2023

# Asset Management System

BYGGET MED DOCKER OG MERN

SIMEN FRITSVOLD

KUBEN VGS | Kabelgaten 10-12

# Table of Contents

# Beskrivelse av mitt prosjekt

## Konseptet

Min oppgave går ut på et system der man kan låne inn og ut utstyr som pc-er, kameraer og mer. Det skal være en oversikt over alt utstyr lagret på en database, når noen låner ut noe så vil det bli lagret koblinger som sier hvem, hva og når. Det skal være mulig å se på denne dataen igjennom nettsiden min.

## Teknologien jeg bruker

### Docker

For min struktur har jeg valgt å bruke docker compose, det er en teknologi der de forskjellige delene blir separert i ulike containere som selv kjører alene. Det er også en enkel løsning for å ordne nettverk siden docker setter opp et eget nettverk imellom containere. Det docker også hjelper til med er at det er lett å kjøre opp hele applikasjonen med en gang og at det også er uavhengig av operativsystem.

### Express

For å sende ut min nettside som er laget i React, så bruker jeg Express. Det er et Javascript rammeverk som hjelper deg med å enkelt skrive web-servere på veldig få linjer, men fortsatt lage kompliserte systemer. Jeg bruker det til å holde styr på sessions og snakke med sluttbrukeren. Denne web-serveren snakker igjen med min Python API som ligger lenger bak i systemet.

### Fast-api

For å koble alt sammen har jeg valgt å bruke fast API Python rammeverket. Det gjør det enkelt å sette opp en API som kan sjekke passord med videre og snakke med databasen. Den skal virke mest som et mellomledd mellom databasen og web-serveren slik at koden ikke ligger ut mot internett.

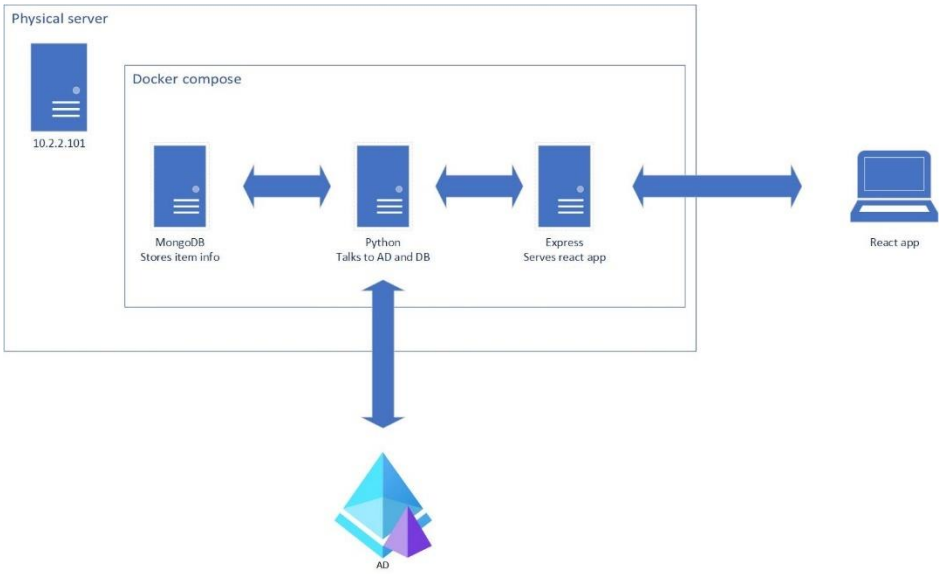
### MongoDB

Databasen jeg har valgt å bruke er MongoDB. Dette er siden det er en dokumentdatabase som jeg syntes passet veldig bra med den typen data jeg skulle lagre. Når vi har ulikt type utstyr så gir det mening å bruke en slik type database som enkelt takler uregelmessige data.

# Hvordan jeg planlegger å utføre dette

## Nettverksdiagram

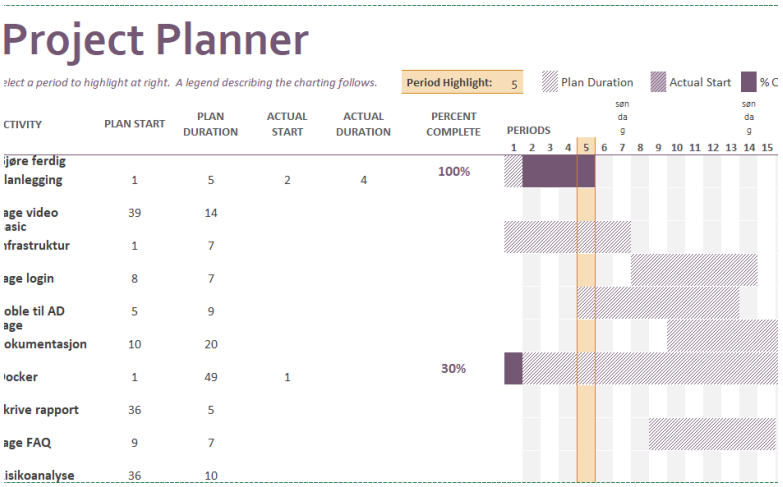
Jeg startet hele prosjektet med å først lage en tegning i Visio. Dette skulle være en plan for hvordan jeg skulle bygge opp hele systemet. Her har man styr over de ulike docker containerene og hva de inneholder.




