av ansattside til programmet.



Figur 5-3: Planskisse Ansattside

### 5.1.4 Legge til sidene

Web-applikasjonen skal også ha mulighet til å legge til ansatte og verktøy. Her må det lages to sider hvor den ene tar for seg å legge til ansatte og den andre tar for seg å legge til verktøy. Med tanke på hvordan designet til disse sidene skulle være, var det et krav at det skulle være enkelt å bruke. Designet vil gi administratorbrukeren mulighet til å legge til all nødvendig informasjon av ansatte og verktøy i felt med angitte navn. Figur 5-4, er planskissen for designet av opprettelsessiden til programmet.



The image shows a wireframe for a web application's 'Opprett' (Create) page. At the top left is a 'Logo' placeholder. To its right are navigation links: 'Verktøy', 'Ansatte', 'Legg til Verktøy', 'Legg til Ansatt', and a blue 'Logg ut' button. Below this is the main heading 'Opprett'. The form consists of 12 vertical input fields, each labeled 'Informasjon1' through 'Informasjon12'. At the bottom left of the form are two buttons: 'Lagre' and 'Avbryt'.

Figur 5-4: Planskisse Opprettside

## 5.2 Kodestruktur

Et mål for web-applikasjonen er at den skal være utvidbar. For å oppnå dette målet må kodestrukturen fungere som et solid fundament for fremtidig utvikling. Det er også viktig at koden blir skrevet slik at den er forståelig for andre og kommentert hvor det ikke er innlysende hva funksjonen til koden er. I dette underkapitlet vil høyere nivå av struktur bli visualisert og deres underliggende filosofi bli forklart. Målet med dette kapitlet er at leseren skal få en grunnleggende forståelse av hvordan applikasjonen fungerer og hvordan videre utvikle den på riktig måte.

### 5.2.1 Oversikt

Applikasjonen ble designet med «Code-First Approach». Applikasjonen har to klassebibliotek i prosjektet, `IToolInventory.Models` og `IToolInventory.Services`. Alt av enheter legges under `IToolInventory.Models` som ansatt og verktøy klassen.

## 5.2.2 Klassebibliotek

IToolInventory.Models inneholder enhetene Tool.cs og Employee.cs. Disse klassene danner informasjons strukturen til verktøy og ansatt. Se Figur 5-5 og Figur 5-6 for strukturen av klassene.

```
public class Tool
{
    [Required(ErrorMessage = "RFID Kode kreves")]
    public long ToolId { get; set; }
    [Required(ErrorMessage = "verktøynummer kreves")]
    public int ToolNumber { get; set; }
    [Required(ErrorMessage = "Model navn kreves")]
    [StringLength(20, MinimumLength = 3, ErrorMessage = "Maksimal lengde 20 bokstaver")]
    public string? Model { get; set; }
    public bool Borrowed { get; set; }
    [StringLength(20, MinimumLength = 3, ErrorMessage = "Maksimal lengde 20 bokstaver")]
    public string? Manufacturer { get; set; }
    [Required(ErrorMessage = "Verktøy kategori kreves")]
    [StringLength(20, MinimumLength = 3, ErrorMessage = "Maksimal lengde 20 bokstaver")]
    public string Category { get; set; }
    public long EmployeeId { get; set; }
    [StringLength(30, MinimumLength = 3, ErrorMessage = "Maksimal lengde 30 tall")]
    public string? EmployeeName { get; set; }
    public string? Description { get; set; }
    [StringLength(20, MinimumLength = 3, ErrorMessage = "Maksimal lengde 20 bokstaver")]
    public string? SerialNumber { get; set; }
    public DateTime? DateOfPurchase { get; set; }
    public int? Reminder { get; set; }
    [Range(0, int.MaxValue, ErrorMessage = "Pris må være et nummer")]
    public string? Price { get; set; }
    [StringLength(10, MinimumLength = 0, ErrorMessage = "Maksimal lengde 10 bokstaver")]
    public string? Condition { get; set; }
    public DateTime? DateOfCalibration { get; set; }
    public string? CalibrationPlace { get; set; }
    public string? PhotoPath { get; set; }
}
```

Figur 5-5 Tool.cs

```
public class Employee
{
    [Required(ErrorMessage = "RFID kode kreves")]
    public Int64 EmployeeId { get; set; }
    [Required(ErrorMessage = "Ansattnummer kreves")]
    public int EmployeeNumber { get; set; }
    [Required(ErrorMessage = "Navn kreves")]
    [StringLength(32, MinimumLength = 0, ErrorMessage = "Maksimal lengde 32 bokstaver")]
    public string EmployeeName { get; set; }
    [Required(ErrorMessage = "Telefonnummer kreves")]
    [RegularExpression(@"^\d{3}\d{2}\d{3}$", ErrorMessage = "Telefonnummer er ikke fullverdig")]
    [StringLength(15, MinimumLength = 0, ErrorMessage = "Maksimal lengde 15 tall")]
    public string PhoneNumber { get; set; }
    public string? PhotoPath { get; set; }
}
```

Figur 5-6 Employee.cs

Alle klassene under IToolInventory.Services bruker enhetene fra IToolInventory.Models. Alt med datatilgang legges under IToolInventory.Services. Applikasjonen skal ha CRUD (Create, Read, Update, Delete) funksjonalitet. Under IToolInventory.Services har applikasjonen to klasser som heter IToolRepository og IEmployeeRepository disse klassene er klasser som skal gi en oversikt over funksjonaliteten til data tilgang klassene uten avansert kode. Se Figur 5-7 og Figur 5-8 for struktur over disse klassene.

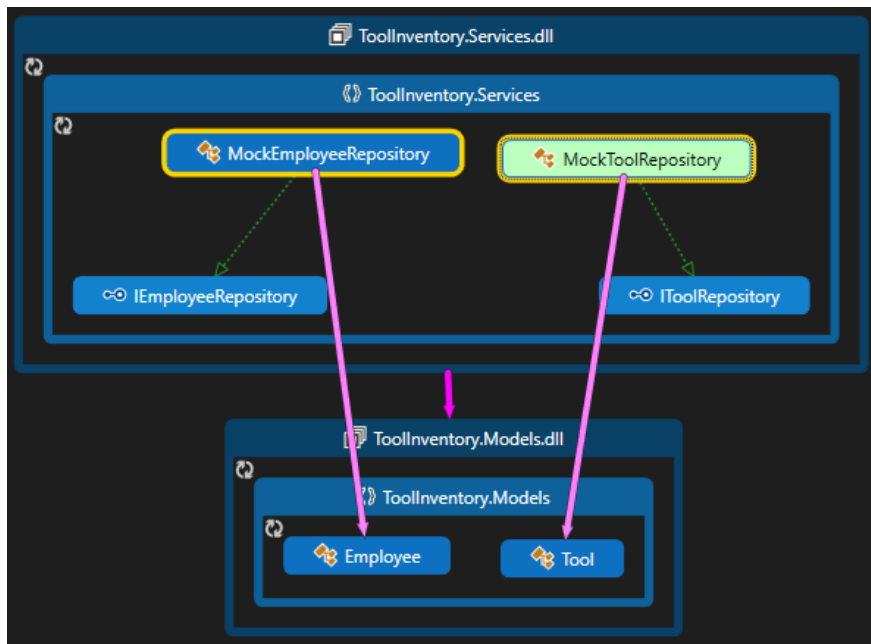
```
public interface IToolRepository
{
    IEnumerable<Tool> Search(string searchTerm)
    IEnumerable<Tool> GetAllTools();
    Tool GetTool(long id);
    Tool Update(Tool updatedTool);
    Tool Add(Tool newTool);
    Tool Delete(long id);
}
```

Figur 5-7 IToolRepository

```
public interface IEmployeeRepository
{
    IEnumerable<Employee> Search(string searchTerm);
    IEnumerable<Employee> GetAllEmployees();
    Employee GetEmployee(long id);
    Employee Update(Employee updatedEmployee);
    Employee Add(Employee newEmployee);
    Employee Delete(long id);
}
```

Figur 5-8 IEmployeeRepository

Som nevnt tidligere ble prosjektet designet med «Code-First Approach» hvor fokuset er på å lage klassene til domenet istedenfor å designe databasen først. Under `IToolInventory.Services` ligger det to klasser til som heter `MockToolRepository` og `MockEmployeeRepository`. I disse klassene ble det hard-kodet inn liste for ansatte og verktøy. Det ble videre utviklet CRUD funksjonalitet av listene. `MockToolRepository` og `MockEmployeeRepository` arvet klassene fra `IToolRepository` og `IEmployeeRepository`. Det er her CRUD funksjonaliteten ble utviklet. Se Figur 5-9, for struktur av disse klassene.

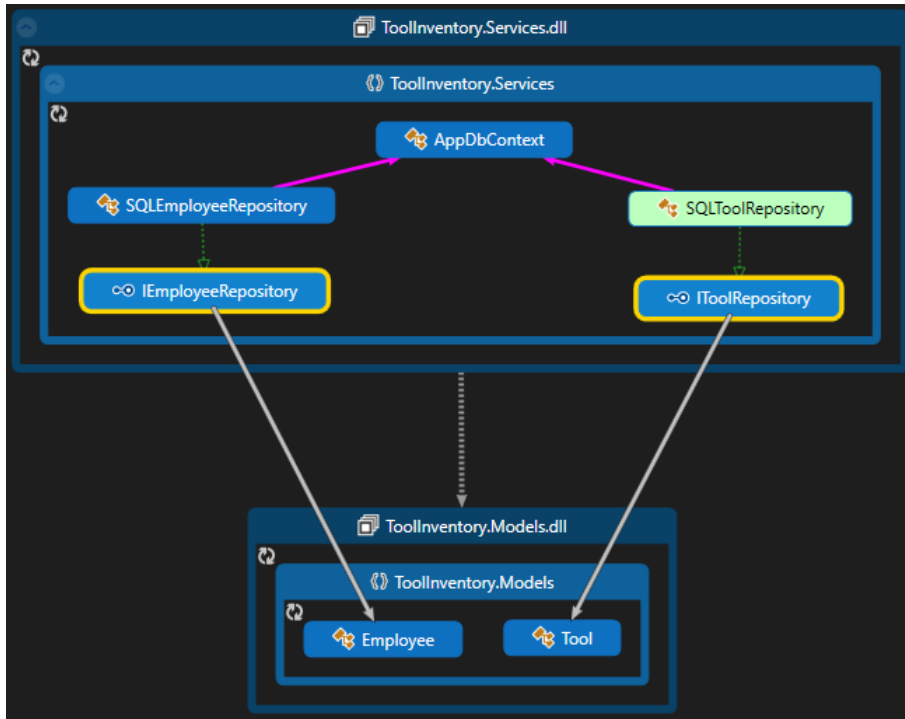


Figur 5-9: Codemap av Mockklasser

Etter CRUD funksjonaliteten har blitt utviklet i MockToolRepository og MockEmployeeRepository var steget å utvikle databasen. Se kapittel 6 for oversikt over databasen. Etter databasen ble utviklet blir klassen AppDbContext utviklet. Denne klassen inneholder databasesettene Employees og Tools.

En connectionstring til databasen bli satt opp i web-applikasjon. I Razor pages blir connectionstringen lagt i appsettings.json filen. Når connectionstringen har opprettet så må AppDbContext og connectionstringen legges til i Startup.cs klassen. SQLToolRepository og SQLEmployeeRepository klassene opprettet. Klassene arver strukturen til IToolRepository og IEmployeeRepository. Klassene skal gjøre CRUD funksjoner på database nivå. SQLToolRepository og SQLEmployeeRepository bruker Entity Framework til uthenting og lagring av informasjon til databasen. Se Figur 5-10 for strukturen av disse klassene.

# Web-applikasjonen



Figur 5-10: Codemap SQLklasser

## 6 Database

I dette prosjektet er det opprettet en database for lagring av nødvendig data. Det vil være en felles database for både RFID-applikasjonen og web-applikasjonen.

Websiden kobles opp mot databasen slik at den kan hente ut informasjon og grafisk fremstille det i tabell. Tabellene vil vise en oversikt over hvilket verktøy som er til utlån, hvilke verktøy som er tilgjengelig, hvem som har lånt verktøy og en historikk over tidligere verktøy som har vært til utlån. Websiden bruker databasen for å muliggjøre samhandlinger mellom bruker og applikasjon. Hensikten med websiden vil være at det blir lettere å holde kontroll på verktøyene, men det vil også fungere som en inventarliste for Caverion. I prosjektet brukes Microsoft Azure for websiden og databasen. Dette vil gi en økt sikkerhet da databasen ikke er lokal, men heller sky basert. Ved å bruke Microsoft Azure får websiden DNS (Domain Name System) adresse istedenfor en IP-adresse, som betyr at problemer med IP-adresser som endrer seg ved f.eks. strømbrudd kan unngås.

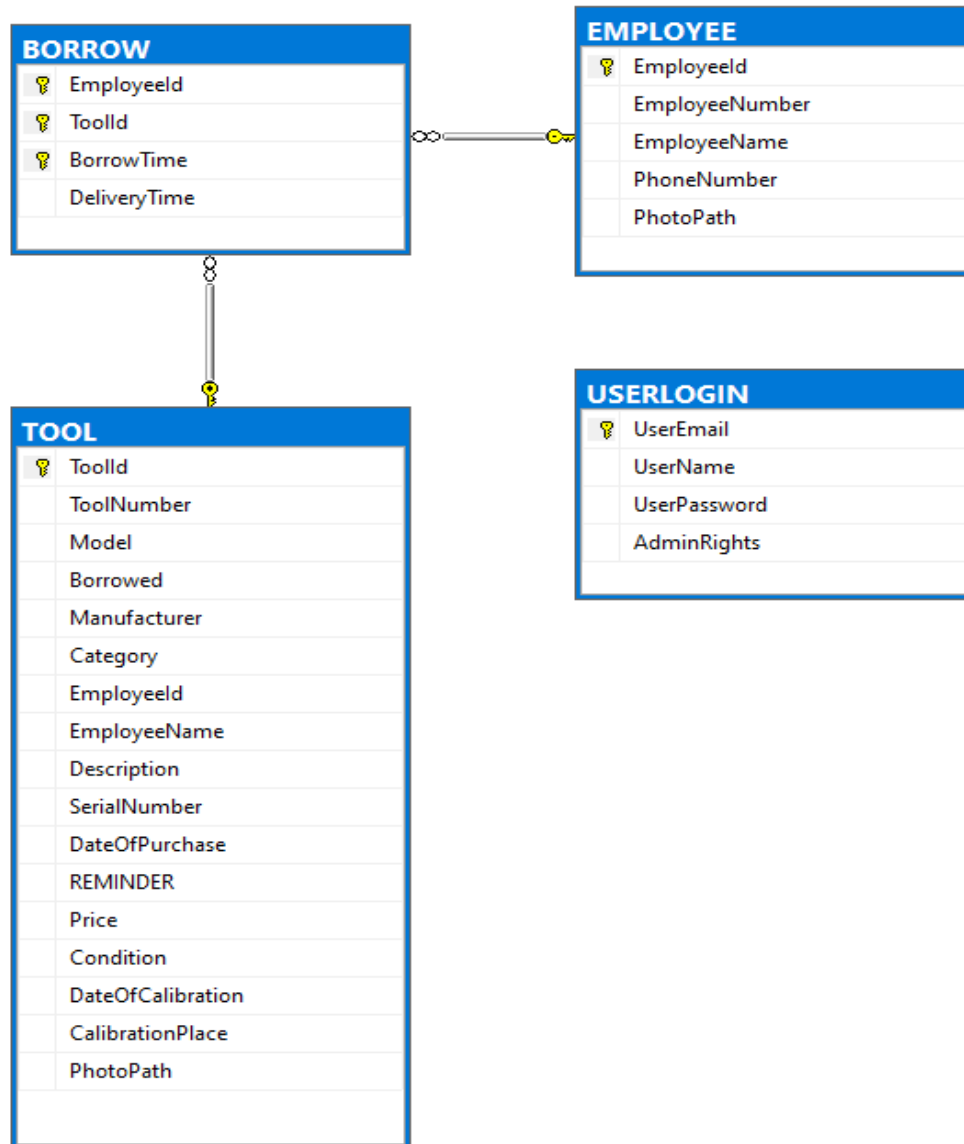
### 6.1 Databasekrav

I dette underkapittelet skal leseren få en oversikt over kravene for databasens oppbygning. Kravene skal gi en forståelse for brukerens behov og gjøre databasen så brukervennlig som mulig. Databasen skal være oversiktlig og inneholde den nødvendige dataen som trengs for at systemet skal fungere som ønsket av Caverion. Databasen skal inneholde data om verktøy, ansatte og utlåns historikk. Denne dataen skal være tilgjengelig å hente ut og oppdateres fra applikasjonene laget i prosjektet. Tabellene skal være intuitive slik av Caverion ved en senere anledning enkelt kan utvide databasen og eksisterende tabeller.

## 6.2 Databasestruktur

Databasen inneholder informasjon om de ansatte, verktøy og utlånshistorikk. Det er fire tabeller og et View som til sammen danner grunnlaget for databasen. Figur 6-1 viser ER-diagrammet (Entity Relations) som definerer forholdet mellom tabellene.

Employee-tabellen inneholder data om de ansatte i Caverion, og alle de ansatte får en egen ansatt-ID basert på data som blir lest fra adgangskortet. Fornavn og etternavn blir lagret i et History-view for å enkelt kunne spore opp den verktøyet og utlåneren. Fra Caverion var det ønskelig at Tool-tabellen skulle inneholde informasjon om produsent, verktøytype, modell og en boolsk variabel som definerer om verktøyet er lånt eller ikke.




Figur 6-1: ER-diagram for databasen

### 6.2.1 Employee-tabell

Denne tabellen inneholder informasjon om de ansatte. For en ansatt i Caverion lagres det navn, etternavn, telefonnummer, ansatt-ID og en photopath som gjør at man kan legge til bilder av de ansatte, men kun ved samtykke. ansatt-ID referer her til «EmployeeId» i Figur 6-2.

Employee-tabellen utgjør en viktig del av systemet da den kobler opp en ansatt mot ett eller flere verktøy, slik at utlånet kan registreres i Borrow-tabellen.




EMPLOYEE	
	EmployeeId
	EmployeeNumber
	EmployeeName
	PhoneNumber
	PhotoPath

Figur 6-2: Viser Employee-tabellen

### 6.2.2 Borrow-tabell

Denne tabellen fungerer som en utlånshistorikk da den kombinerer data fra Employee-tabellen og Tool-tabellen. Når et verktøy blir lånt ut vil RFID-applikasjonen slå sammen ansatt-ID og verktøy-ID med dato og klokkeslett når den ble lånt ut, og dato og klokkeslett når den ble levert, som vist i Figur 6-3. Siden denne tabellen henter data fra Employee og Tool-tabellen, blir det enklere å lage en nøyaktig historikk på hvem og når verktøyet har blitt lånt ut.

BorrowTime blir her brukt som en primærnøkkel for å ha en unik identifikator på transaksjonen annet enn EmployeeId og ToolId.


BORROW	
	EmployeeId
	ToolId
	BorrowTime
	DeliveryTime

Figur 6-3: Viser Borrow-tabellen



### 6.2.3 Tool-tabell


Tool-tabellene inneholder viktig informasjon om verktøyet, og noen eksempler på lagret data er produsent, modell, kalibreringsdato, kategori og verktøy-ID. I denne tabellen er det satt inn en kolonne «Borrowed» av datatypen BIT, som da viser om verktøyet er utlånt eller inne på lageret. Dataen i denne tabellen blir brukt i som direkte informasjon i web-applikasjonen

TOOL	
	ToolId
	ToolNumber
	Model
	Borrowed
	Manufacturer
	Category
	EmployeeId
	EmployeeName
	Description
	SerialNumber
	DateOfPurchase
	REMINDER
	Price
	Condition
	DateOfCalibration
	CalibrationPlace
	PhotoPath

Figur 6-4: Viser Tool-tabellen

### 6.2.4 UserLogin-tabell

UserLogin-tabellen blir brukt for å gi de ulike ansatte tilgang på websiden. En bruker blir opprettet med ansatt e-post, ansattnavn, passord og administrator rettigheter, se Figur 6-5. For å sette administrator rettigheter må AdminRights setter til True eller False, det vil si at brukere med AdminRights = True, vil ha administrator rettigheter til å legge til eller slette ansatte og verktøy.

USERLOGIN	
	UserEmail
	UserName
	UserPassword
	AdminRights

Figur 6-5: UserLogin-tabell med administratorrettigheter

## 7 Kostnader og estimater

Dette kapittelet vil ta for seg kostnadsestimering og budsjettering for prosjektet. Prisene vil basere seg på innhentede priser fra ulike leverandører, der noen vil inkludere rabatter.

Budsjettet er satt av Caverion, og det vil være gruppens ansvar å ikke overstige den gitte summen. Utstyrlisten og innkjøpslisten er basert på utstyret som blir tatt i bruk i dette prosjektet og fungerer som en anbefaling, og det vil være opp til Caverion om de ønsker å følge listen basert på resultatene.

### 7.1 Utstyrliste

Ustyrlisten som er representert av Tabell 7-1 viser en komplett liste over utstyr som har blitt valgt til å prosessere data, registrere verktøy/ansatte og grafisk fremstille data på en storskjerm. Utstyret som er valgt for dette prosjektet, er valgt med tanke på behov og robusthet, igjen valgt etter kravene til Caverion. RFID-brikker blir valgt etter hva slags materiale de skal festes på og blir valgt etter hvor hardføre de er, da de må tåle påkjenning fra ytre omgivelser og hyppig bruk.

Tabell 7-1: Viser en komplett liste over utstyr som er brukt i prosjektet.

#	Utstyr	TAG-Navn	Kommentar
1	RFID Read/Write	RW-0001	RFID sensor for lesing av verktøybrikke
2	RFID Controller	RC-0001	RFID kontroller for kommunikasjon mellom desktop app og RFID-leseren
3	RFID Reader	RR-0001	RFID leser for lesing av ansattes adgangskort
4	HP Mini Desktop	MD-0001	Mini desktop for kjøring av RFID-applikasjon og web-applikasjon
5	Big screen	BS-0001	Storskjerm for grafisk fremstilling av web applikasjonen
6	RFID-brikke metal	RT-0001	RFID-brikke for plassering på metalloverflater
7	RFID-brikke plastikk	RT-0002	RFID-brikke for plassering på plastoverflater
8	PSU 24V	PS-0001	Strømforsyning til kontroller og RFID-lesere
9	Kabel SUBD9		Kobling mellom kontroller og datamaskin (RS-232)
10	Kabel V1-W		Kobling mellom RFID-leser og kontroller
11	Kabel ABG		Strømkabel til kontroller 24V

## 7.2 Kostnad

Prisene i Tabell 7-2 er gitte priser hentet fra ulike leverandører, med ulik grad av prisavslag. Dette er en veiledende prisantydning som tar for seg prisen for hva systemet for dette prosjektet kostet. I Tabell 7-2 er det ikke tatt med de månedlige kostnadene for Azure databasen eller Azure App services, da dette er tjenester som må spesifiseres etter behov og prisen vil variere.

Tabell 7-2: Viser innkjøpsliste med pris

Navn	Beskrivelse	Antall	Pris pr stk	Totalpris
Pepperl + Fuchs RFID-leser	RFID-leser (antenne)	1	6735,85	kr 6 735,85
Pepperl + Fuchs RFID kontroller	Kontroller til RFID-leser	1	6410,25	kr 6 410,25
HP Mini desktop	Liten stasjonær PC	1	8790,00	kr 8 790,00
V1S-G-5M-PUR-ABG-SUBD9	Kabel kontroller til RS-232	1	297	kr 297,00
V1-G-5M-PUR-ABG-V1-W	Kabel RFID-leser til kontroller	1	310	kr 310,00
V1-G-5M-PUR-ABG	Strømkabel til kontroller	1	151,5	kr 151,50
D-sub to USB	Serial adapter til USB	1	320	kr 320,00
<b>Total sum</b>				<b>kr 23 014,60</b>

## 8 Diskusjon

Diskusjonskapitlet tar for seg en kort evaluering av metodene som er blitt brukt og løsningen gruppen har kommet fram til. I dette kapitlet fremkommer også feilkildene som oppsto under systemets utviklingsfase. Det er også lagt til grunn for at applikasjonene kan videreutvikles med ideer gruppen ikke fikk lagt til på grunn av mangel på tid.

### 8.1 Korona situasjonen

Ved starten av prosjektet ble det tilordnet grupperom, ved universitet der prosjektet var planlagt å bli gjennomført.

Det ble fort klart at situasjonen med koronaviruset ville hindre gruppen i å få arbeidet fysisk på universitetets områder. Under utviklingen av prosjektet kom USN ut med en kunngjøring om at universitetet skulle stenges ned. Dette førte til at gruppen ikke kunne jobbe fysisk sammen, som videre hemmet fremgangen til prosjektet. Etter møte med veileder og bedrift fikk gruppen tillatelse til å levere en prototype av systemet til Caverion.

### 8.2 RFID-applikasjon

Målet med RFID-applikasjonen var å lage et system som håndterer utlån og levering av verktøy ved hjelp av en RFID-leser med høy og lav driftsfrekvens og en ultrahøy frekvent RFID-leser, og lagrer dataen i en database. Systemet fungerer som forventet og utfører transaksjonen som tiltenkt, men på grunn av ulike feilkilder kan ikke systemet prestere best mulig. Det ble tidlig i prosjektet diskutert hva slags type sensortechnologi som skulle brukes til å registrere verktøyene. Valget falt på RFID-leser fra Pepperl + Fuchs, da de leverer RFID-lesere som tilfredsstillt kravene fra Caverion. Modellen ble valgt basert på to kriterier, avstanden den klarer å lese og at den har dobbel polarisering. Med dobbel polarisering menes det at RFID-leseren sender ut vertikale og horisontale elektromagnetiske bølger. Dobbelt polarisering gir en økt grad av frihet for lesing når orienteringen på brikkene er uforutsigbar, som også blir forklart i kapittel 3.2.1.

Det ble diskutert internt i gruppen om RFID-applikasjonen skulle være en konsollapplikasjon eller en windows-applikasjon, men på grunn av brukervennlighet ble konsollapplikasjon valgt bort til fordel for en windows-applikasjon. Det ble konkludert med at å lage et brukervennlig grensenitt i konsollapplikasjon ville nesten vært umulig, spesielt med tanke på å vise data grafisk. En prototype av applikasjonen ble så utviklet i windows forms. Prototypen var i stand til å lese fra RFID-brikker, sjekke dataen mot databasen og utføre en transaksjon på utlånet, og prosessen var til en viss grad helautomatisk. Etter installasjonen av prototypen hos Caverion ble det gjort små justeringen i koden slik at applikasjonen kunne tas i bruk uten RFID-leser, med manuell inntastinger. Den manuelle versjonen ble brukt til videreutvikling av applikasjonen og feilsøking i koden. Dette ville vise seg å være til stor hjelp da prototypen stod hos Caverion over lengre tid.

Før testingen av systemet ble RFID-leseren justert ned i styrke slik at leseavstanden skulle reduseres. Pepperl + Fuchs har et eget software som brukes til å konfigurere RFID-brikker og leser. Under testingen av systemt ble gruppen gjort oppmerksom på at RFID-lesere allikavel klarte å lese brikkene som lå flere meter unna. Dette skyldes refleksjonene fra en verktøyskuff som står rett foran leseren, se Figur 8-1. De elektromagnetiske bølgene reflekteres fra verktøyskuffen og videre mot det perforerte panelet, og fører til lesinger av brikker som ikke er i umiddelbar nærhet.



Figur 8-1: Verktøylager med illustrasjon av elektromagnetisk refleksjon fra verktøyskuff til perforert panel.

### 8.3 Web-applikasjon

Det opprinnelige målet for web-applikasjonen var å vise om et verktøy var ute eller inne, når det ble tatt ut, personen som låner, bilde, filtreringsmuligheter, søkefunksjon og administrator-funksjonalitet for endring, legge til verktøy og slette verktøy.