Jeg tenker også å eventuelt oppdatere dette kartet ettersom det utvikler seg videre og blir annerledes.

## Holde styr på timebruken i Excel

Jeg holder også styr på hva som må gjøres og hva som har blitt gjort i Excel. Det er noe som oppdateres av og til slik at jeg selv kan holde styr. Se Excel filen for mer detaljer.




# Teknisk dokumentasjon

For å se min tekniske dokumentasjon så har jeg andre filer i samme mappe som denne hvor man kan se det.

# Lover og regler

## Personvernloven

Siden jeg behandler persondata i systemet mitt, er «Lov om behandling av personopplysninger» relevant.

I min database så lagrer jeg persondata om folk som låner utstyr. Denne dataen er nødvendig for systemet og jeg lagrer ikke noe mer enn det som trengs. Med denne dataen sørger jeg for at skolen har oversikt over hvem som har ansvar for hva. Jeg har ikke lagt inn en egen måte å slette data om seg selv på, men jeg kunne lagt en måte å søke opp navn også slette alt som har med det navnet å gjøre.

## Universell utforming

Siden systemet skal brukes av ulike typer folk er kapittel 3 «Universell utforming og individuell tilrettelegging» i «Lov om likestilling og forbud mot diskriminering» relevant.

Med tanke på at dette er et system som kan brukes i skole sammenheng så har jeg tatt hensyn til universell utforming. Dette har jeg gjort gjennom å lage et oversiktlig utseende på nettsiden, som bruker riktige HTML-tags slik at en blind person kunne forstått fra opplesning hva som skal gjøres.

## Ulike risikoer og hvordan håndtere dem

| **Mulig uønsket hendelse/ belastning** | **Vurdering av sannsyn-lighet** | **Vurdering av konsekvens:** | | | | **Risiko-verdi** | **Kommentarer/status**<br>**Forslag til tiltak** |
|---|---|---|---|---|---|---|---|
| | (1-5) | Menneske (A-E) | Ytre miljø (A-E) | Øk/ materiell (A-E) | Om-dømme (A-E) | | |
| DDos angrep | 2 | E | E | D | C | D | Tjenesten er ikke tilgjengelig fra internett så det vil være interne brukere som ddos-er.<br>Tiltak: Styre hvor mange spørringer en vanlig bruker kan gjøre per sekund. |
| Personinformasjon blir stjålet | 2 | A | E | E | A | A | Løsningen trenger å bli testet av noen som kan sikkerhet.<br>Tiltak: Få Arvid til å prøve å hacke siden. |
| Sårbarheter i egen kode eller avhengigheter. | 3 | B | E | C | A | B | React delen har mye avhengigheter.<br>Tiltak: Sørge for at man bruker de |

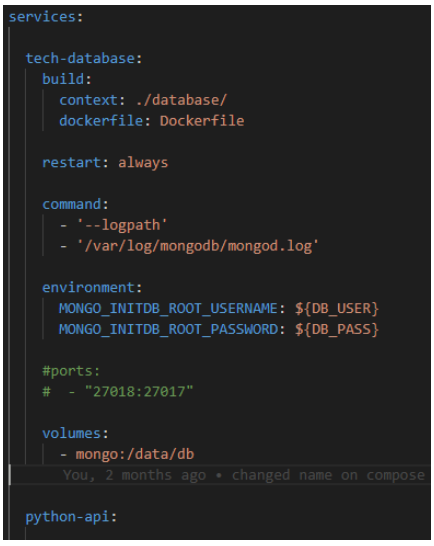
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | nyeste bibliotekene. Automatisk sjekking av prosjektet ditt på github. |
| Feil i koden ved videreutvikling. | 4 | C | E | A | B | B | Koden kan bli veldig komplisert over lang utvikling. Kan føre til uoversiktlig kode.<br><br>Tiltak: Skrive tester for koden slik at når man gjør en endring vet man om det funker. |

***Konsekvens***
*A. Svært alvorlig*
*B. Alvorlig*
*C. Moderat*
*D. Liten*
*E. Svært liten*

***Risikoverdi (beregnes hver for seg):***
***Menneske = Sannsynlighet x Konsekvens Menneske***
***Ytre miljø = Sannsynlighet x Konsekvens Ytre miljø***
***Økonomi/materiell = Sannsynlighet x Konsekvens Øk/matriell***
***Omdømme = Sannsynlighet x Konsekvens Omdømme***

# Backup rutiner

For å ta backup så må man gå inn og endre litt på koden. Man må gå i docker compose filen og se etter port under databasen. Her skal man gjøre den gjøre dette til en del av koden ved å fjerne hashtagsene. Da blir databasen tilgjengelig på den fysiske maskinen. Da kan man bruke «mongodump» som er et verktøy for å ta backup av databasen.




# Kilder

https://docs.docker.com

https://www.w3schools.com

https://expressjs.com/en/4x/api.html

https://fastapi.tiangolo.com