Under utviklingen av web-applikasjonen ble det utviklet en sluttbrukerapplikasjon og en administratorapplikasjon. For å skille mellom hvem som skulle ha muligheten til å redigere et verktøy. Opprinnelig så ble visningen av om verktøyet var ute/inne lagt til som ikoner, hake for inne og et kryss for at verktøyet var på utlån. Caverion ønsket å fjerne dette og heller legge dette inn som en funksjon hvor en ser personen som låner. Slik dette vil fungere er at hvor det ikke står en utlåner betyr det at verktøyet er på lageret. Hvor det står en utlåner betyr det at verktøyet er på utlån. En annen funksjon som ikke ble iverksatt var når verktøyet blir tatt ut. Denne funksjonen ligger i RFID-applikasjonen allerede og var ikke ønskelig å ha på web-applikasjon av Caverion.

Originalt var en liste over ansatte en funksjon som ikke var planlagt skulle være inkludert. I et møte med Caverion ble det diskutert om dette var en god funksjon å ha med i web-applikasjonen. Her ble det konkludert med at dette var et godt tilskudd til web-applikasjonen for å lett finne kontaktinformasjonen til den enkelte ansatte som låner et verktøy en selv ønsker å låne.

Innloggingsfunksjonen var i utgangspunktet et av kravene til web-applikasjonen, men ble ikke prioritert før gruppen hadde fått på plass grunnleggende funksjonalitet. Når utviklingen av innloggingsfunksjonen ble iverksatt, ble det mulighet for å samkjøre sluttbrukerapplikasjonen og administratorapplikasjonen ved hjelp av session. Det session gir muligheten til er at den som logger inn på applikasjonen har en rolle. Enten sluttbruker eller administrator. Dette gjorde det mulig å begrense utgiftene ved bruk av Azure. Innloggingsfunksjonen ble ikke diskutert før i slutten av prosjektet som gjorde at den ikke er optimalisert for endring av passord. Passordet blir satt av brukeren første gang brukeren logger inn på web-applikasjonen. Brukeren vil da allerede ligge i databasen med epostadresse og en rolle som definerer administratortilgang. Videre var det tiltenkt at det skulle opprettes en funksjon for å legge til nye brukere, men ble ikke påbegynt da tiden ikke strakk til. Funksjonaliteten ligger der slik at det er mulig for videre utvikling. Et problem som oppsto da sluttbrukerapplikasjonen og administratorapplikasjonen ble slått sammen var at filtreringen av tabellen ikke fungerte lenger. For filtreringen av tabellen så blir CSS og JavaScript brukt som blir referert til i programmet. Gruppen klarte ikke å finne løsning på dette problemet.

## 8.4 Videre utvikling

### 8.4.1 Desktop

Under utviklingen av RFID-applikasjonen var det et stort fokus på maskinvareuavhengighet. Det var tiltenkt at det skulle være enkelt å endre på maskinvaren uten å måtte endre for mye på koden. I koden vil relevante metoder og klasser være navngitte med navn som er selvforklarende, med parametere og variabler som står i stil til metodenavn. Det vil være nødvendig å skjerme RFID-brikker som ligger i umiddelbar nærhet av RFID-leseren for å unngå uønsket lesing.

### 8.4.2 Web-applikasjon

Da gruppen skulle utvikle databasen så var det uenigheter om hva som skulle være primærnøkkel. Det falt til slutt på å bruke RFID nummeret som primærnøkkel. Her oppstår det et problem:

Web-applikasjon bruker Entity Framework til å utføre sine funksjoner. Ved redigering av et verktøy eller ansatt så blir primærnøkkel brukt til å finne det nøyaktige verktøyet eller ansatt som skal redigeres. Ved endring av primærnøkkel vet ikke lenger Entity Framework hvilket verktøy eller ansatt som skal redigeres. For å endre RFID nummeret til et verktøy eller ansatt så må dette gjøres i Azure. Her kan løsningen være å endre primærnøkkel til verktøynummer som i praksis ikke skal endres eller bruke SQL-Query.

Innloggings funksjon ble utviklet sent i prosjektet. Dette førte til at gruppen ikke fikk implementert en funksjon for å legge til brukere. Hvis nåværende løsning skulle vise seg å by på problemer, må det vurderes å legge til en funksjon som gjør det mulig å legge til nye brukere.

## 9 Oppsummering

Målet med prosjektet var å planlegge og utvikle et utlånssystem for Caverion, som minimerer den manuelle prosessen av utlån eller innlevering av verktøy. Tanken bak utlånssystemet var at det skulle bli enklere for de ansatte å utføre utlånsprosessen, men også gi verktøyansvarlig en bedre oversikt over verktøyene som er inne eller til utlån. Systemet er delt opp i to applikasjoner, hvor RFID-applikasjonen håndterer alt av registrering og web-applikasjonen håndter visning og redigering.

RFID-applikasjonen ble utviklet i Visual Studio grunnet gruppens kjennskap til utviklingsmiljøet, men også fordi Caverion ønsket et system som var minst mulig lisensbasert. Windows Forms ble valgt fremfor konsoll-applikasjon. Det ble bestemt av gruppen ettersom at det ville være enklere å designe en brukervennlig applikasjon i Windows Forms. Valg av RFID som sensorteknologi baserer seg på kravet om at systemet skal være automatisk. For å oppnå kravet, valgte gruppen en UHF RFID-leser fra Pepperl + Fuchs. Med en UHF RFID-leser kan RFID-brikkene detekteres over større avstander, enn ved bruk en LF RFID-leser. Dette fører til at de ansatte ikke trenger å vise brikkene til leseren. Brikken detekteres når en ansatt passerer leseområdet med et merket verktøy. I dette prosjektet har Pepperl + Fuchs stilt RFID-leser og IDENTController til disposisjon for gruppen gjennom hele prosjektperioden.

Web-applikasjonen ble også utviklet i Visual Studio. Web-applikasjonen ble laget med Razor Pages som var et ønske fra veileder. Web-applikasjonen er ment til å være en applikasjon hvor brukeren kan se hvilke verktøy som var utlånt eller var tilgjengelig på lageret, redigere, opprette verktøy og ansatte, og få kontakt informasjon til utlåner av verktøy. Web-applikasjonen møter disse kravene og har en rollefunksjonalitet for å skille mellom administrator og sluttbruker. Sluttbrukeren har mulighet til å få informasjon om verktøy og utlåner av verktøyet. Administrator rollen er ment å bli brukt av verktøyansvarlig som skal redigere, opprette verktøy og ansatte. Kodestrukturen til web-applikasjonen er delt i forskjellige klassebibliotek for å lett gjøre det mulig å bli kjent med applikasjonen. Dette gir muligheten for fremtidig utvikling av programmet.

Noen av forbedringene som kan gjøres ved videre utvikling av systemet er å endre primærnøkkel til å være verktøynummer og ansattnummer. Bygge ut funksjonaliteten for endring av passord for innlogging.

Selv om gruppen og Caverion hadde mange ideer for funksjonalitet i applikasjonene, måtte gruppen begrense hvilke funksjoner som skulle prioriteres. Kravene i prosjektbeskrivelsen hadde høyere prioritet, og siden tiden ikke strakk til fikk ikke gruppen implementert andre ønskede funksjoner som ikke var i oppgavebeskrivelsen.

# Referanser

- [1] E. Rossen, «C#», I store norske leksikon, 2021. [Online]. Hentet fra: <https://snl.no/C%23>. (Lastet ned: 29.04.2021)
- [2] smart-TEC, RFID-Technology, 2021. Hentet fra: <https://www.smart-tec.com/en/auto-id-world/rfid-technology>. (Lastet ned: 11.01.2021).
- [3] iFour Technolab, «ASP.NET VS ASP.NET CORE», ifourtechnolab.com. Hentet fra: <https://www.ifourtechnolab.com/blog/differences-between-asp-net-and-asp-net-core-asp-net-vs-asp-net-core>. (Lastet ned: 11.02.2021)
- [4] kudvenkat. «ASP.NET core razor pages tutorial», Youtube.com [Spilleliste]. Hentet fra: [https://www.youtube.com/playlist?list=PL6n9fhu94yhX6J31qad0wSO1N\\_rgGbOPV](https://www.youtube.com/playlist?list=PL6n9fhu94yhX6J31qad0wSO1N_rgGbOPV) (Lastet ned:02.03.2021)
- [5] HP Halvorsen. «ASP.NET Web Programming», halvorsen.blogg, Hentet fra:[https://halvorsen.blog/documents/programming/web/asp\\_net.php](https://halvorsen.blog/documents/programming/web/asp_net.php) (Lastet ned: 11.02.2021).
- [6] Forfatter, «SQL», Wikipedia, 2021 [Online]. Henter fra: <https://en.wikipedia.org/wiki/SQL> (Lastet ned: 11.02.2021).
- [7] Microsoft, «Hva er Azure», 2021, Hentet fra: <https://azure.microsoft.com/nb-no/overview/what-is-azure/>. (Lastet ned: 22.02.2021).
- [8] Entity Framework Tutorial, «What is code first», entityframeworktutorial.net. Hentet fra: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>. (Lastet ned: 11.02.2021).
- [9] Pepperl + Fuchs, IUH-F190-V1-FR1-01, 2021. Hentet fra: [https://files.pepperl-fuchs.com/webcat/navi/productInfo/doct/tdoct2945r\\_eng.pdf?v=20201008094114](https://files.pepperl-fuchs.com/webcat/navi/productInfo/doct/tdoct2945r_eng.pdf?v=20201008094114). (Lastet ned: 28.01.2019).
- [10] Pepperl + Fuchs, IC-KP2-2HRX-2V1, 2021. Hentet fra: [https://files.pepperl-fuchs.com/webcat/navi/productInfo/doct/tdoct1710e\\_eng.pdf?v=20200320004915](https://files.pepperl-fuchs.com/webcat/navi/productInfo/doct/tdoct1710e_eng.pdf?v=20200320004915). (Lastet ned: 28.01.2021).
- [11] rFIDEAS, Wave ID Plus, 2021. Hentet fra: [https://www.rfideas.com/sites/default/files/2020-05/WAVE-ID\\_Plus\\_PA-104F.pdf](https://www.rfideas.com/sites/default/files/2020-05/WAVE-ID_Plus_PA-104F.pdf). (Lastet ned: 28.01.2021).
- [12] Hewlett-Packard, HP ProDesk 400 Series, Hentet fra: <https://www8.hp.com/us/en/desktops/business/prodesk-400.html>, (Lastet ned: 22.03.2021).



# Vedlegg

Vedlegg A Prosjektbeskrivelse

## Bachelor Thesis

**Title: Automatisk registrering av uttak/innlevering av verktøy**

**USN supervisor:** Hans-Petter Halvorsen

**External partner:** Caverion

### **Task background:**

Caverion har en del kostbart verktøy og har tidligere prøvd å implementere ulike system for å registrere uttak/innlevering av dette. Disse systemene har i ulik grad vært en manuell prosess, som sklir ut over tid. Det er ønskelig at dette blir en helt automatisk prosess.

### **Task description:**

Oppgaven består av flere mindre arbeidsoppgaver som til sammen danner grunnlaget for en totalløsning. Løsningsforslagene er kun veiledende og vil bli bestemt nærmere i møte mellom bedrift, studenter og USN veileder.

**Database:** Opprette en database for verktøy. Noen eksempler på felter som er aktuelle:

- Verktøynummer
- Produsent
- Modell
- Innkjøpsdato
- Kalibreringsdato
- Pris
- Eier
- Kategorier
- Underkategorier

**System:** Lage et system som automatisk registrer tidspunkt og person i databasen når verktøy blir tatt ut/(inn) av lageret. En løsning kan være:

- Kortleser ved dør inn til verktøylager, som gjør at verktøyet som blir tatt ut blir registrert på denne personen.
- Alt verktøy blir merket med et RFID-brikke
- RFID antenne på innsiden og utsiden av dør, for å registrere om verktøy blir tatt ut eller levert inn
- Her må tilgjengelige teknologier, enheter, mm vurderes.

**Program:** Lage et program for å enkelt kunne legge til nytt verktøy eller endre på eksisterende i databasen, helst med et webgrensesnitt for enklere tilgang.

Lage et program for grafisk visning på en skjerm:

- Om verktøyet er ute/inne

- Når det ble tatt ut
- Person
- Evt. bilde av verktøy
- Her er det muligens ønskelig med en touchskjerm for å kunne bla i listen, kanskje med en liten søkefunksjon eller noen filtreringsmuligheter, men uten tilgang til å gjøre endringer.

Det bør være mulighet for å få opp lister over tilgjengelig verktøy, evt hvem som har lånt det og når, osv. Det bør være mulighet for å sende purringer for å få tilbake verktøy, m.m.

Systemet og programmene bør lages på en slik måte at de blir mest mulig generelle og konfigurerbare.

Systemet bør tilfredsstillende GDPR og tilfredsstillende grunnleggende sikkerhetskrav, innlogging, m.m.

Caverion ønsker i hovedsak en totalløsning som er minst mulig lisensbasert. Det må gjøres vurderinger av utviklingsverktøy, programmeringsspråk, rammeverk, m.m. slik at dette blir ivaretatt på best mulig måte.

### **Student category:**

IA

### **Practical arrangements:**

**Caverion må stille med Sensor ifm. karaktersetting av Bacheloroppgaven. Denne personen bør ikke være direkte involvert i veiledningen av studentene.**

Rapporten skal være åpen og tilgjengelig for allmennheten. Dette er et grunnleggende krav fra USN.

Studentene jobber i hovedsak på USN. Caverion sine lokaler er inne på Herøya Industripark sitt område, som betyr at studentene ikke har tilgang til lokalet uten følge. Hvis det kommer frem at det er et større behov for å være på Caverion sine lokaler for å utføre oppdraget, vil studentene måtte ta Herøya Industripark sitt adgangskontrollkurs. Dette er ett tredelt kurs, hvor de to første modulene er nettbasert, mens siste modulen er et klasseromskurs. (Totalt ca. 1 dag)

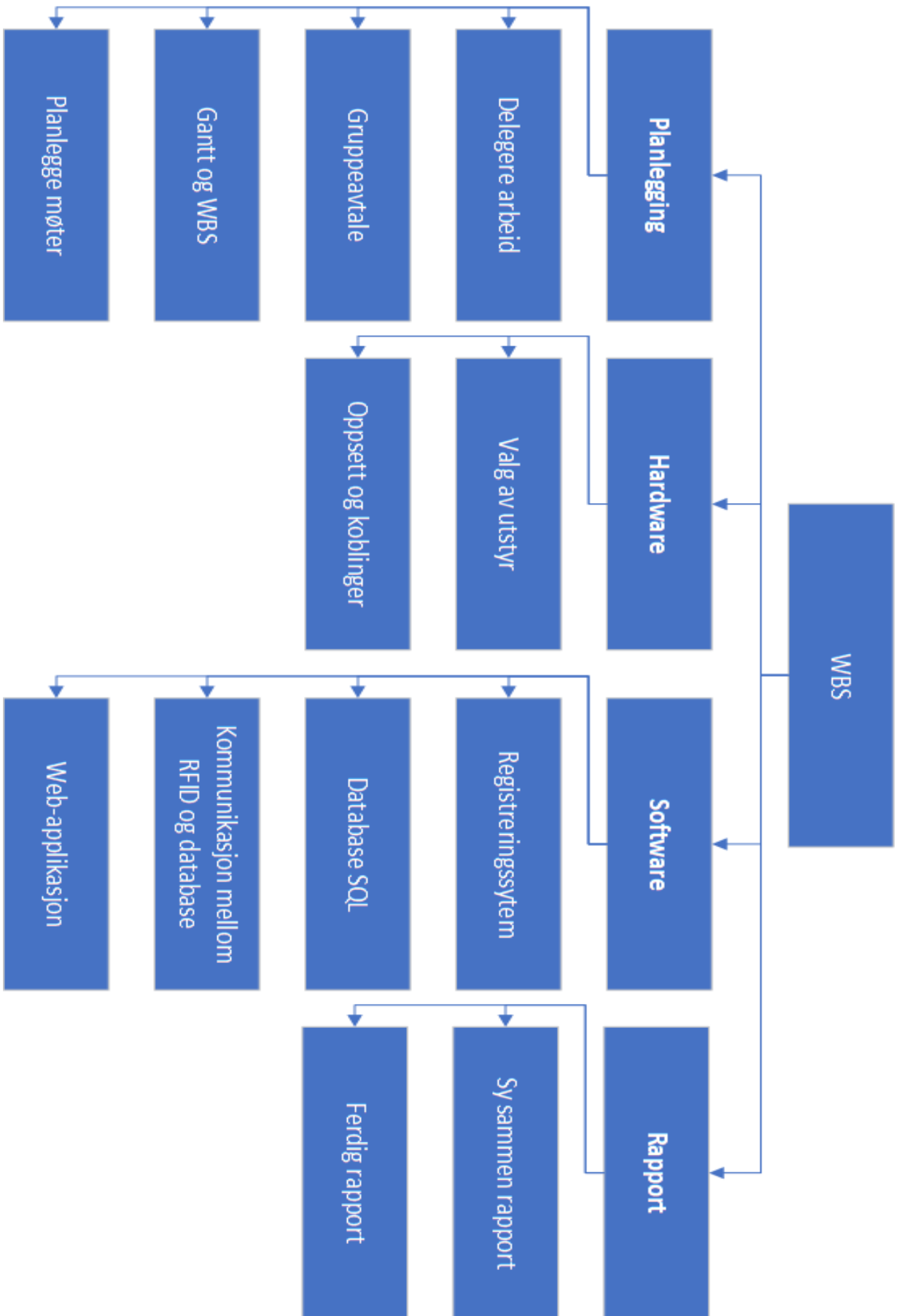
- Et kostnadsoverslag må gjøres før eventuelle innkjøp.
- Caverion sin kontaktperson vil være ansvarlig for innkjøp.

### **Signatures:**

Supervisor (date and signature):

Students (date and signature):

Vedlegg B – WBS



Vedlegg C – Fremdriftsplan

ID	Task Mode	Task Name	Duration	Start	Finish	Pedecessors	Resource Names
1		<b>Dokumenter</b>	<b>33 days</b>	<b>Thu 14.01.21</b>	<b>Mon 01.03.2</b>		
2		Gruppeavtale	2 days	Thu 14.01.21	Fri 15.01.21		
6		Planskisser	18 days	Thu 14.01.21	Mon 08.02.2:		
5		ER diagram	6 days	Thu 21.01.21	Thu 28.01.21		
3		Gantt diagram	1 day	Mon 08.02.2:	Mon 08.02.2:		
4		WBS	1 day	Mon 08.02.2:	Mon 08.02.2:		
7		Få godkjent planskisse	0 days	Thu 11.02.21	Thu 11.02.21		
9		Budsjettforslag	4 days	Mon 15.02.2:	Thu 18.02.21		
10		Ustyrsliste	2 days	Wed 17.02.2:	Thu 18.02.21		
8		Klassediagram	2 days	Fri 26.02.21	Mon 01.03.2:11		
11		<b>Software og programmering</b>	<b>78 days</b>	<b>Thu 21.01.21</b>	<b>Mon 10.05.2</b>		
12		SQL database R1	3 days	Thu 21.01.21	Mon 25.01.2:		
13		Demo program	11 days	Thu 28.01.21	Thu 11.02.21		
14		SQL database R2	2 days	Thu 11.02.21	Fri 12.02.21		
15		.Net CRUD	6 days	Thu 11.02.21	Thu 18.02.21		
16		Utlåns prototype med RFID R1	6 days	Thu 11.02.21	Thu 18.02.21		
17		SQL database R3	1 day	Sun 14.02.21	Sun 14.02.21		
28		Webisode for oversikt (utlån)	6 days	Thu 18.02.21	Thu 25.02.21		
18		SQL database R4	1 day	Thu 25.02.21	Thu 25.02.21		
19		Utlåns prototype med RFID R2	15 days	Fri 26.02.21	Thu 18.03.21		
20		SQL database R5	1 day	Thu 04.03.21	Thu 04.03.21		
25		Prototype web-applikasjon	5 days	Sun 04.04.21	Thu 08.04.21		

Task	Inactive Summary	External Tasks
Task		
Split		
Milestone Summary		
Project Summary		
Inactive Task		
Inactive Milestone		

Project: Fremdriftsplan  
Date: Tue 25.05.21

Page 1

Vedlegg

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	W	T	F	S	S	18 Jan '21	M	T	W	T
21		SQL database R6	1 day	Thu 08.04.21	Thu 08.04.21												
26		Sette opp prototype hos Caverion	1 day	Thu 08.04.21	Thu 08.04.21												
27		Testing av prototype hos Caverion	8 days	Thu 08.04.21	Mon 19.04.21												
22		SQL database R7	1 day	Tue 20.04.21	Tue 20.04.21												
23		SQL database R8	1 day	Thu 06.05.21	Thu 06.05.21												
24		SQL database R9	1 day	Mon 10.05.21	Mon 10.05.21												
29		Inloggingsfunksjon web-applikasjon	1 day	Mon 10.05.21	Mon 10.05.21												
41		<b>Rapport relarert</b>	<b>88 days</b>	<b>Thu 21.01.21</b>	<b>Tue 25.05.21</b>												
49		Utkast Kostnad og Estimater	3 days	Thu 21.01.21	Sat 23.01.21												
46		Utkast Materialer og metoder	4 days	Tue 09.02.21	Fri 12.02.21												
43		Inledning og forside	3 days	Wed 10.02.21	Fri 12.02.21												
44		Utkast System og maskinvare	3 days	Wed 17.03.21	Fri 19.03.21												
45		Utkast Database	3 days	Mon 22.03.21	Wed 24.03.21												
47		Utkast RFID-applikasjon	3 days	Mon 05.04.21	Wed 07.04.21												
48		Utkast Web-applikasjon	3 days	Mon 05.04.21	Wed 07.04.21												
51		Fastslå endelig porsjekt tittel	1 day	Fri 16.04.21	Fri 16.04.21												
50		Utkast Diskusjon	3 days	Mon 03.05.21	Wed 05.05.21												
52		Sensur av rapport	6 days	Thu 13.05.21	Thu 20.05.21												
42		Utkast Oppsummering	3 days	Mon 17.05.21	Wed 19.05.21												
53		Innlevering av rapport	0 days	Tue 25.05.21	Tue 25.05.21												

Task	Inactive Summary	External Tasks
Task		
Split		
Milestone		
Summary		
Project Summary		
Inactive Task		
Inactive Milestone		
Manual Task		
Duration-only		
Manual Summary Rollup		
Manual Summary		
Start-only		
Finish-only		
External Milestone		
Deadline		
Progress		
Manual Progress		

Vedlegg

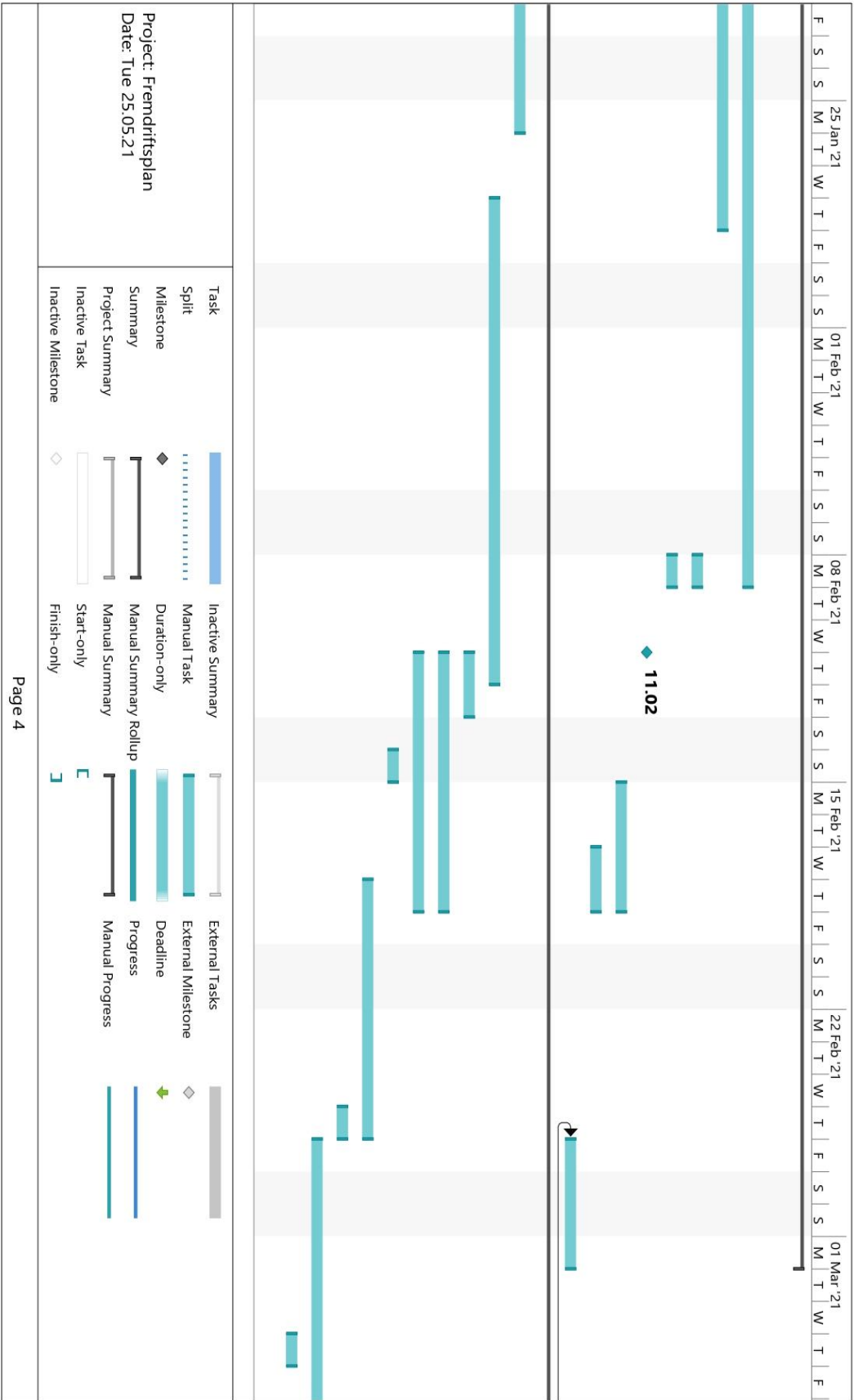
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	W	T	F	S	S	M	T	W	T
30		<b>Arbeidskrav</b>	<b>85 days</b>	<b>Thu 28.01.21</b>	<b>Thu 27.05.21</b>											
31		Vedta prosjektbetskrivelse	13 days	Thu 28.01.21	Mon 15.02.21											
32		Innlevering Arbeidskrav	0 days	Sun 28.02.21	Sun 28.02.21											
34		Filming av Expo video	1 day	Fri 21.05.21	Fri 21.05.21											
33		Innlevering av video til EXPO	0 days	Tue 25.05.21	Tue 25.05.21											
35		USN Expo	0 days	Thu 27.05.21	Thu 27.05.21											
36		<b>Møter</b>	<b>65 days</b>	<b>Thu 18.02.21</b>	<b>Thu 20.05.21</b>											
37		Møte nr.1	0 days	Thu 18.02.21	Thu 18.02.21											
38		Møte nr.2	0 days	Thu 25.03.21	Thu 25.03.21											
39		Møte nr.3	0 days	Thu 29.04.21	Thu 29.04.21											
40		Møte nr.4	0 days	Thu 20.05.21	Thu 20.05.21											

Project: Fremdriftsplan  
Date: Tue 25.05.21

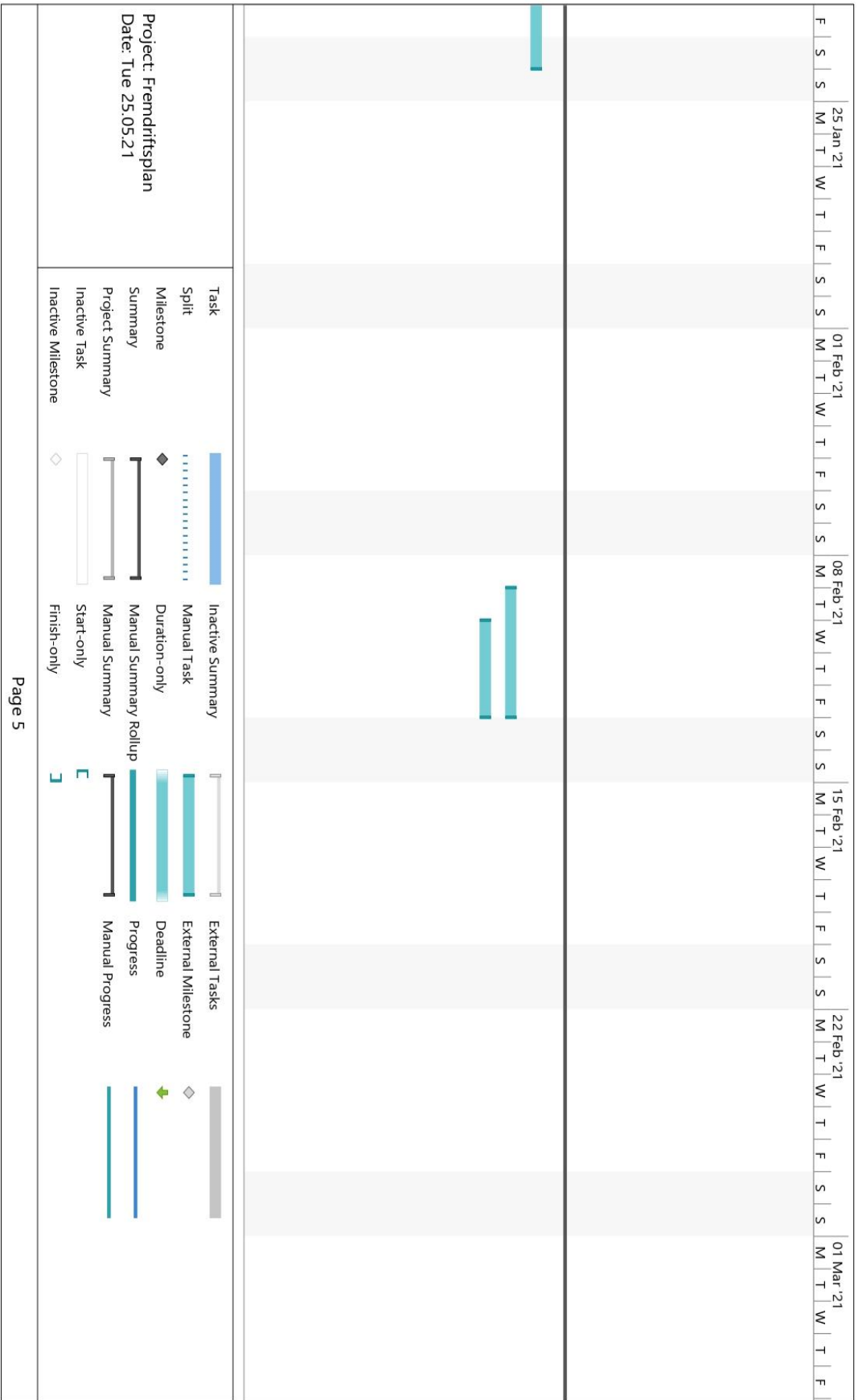
Task	Inactive Summary	External Tasks
Split		
Milestone		
Summary		
Project Summary		
Inactive Task		
Inactive Milestone		

Page 3

Vedlegg



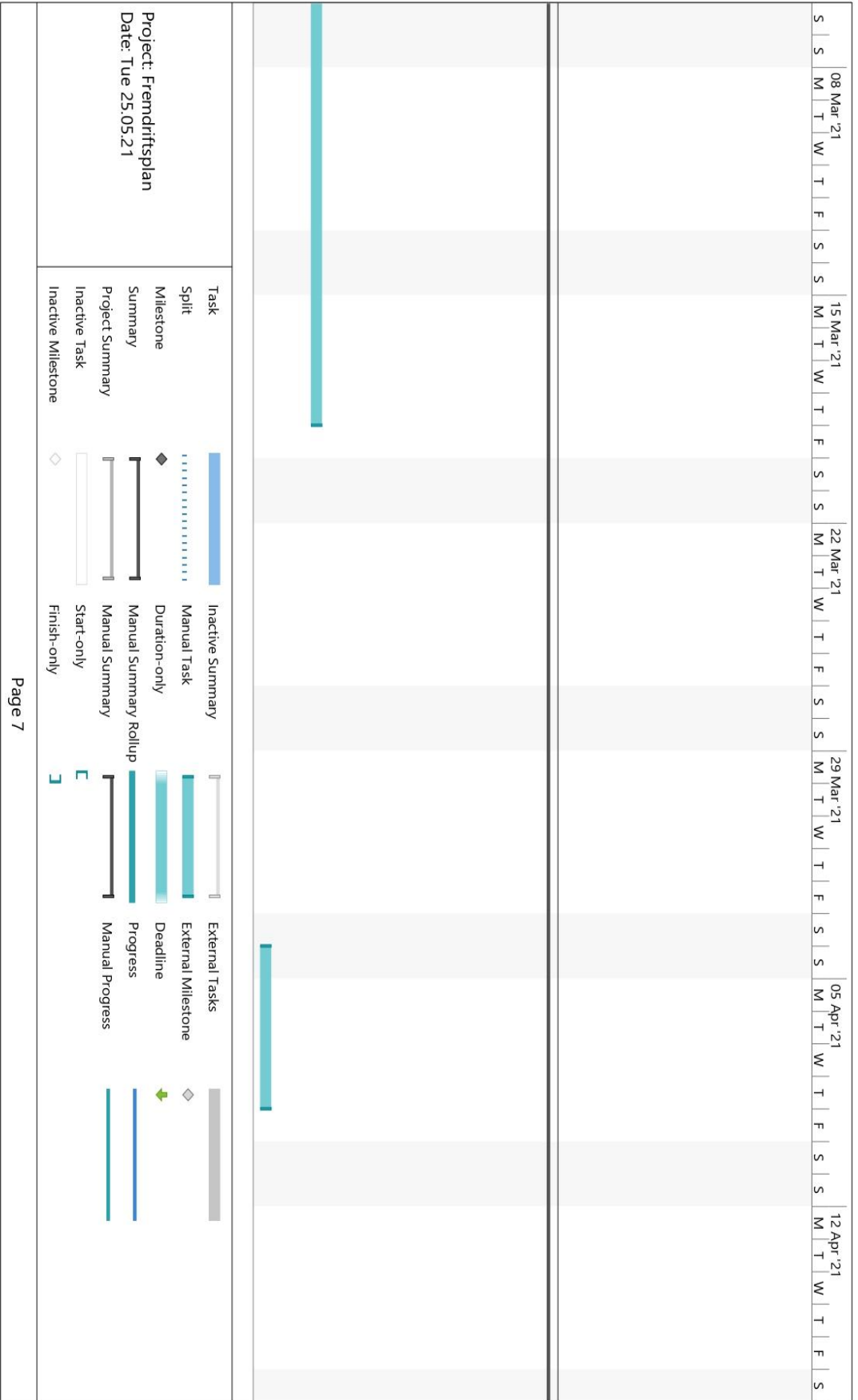
Vedlegg



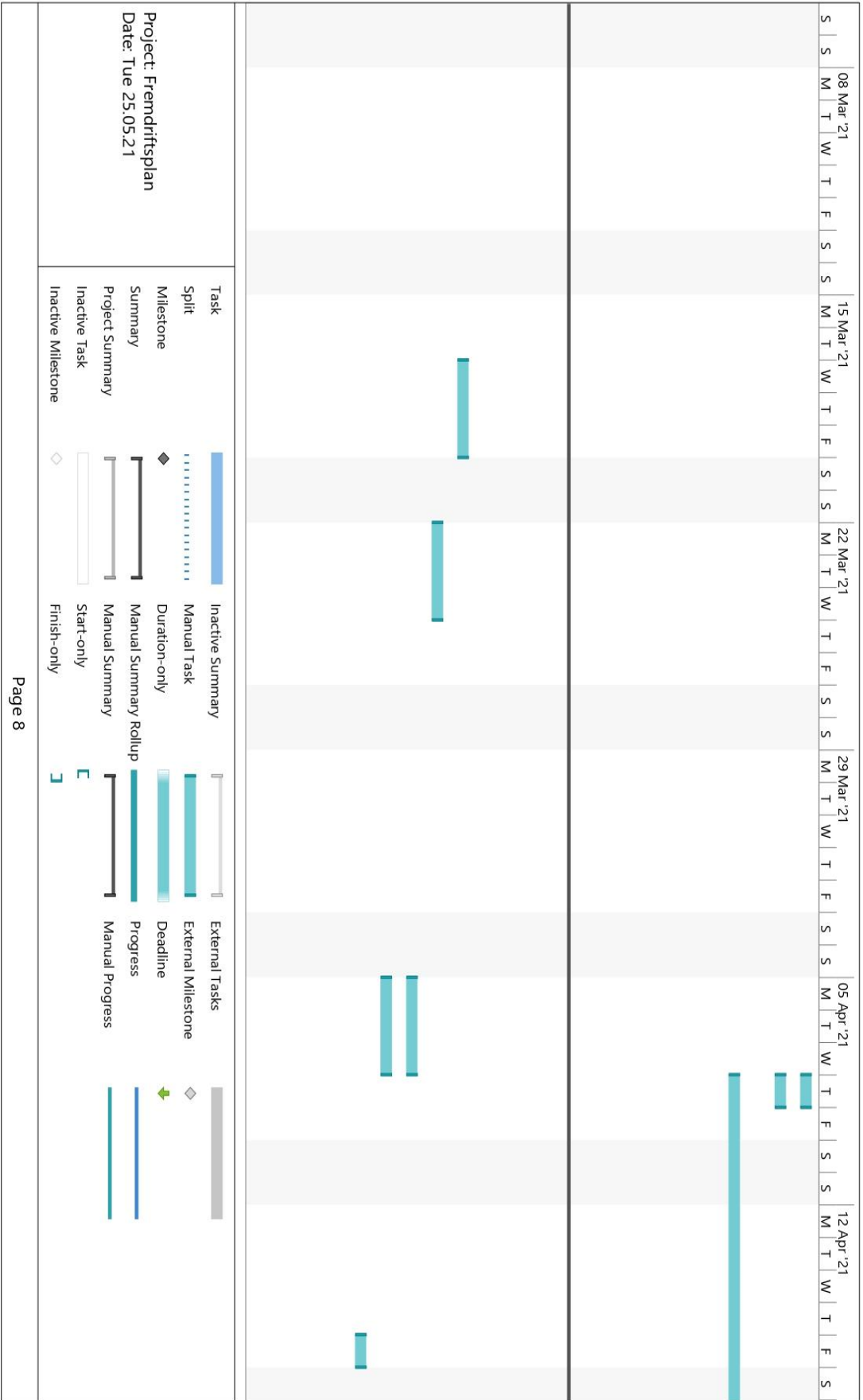




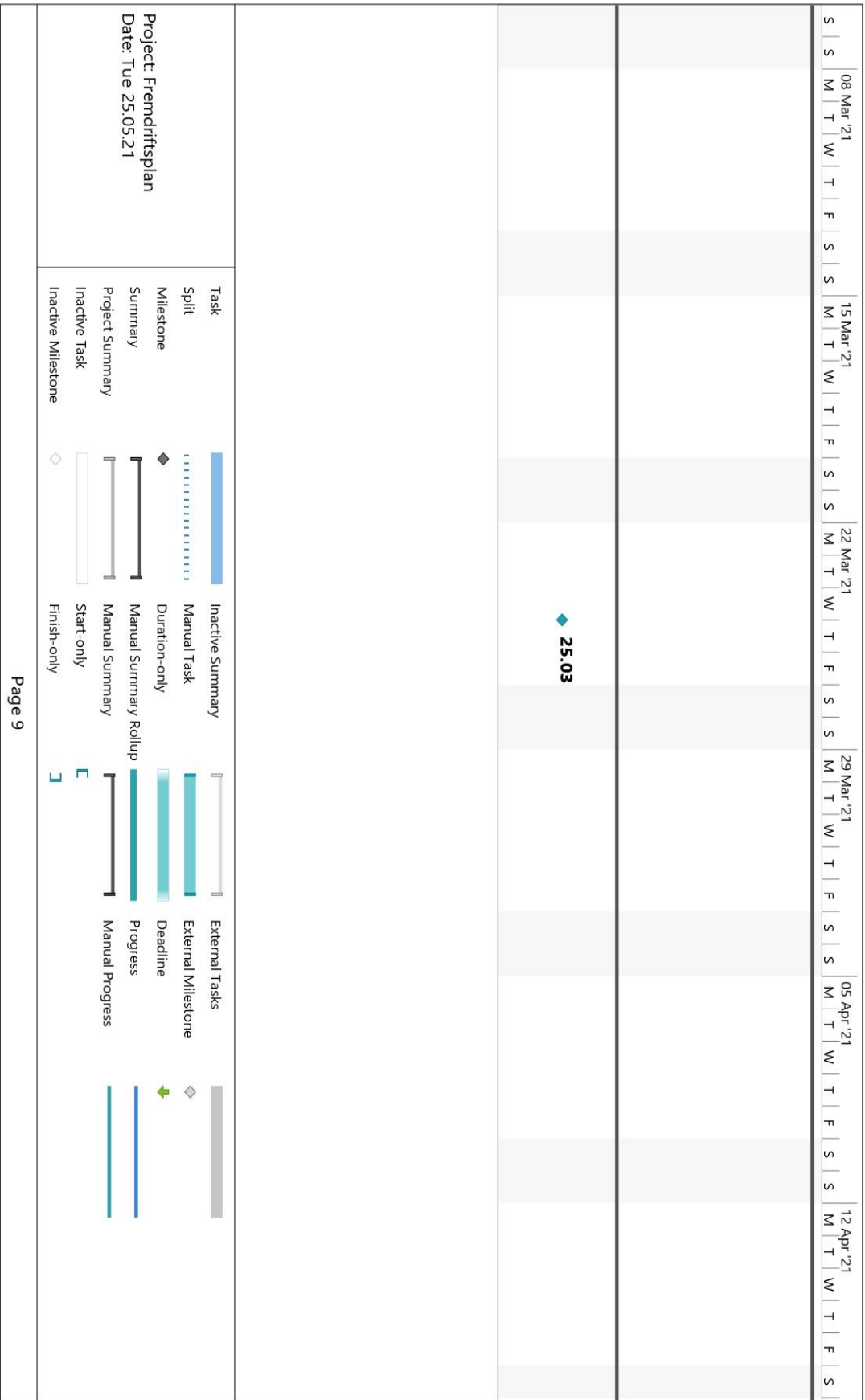
Vedlegg



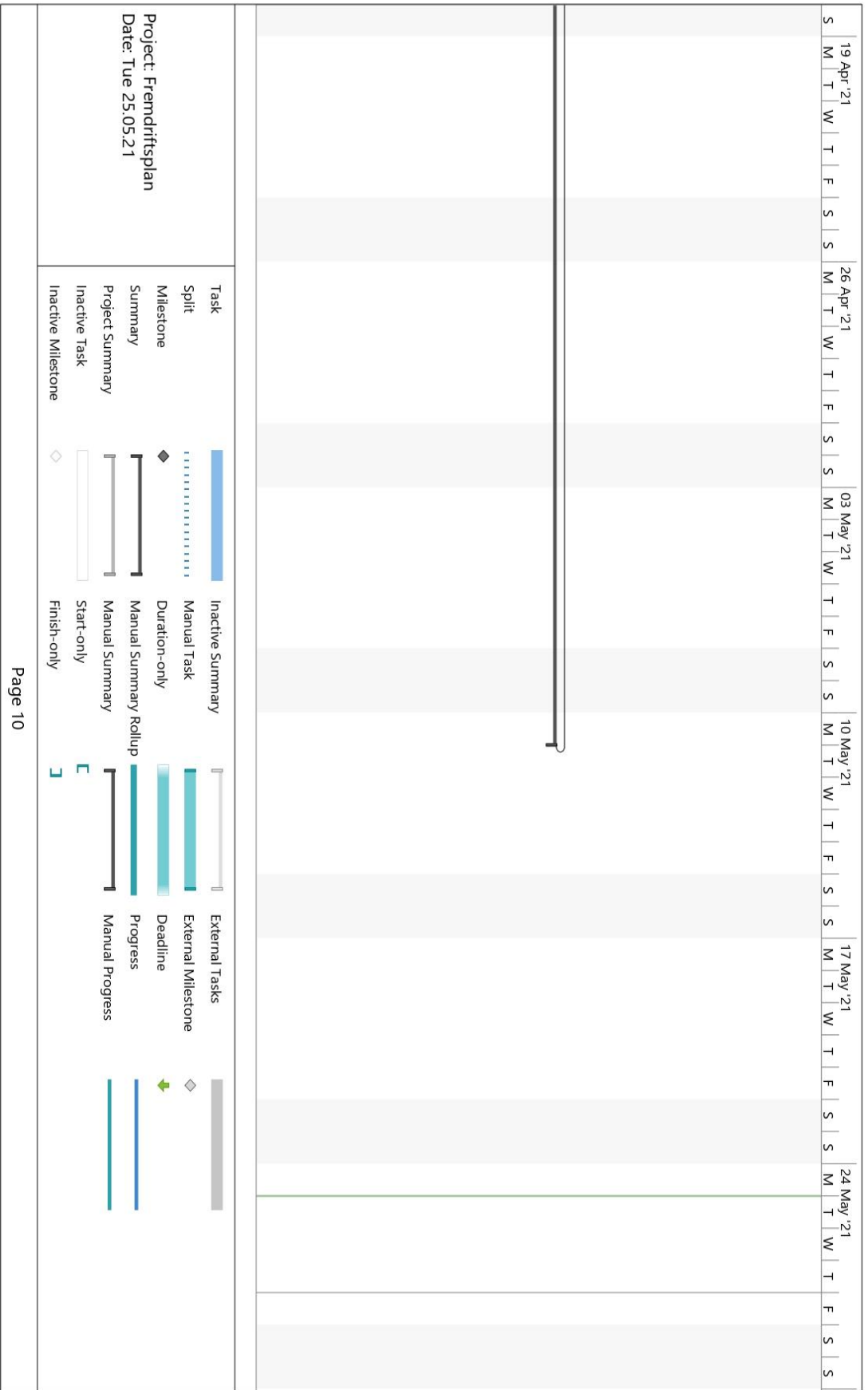
Vedlegg



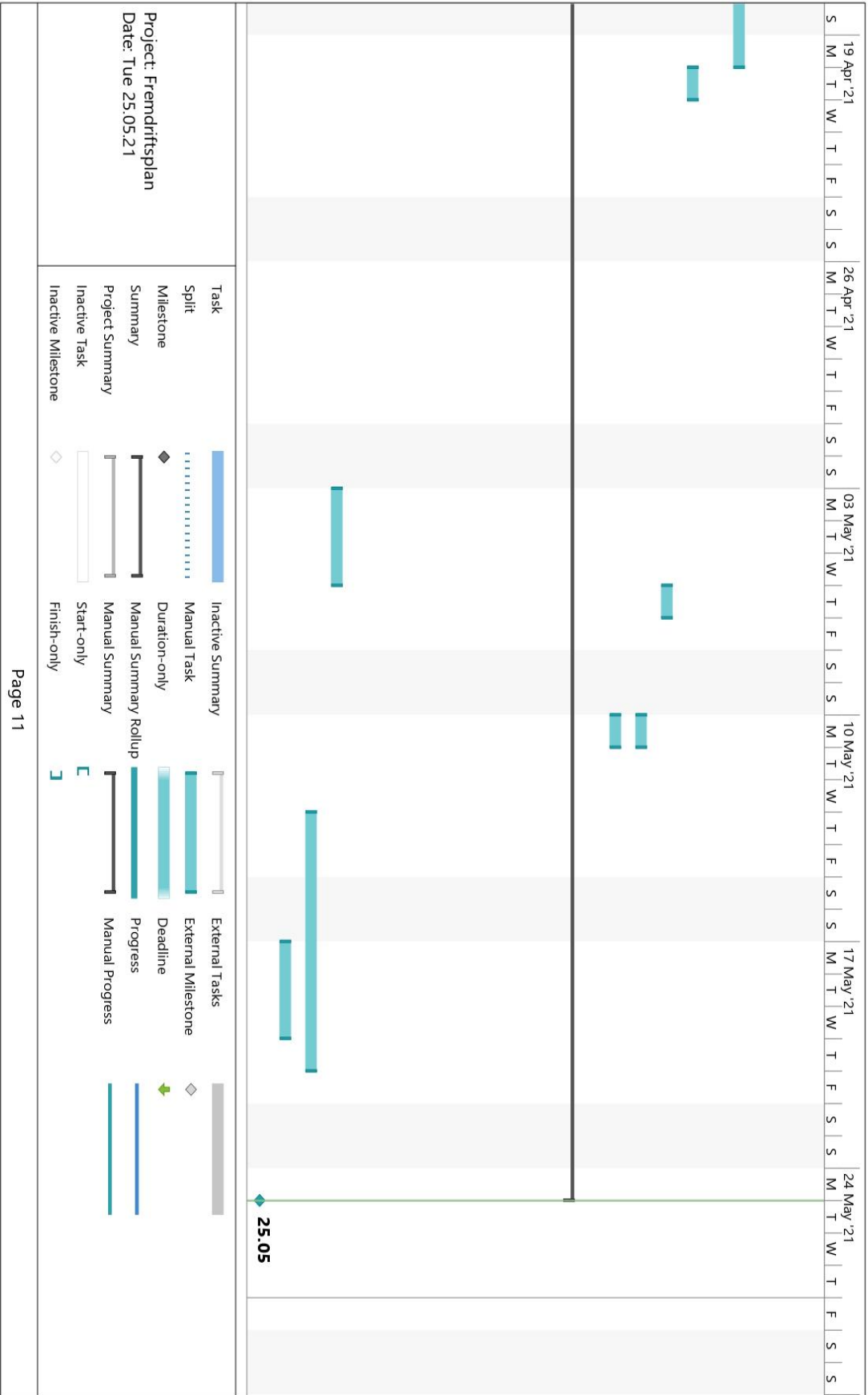
Vedlegg

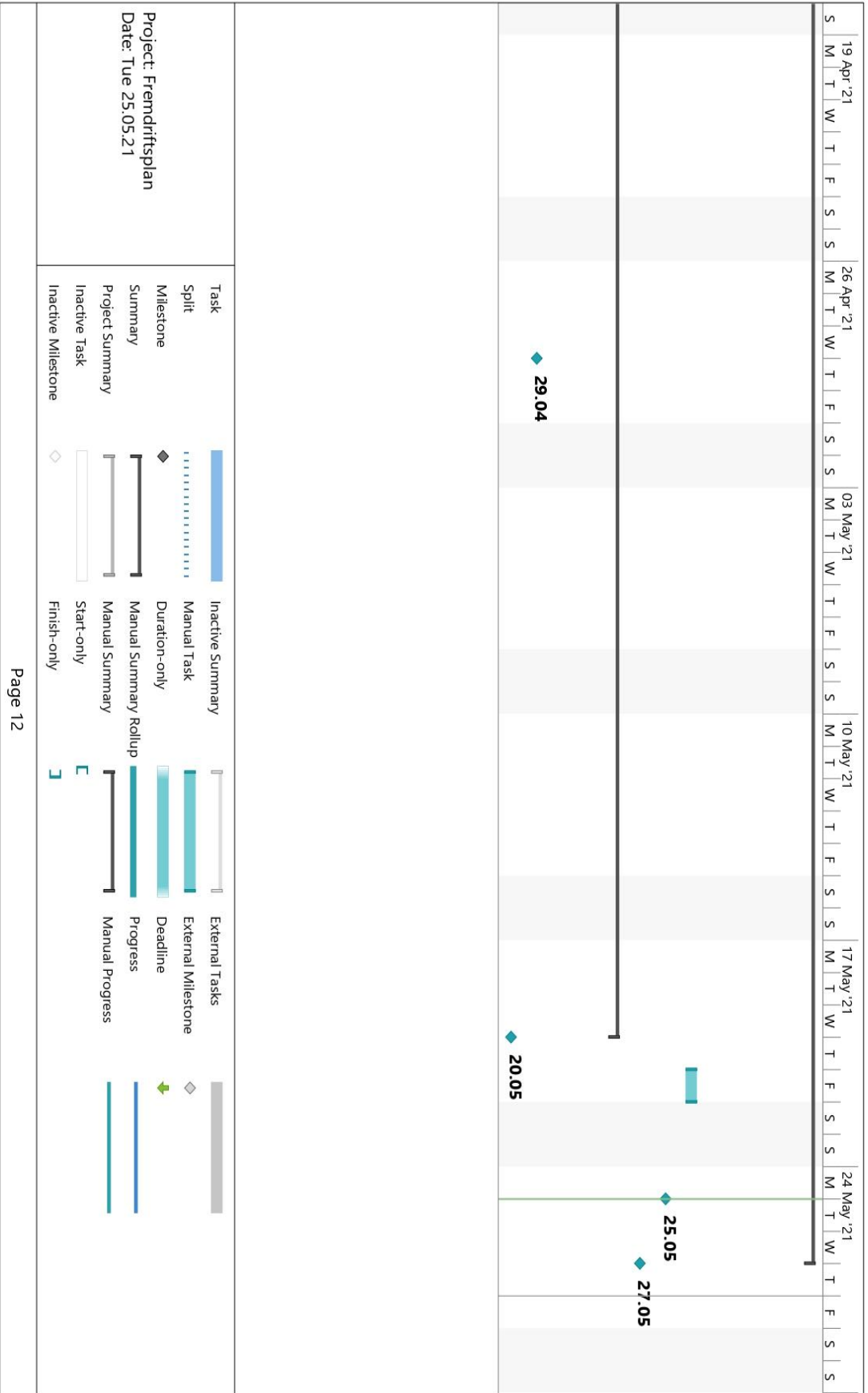


Vedlegg



Vedlegg





Vedlegg D – Gruppeavtale

## Gruppeavtale for IA6-6-21

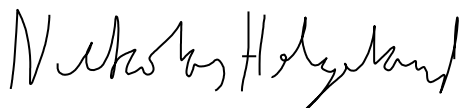
Denne gruppeavtalen gjelder for Bacheloroppgave ved 6. semester av Bachelor i ingeniørfag. Avtalen gjelder fra 11.01.2020 til og med 18.06.2021

### Krav til prosjektdeltagere

1. Alle møter til avtalt tid og sted
2. Avtalte arbeidsoppgaver skal være gjort til avtalt tid
3. Alt fravær og forsinkelser skal varsles ifra snarest mulig via felleskommunikasjonsplattform
4. Problemer med å følge framdriften i prosjektet skal meldes ifra til gruppen. Deretter tas det opp i et uformelt møte
5. Respektere hverandres synspunkter
6. Alle i gruppen skal samarbeide om å nå felles mål
7. Personlig og unødvendig bruk av mobiltelefon i møtesammenheng er ikke tillatt. Mobiltelefon kan brukes som et arbeidsverktøy hvis nødvendig
8. Er det uenighet i gruppen, skal en tredjepart fra skolen brukes som rådgiver
9. Rutine ved brudd på reglene:
  1. Prøve å løse problemet sammen med gruppemedlemmene
  2. Levere skriftlig advarsel til vedkommende fra de andremedlemmene der:
    - i. Regelbruddet blir påpekt
    - ii. Beskrive hva som kreves av vedkommende for å løse saken
    - iii. Fortelle om evt. konsekvenser om vedkommende ikke følger oppadvarselen
  3. Samtale med veileder for å løse problemet

MS Teams/Zoom, 11.01.2021

Nickolas Helgeland



Jan Erik Lunde





## Vedlegg E – Brukermanual Web-applikasjon

# Brukermanual Web-applikasjon

Denne brukermanualen vil ta for seg bruk av Web-applikasjon for sluttbruker og administrator. Brukermanualen er delt i to deler og består av:

- Sluttbruker
- Administrator

## Sluttbruker

Når Web-applikasjonen åpnes så er det første som møter brukeren en innloggingsside, som vist i Figur 2, her må brukeren logge inn med en gyldig bruker. Verktøysansvarlig har ansvar for å opprette gyldig bruker for alle de ansatte.

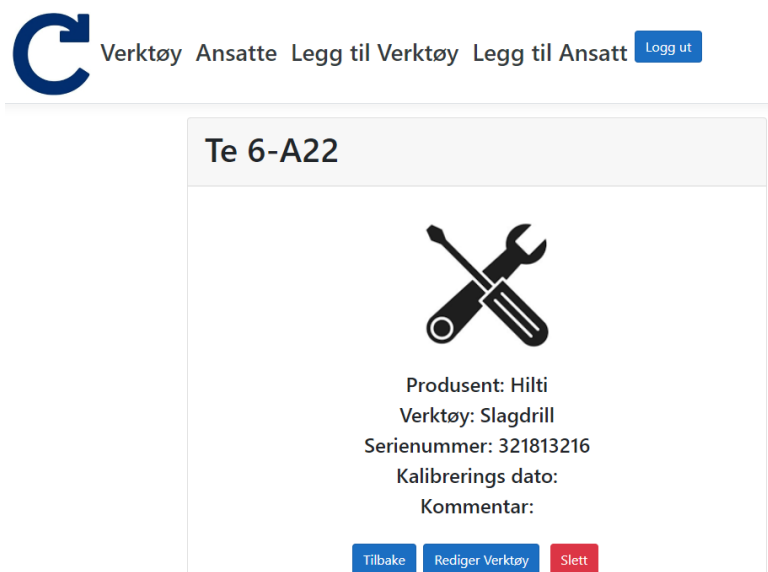
Figur 2: Innloggingsside

Etter brukeren har logget seg inn så blir brukeren møtt med verktøylisten, som vist i Figur 3. Verktøy listen skal gi brukeren en oversikt over hvilke verktøy som er tilgjengelig og utlånt. I verktøylisten så er modellnavnet hevet ut med blå farge.

Verktøynummer	Verktøy	Modell	Site ansvarlige person	Kalibrerings Dato	Kommentar
1	Slagdrill	<a href="#">Te 6-A22</a>			
2	Fibertester	<a href="#">CertFiber Pro Quad</a>			
3	Datatester	<a href="#">DSX-500</a>			
4	Fibermikroskop	<a href="#">Vensiv</a>			
26	Vinklekutter	<a href="#">AG 125-A22</a>			
6	Bajonettag	<a href="#">SR 6 A-22</a>			
7	Bajonettag	<a href="#">Bajonettag1</a>			
8	Muttertrekker	<a href="#">SIW 22A</a>			
9	Muttertrekker	<a href="#">Muttertrekker1</a>			
10	Muttertrekker	<a href="#">Muttertrekker2</a>			

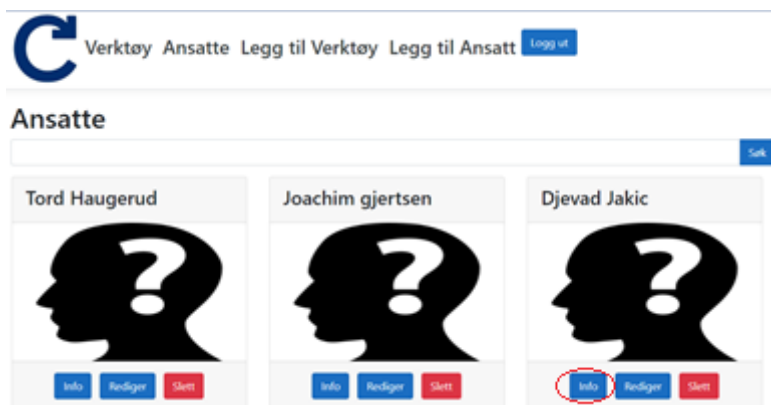
Figur 3: Verktøyside

Ved å trykke på modell navnet så blir brukeren sendt til informasjonssiden for verktøy, som vist i Figur 4. Her får brukeren mer informasjon om verktøyet og et visuelt bilde av verktøyet. Verktøysansvarlig har ansvar for å legge til bilde til hvert verktøy.



Figur 4: Informasjonsside for verktøy

Videre så kan brukeren trykke på Ansatte i menyen for å få opp liste over de forskjellige ansatte, som vist i Figur 5. Her har brukeren mulighet til å trykke på «Info» knappen for å få mer informasjon over den enkelte ansatte. Brukeren blir også sendt til informasjonssiden av den enkelte ansatte fra verktøylisten. I verktøylisten så er det et eget felt for utlåneren av et verktøy. Utlåneren er hevet ut i blå farge slik modell navnet er.




Figur 5: Ansattside

Ved å klikke på navnet til utlåneren, fra verktøylisten, eller «Info» knappen, på ansattlisten, så blir brukeren sendt til informasjonssiden for ansatte, som vist i Figur 6. Her får brukeren telefonnummeret til den enkelte ansatte, samt et bilde for å lettere igjen kjenne en ansatt. Dette gir brukeren mulighet til å ringe til den enkelte ansatte som låner et verktøy som brukeren selv ønsker å bruke.

## Informasjon om ansatt

Kim Roger Haugseter



Telefon : 0

[Tilbake](#) [Rediger](#) [Slett](#)

Figur 6: Informasjonsside for ansatte

## Administrator

Web-applikasjonen har også en administratorfunksjon. Denne funksjonen er ment å bli brukt av verktøyansvarlig. Som administrator så har brukeren mulighet til å redigere, slette og legge til verktøy og ansatt.

For å legge til verktøy så trykker brukeren på «Legg til Verktøy» i menyen og blir sendt til opprettelsessiden, som vist i Figur 7. Her får brukeren mulighet til å legge til all informasjonen om et verktøy og legge til et bilde av verktøyet. Ved å trykke på «Lagre» så har brukeren opprettet et nytt verktøy. Ved å trykke på «Avbryt» så blir brukeren sendt tilbake til verktøylisten.

Verktøy Ansatte Legg til Verktøy Legg til Ansatt [Logg ut](#)

### Opprett Verktøy



RFID	<input type="text" value="0"/>
Verktøynummer	<input type="text" value="0"/>
Modell	<input type="text" value="Modell"/>
Produsent	<input type="text" value="Eks: Hilti"/>
Verktøy	<input type="text" value="Eks: Drill"/>
Kommentar	<input type="text" value="Eks: Verktøyet ble ødelagt"/>
Serienummer	<input type="text" value="Eks: 11223344"/>
Innkjøpsdato	<input type="text" value="mm/dd/yyyy"/>
Påminnelse	<input type="text" value="Påminnelse"/>
Pris	<input type="text" value="Eks: 100 000Kr"/>
Tilstand	<input type="text" value="Eks: God"/>
Kalibreringsdato	<input type="text" value="mm/dd/yyyy"/>
Kalibreringssted	<input type="text" value="Eks: Caverion, Herøya"/>
Bilde	<input type="text" value="Klikk her for å endre bilde"/> <a href="#">Browse</a>

[Lagre](#) [Avbryt](#)

Figur 7: Opprettelsesside for verktøy

Ved redigering av verktøy så trykker brukeren på «Rediger Verktøy» fra informasjonssiden og blir sendt til redigeringssiden, som vist i Figur 8. Her kan brukeren endre informasjonen for det enkelte verktøyet. Ved å trykke på «Lagre» så lagrer brukeren den nye informasjonen til verktøyet. Ved å trykke på «Avbryt» så blir brukeren sendt tilbake til verktøylisten.

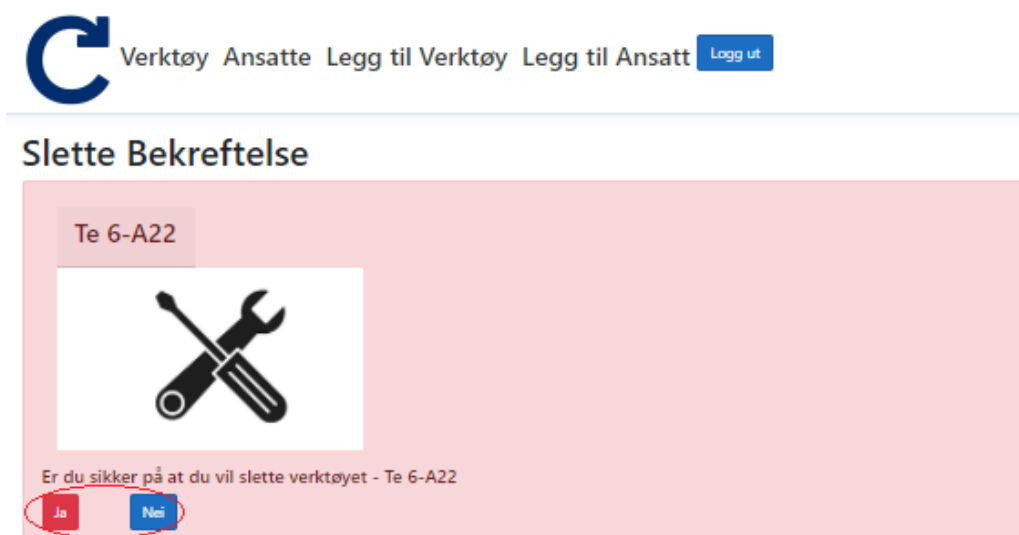
## Rediger Verktøy Te 6-A22

Verktøynummer	<input type="text" value="1"/>
Produsent	<input type="text" value="Hilti"/>
Verktøy	<input type="text" value="Slagdrill"/>
Modell	<input type="text" value="Te 6-A22"/>
Serienummer	<input type="text" value="321B13216"/>
Innkjøpsdato	<input type="text" value="mm/dd/yyyy"/> 
Pris	<input type="text" value="6,000"/>
Påminnelse	<input type="text" value="60"/>
Tilstand	<input type="text" value="Eks: God"/>
Kalibreringsdato	<input type="text" value="mm/dd/yyyy"/> 
Kalibreringssted	<input type="text" value="Eks: Caverion, Herøya"/>
Kommentar	<input type="text" value="Eks: Verktøyet ble ødelagt"/>
Bilde	<input type="text" value="Klikk her for å endre bilde"/> <a href="#">Browse</a>



Figur 8: Redigeringside for verktøy

For å slette et verktøy så må brukeren trykke på «Slett» fra informasjonssiden og blir sendt til en slette bekreftelse, som vist i Figur 9. Slette bekreftelsen viser modellnavnet til valgt verktøy og bilde av verktøyet, for å lett igjen kjenne valgt verktøy. Her får brukeren valget om å slette eller avbryte slettet av valgt verktøy. Ved å trykke på «Ja» så sletter brukeren verktøyet. Ved å trykke på «Nei» så blir brukeren sendt tilbake til verktøylisten.



Figur 9: Slette bekreftelse for verktøy

For å legge til ansatt så trykker brukeren på «Legg til Ansatt» i menyen og blir sendt til opprettelsessiden, som vist i Figur 10. Her får brukeren mulighet til å legge til all informasjonen om en ansatt og legge til et bilde av den ansatte. Ved å trykke på «Lagre» så har brukeren opprettet en nytt ansatt. Ved å trykke på «Avbryt» så blir brukeren sendt tilbake til ansattlisten.

Figur 10: Opprettelsesside for ansatte

Ved redigering av en ansatt så trykker brukeren på «Rediger Ansatt» fra informasjonssiden og blir sendt til redigeringsiden, som vist i Figur 11. Her kan brukeren endre informasjonen for den enkelte ansatte. Ved å trykke på «Lagre» så lagrer brukeren den nye informasjonen til den ansatte. Ved å trykke på «Avbryt» så blir brukeren sendt tilbake til ansattlisten.

Figur 11: Redigeringside for ansatte

For å slette en ansatt så må brukeren trykke på «Slett» fra informasjonssiden og blir sendt til en slette bekreftelse, som vist i Figur 12. Slette bekreftelsen viser navnet til den valgte ansatt og bilde av den ansatte, for å lett igjen kjenne den ansatte. Her får brukeren valget om å slette eller avbryte slettet av valgt ansatt. Ved å trykke på «Ja» så sletter brukeren den ansatte. Ved å trykke på «Nei» så blir brukeren sendt tilbake til ansattlisten.

Figur 12: Slettebekreftelse for ansatte

Når brukeren skal forlate nettsiden så trykker brukeren på «Logg ut» og blir sendt tilbake til innloggingssiden.

## Vedlegg F – Brukermanual RFID-applikasjon

# Brukermanual RFID-applikasjon

Denne brukermanualen vil ta for seg oppsettet av RFID-applikasjonen og koblingen av Pepperl + Fuchs sitt RFID-system. Brukermanualen er delt opp i to deler og består av:

- Kobling av RFID-system
- Installasjon og bruk av RFID-applikasjonen

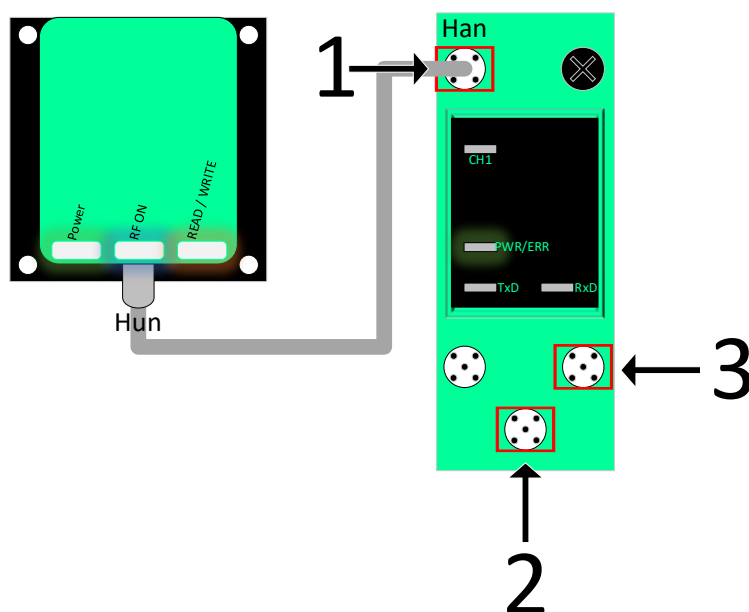
## Kobling av RFID-system

RFID-systemet består av:

1. UHF RFID-leser (IUH-F190-V1-FR1-01).
2. kontrollgrensesnitt (IC-KP2-2HRX-2V1).
3. rfIDEAS Wave ID LF-HF RFID-leser
4. Forbindelseskabel M12 han til hun konnektor (V1-G-5M-PUR-ABG-V1-W)
5. M12 konnektor hunnplugg med kabel for 24V DC spenning (V1-G-5M-PUR-ABG).
6. Adapter kabel M12 til Sub-D 9-pin (V1S-G-5M-PUR-ABG-SUBD9)
7. Serial adapter D-Sub 9-pin til USB Type A

Komplett utstyrliste er å finne i rapporten (kom tilbake med referering). Det vil ikke bli gått gjennom hvordan systemet skal monteres, da dette er

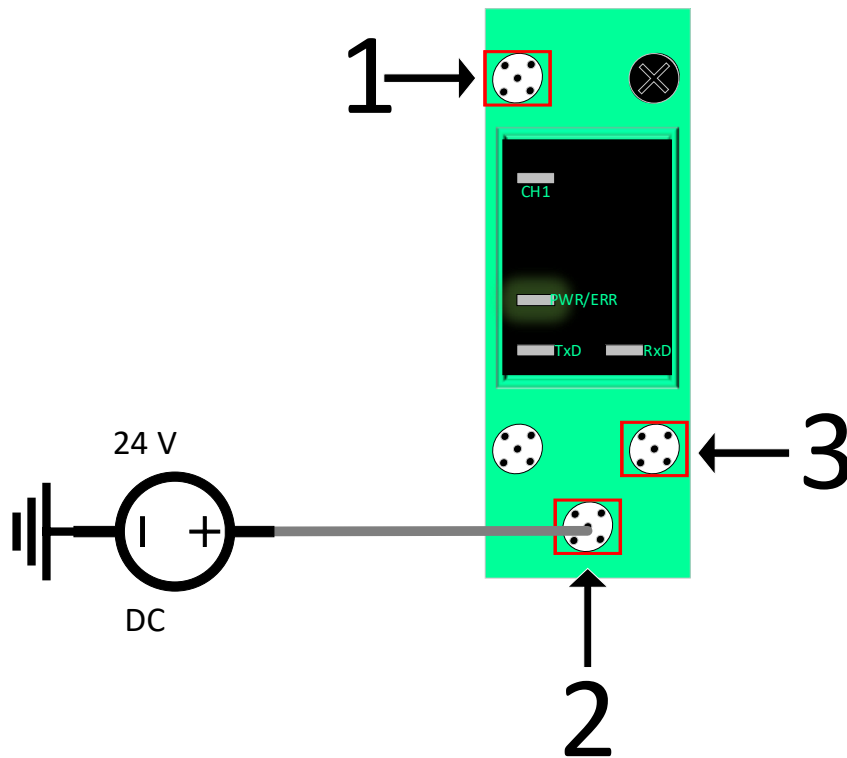
1. For å koble sammen systemet tar man først og kobler sammen RFID-leseren til IDENT-Controlleren med en forbindelseskabel med han og hunnplugg. Hanpluggen settes inn i IDENTcontrolleren og hunnpluggen setter inn i RFID-leseren som anvist i Figur 13.



Figur 13: RFID-leser koblet til IDENTController

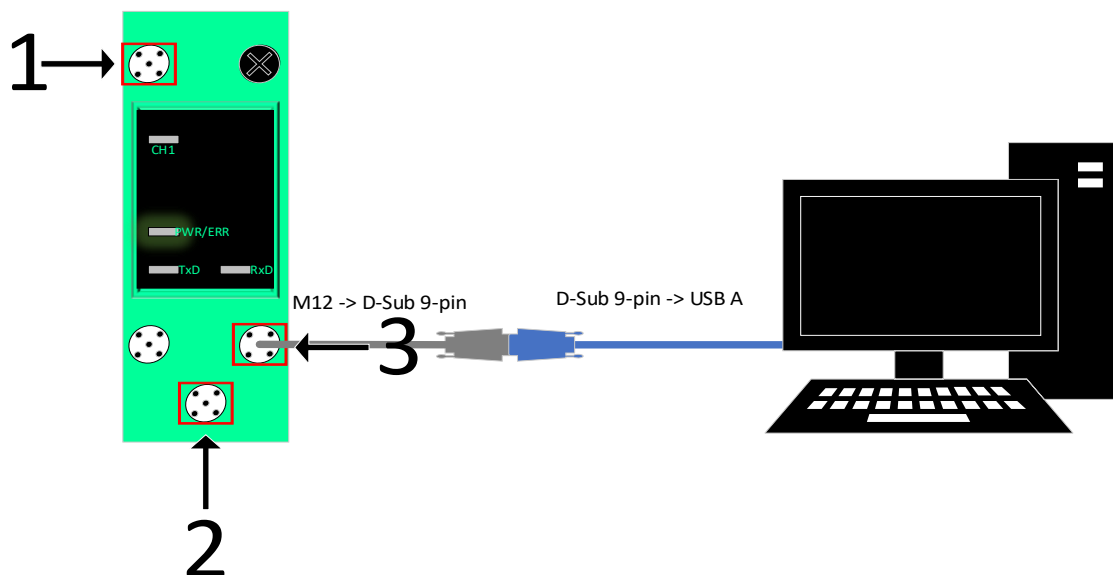


2. Videre i kobles IDENTControlleren til en 24V spenningskilde med en M12 hanplugg som anvist i Figur 14. Her kan det være lurt å installere en bryter for å skru den av og på.



Figur 14: IDENTController koblet til 24 V spenningskilde

3. Videre i kobles IDENTControlleren til datamaskinen med en M12 hanplugg til D-sub 9-pin og en serial adapten videre inn i datamaskinen som anvist i Figur 15 og oppsettet av RFID-systemet er ferdig. For mer detaljert informasjon se brukermanual for RFID-leser og IDENTControll.

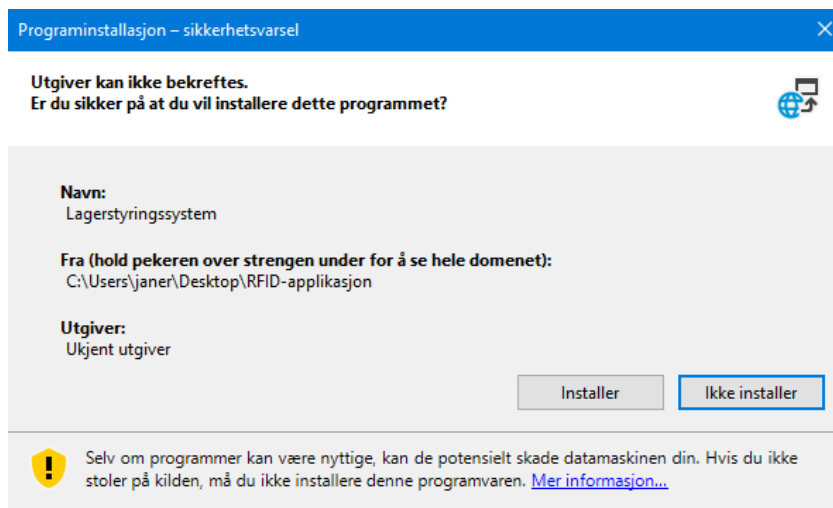


Figur 15: IDENTController koblet til Datamaskin

1. Helt til slutt kobles rfIDEAS RFID-leseren inn i datamaskinen med USB kabel. Driveren for leseren vil bli installert automatisk.

## Installasjon av RFID-applikasjon

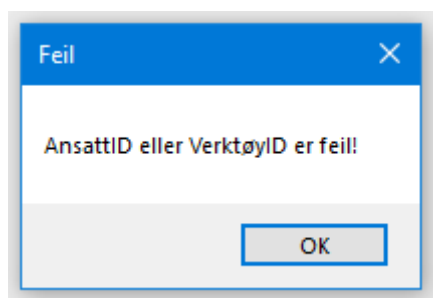
Pakk ut zip-filen til en egnet mappe enten på skrivebordet eller på disken. Når zip-filen er pakket ut må man kjøre setup.exe filen. Videre trykker man på installer for å installere applikasjonen, se Figur 16. Når installasjonen er ferdig, vil applikasjonen starte opp. Applikasjonen lager en snarvei på skrivebordet. Hvis applikasjonen ikke er å finne på skrivebordet kan den søkes opp i startmenyen med søkeord «Lagerstyringssystem».



Figur 16: Ledetekstvindu for installasjon

## Bruk av RFID-applikasjon

Når du åpner applikasjonen, vil den automatisk åpne seg i full skjerm. I brukergrensesnittet vil det være et inntastingsfelt for Ansatt-ID og et for Verktøy-ID. I feltet for Ansatt-ID skal det **ikke** tastes inn noe manuelt, med mindre et verktøy skal leveres eller lånes manuelt. Det samme gjelder feltet for Verktøy-ID. Ved en manuell registrering så fylles inntastingsfeltene inn med Ansatt-ID og Verktøy-ID, så trykker man på registrerknappen. Når rfIDEAS RFID-leseren leser inn et ansattkort vil det bli skrevet inn i Ansatt-ID feltet. Hvis ikke systemet gjenkjenner en Ansatt-ID eller Verktøy-ID vil det komme opp en feilmelding som vist i Figur 17.



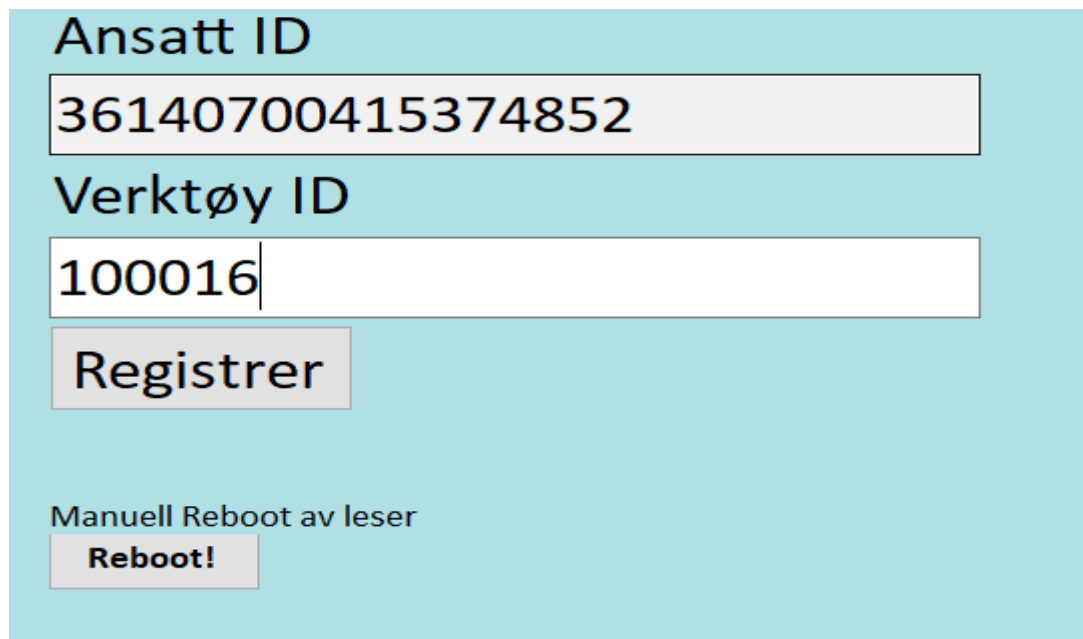
Figur 17: Feilmelding ved ugyldig inntasting av Ansatt-ID eller Verktøy-ID.

Når Ansatt-ID er lest inn og godkjent, vil feltet for inntasting bli låst slik at det ikke kan skrives til som vist i Figur 18. Pepperl + Fuchs RFID-leseren vil deretter begynne å lese helt til den finner en gyldig verktøybrikke.

A screenshot of the application's main interface. It has a light blue background. At the top, it says "Ansatt ID" in bold. Below that is a text input field containing the number "36140700415374852". Underneath is another text input field labeled "Verktøy ID", which is currently empty. Below the input fields is a button labeled "Registrer". At the bottom, there is a section titled "Manuell Reboot av leser" with a button labeled "Reboot!".

Figur 18: Innlesing av gyldig Ansatt-ID.

Når RFID-Leseren finner en verktøybrikke blir dataen fra brikken lest inn i inntastingsfeltet for Verktøy-ID som vises i Figur 19. Hvis ikke Verktøy-ID er gyldig vil det komme opp en feilmelding som vist i Figur 17.



The screenshot shows a light blue interface with the following elements:

- Ansatt ID**: A text box containing the number 36140700415374852.
- Verktøy ID**: A text box containing the number 100016.
- Registrer**: A button located below the Tool ID field.
- Manuell Reboot av leser**: A label above a **Reboot!** button.

Figur 19: Ferdig utfylt Ansatt-ID og Verktøy-ID.

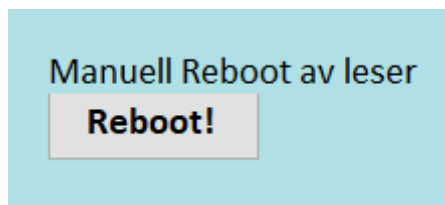
Når systemet har lest inn gyldige ID-er vil det enten utføre en utlåns- eller leveringstransaksjon. Dette skjer automatisk og vil vises i feltet for historikk. I Figur 20 vises det hvordan en utført transaksjon blir grafisk fremstilt.



<b>Nickolas Helgeland 44664466 23.05.2021 20:07:50</b> <b>Hilti SB 4-A22 Status: Utlånt</b>
--

Figur 20: Eget felt for utlån/leveringshistorikk.

Hvis det skulle skje at inntastingsfeltet for Ansatt-ID forblir låst eller at RFID-leseren henger seg opp, trykker man på Reboot-knappen. Denne knappen fjerner låsen på inntastingsfeltet og tilbakestiller RFID-leseren til dvalemodus. Reboot-knapper er vist i



Figur 21: Reboot knapp for lesere og inntastingsfelt.