

# Management Systems ApS

KEA Webudvikling Bachelorprojekt

Forår 2024

Af Peter Frankild

Vejleder: Santiago Donoso

Antal tegn: 64829

Afleverings dato: 13.08.2024

Gitlab repository 14.06.2024:

<https://gitlab.com/bachelorprojekt2/managementsystems>

[https://gitlab.com/bachelorprojekt2/management\\_systems\\_landing\\_page](https://gitlab.com/bachelorprojekt2/management_systems_landing_page)

Test deploy af Applikationer 13.08.2024:

# Tilføjet MailChimp til kontaktformular.

[https://bachelorprojekt2.gitlab.io/landingpage\\_managementsystems](https://bachelorprojekt2.gitlab.io/landingpage_managementsystems)

# Tilføjet 33 test af views.py filer, slettet 2 def funktioner, optimeret indents og farve variabler.

<https://www.kea-dev-managementsystems.xyz>

Login roller: admin\_projects subsystem1 @Pc123456 user\_projects subsystem1 @Pc123456

# Indholdsfortegnelse

|       |                                  |    |
|-------|----------------------------------|----|
| 1     | Introduktion                     | 4  |
| 2     | Problemformulering               | 4  |
| 3     | Målformulering                   | 4  |
| 4     | Projektplanlægning               | 5  |
| 5     | projektets muligheder            | 5  |
| 6     | Konkurrerende virksomheder       | 6  |
| 6.1   | Virksomhedsanalyse               | 6  |
| 7     | Programmeringssprog og Framework | 7  |
| 7.1   | Python                           | 7  |
| 7.1.1 | Flask                            | 8  |
| 7.1.2 | Django                           | 9  |
| 7.2   | PHP                              | 9  |
| 7.2.1 | Laravel                          | 10 |
| 7.2.2 | Symfony                          | 11 |
| 7.3   | Dashboard                        | 11 |
| 8     | Mobile Framework                 | 11 |
| 9     | CSS                              | 12 |
| 10    | Databaser                        | 12 |
| 10.1  | Relational databaser             | 12 |
| 10.2  | Dokument databaser               | 13 |
| 10.3  | Research af databaser            | 13 |
| 10.4  | Entities Relation diagram        | 14 |
| 11    | Kravspecifikationer              | 15 |
| 12    | Business Model Canvas            | 15 |
| 12.1  | Stakeholders                     | 15 |
| 12.2  | User stories                     | 16 |
| 12.3  | Virksomhedstype                  | 16 |
| 13    | Sitemap                          | 16 |
| 14    | Wireframes                       | 16 |
| 15    | Scenarios                        | 17 |
| 15.1  | Key-path scenarios storyboards   | 19 |
| 16    | Adfærds variabler                | 21 |
| 17    | persona                          | 22 |
| 18    | UX og Design regler              | 23 |
| 18.1  | Farve og font                    | 23 |

|                               |             |
|-------------------------------|-------------|
| 18.2 Ikoner                   | 23          |
| 19 Udviklingsstrategi         | 23          |
| 19.1 Delmål                   | 24          |
| 19.2 Endeligt mål             | 24          |
| 20 Valg af Software           | 25          |
| 20.1 Planlægningsværktøjer    | 25          |
| 20.2 EDI                      | 25          |
| 20.3 CI/CD                    | 26          |
| 20.4 AI & ChatGPT             | 26          |
| 21 Framework                  | 26          |
| 21.1 Fil struktur             | 27          |
| 21.2 Engine                   | 28          |
| 21.3 Billeder                 | 28          |
| 21.4 Pagination               | 29          |
| 21.5 Rest API og json fetch   | 29          |
| 21.6 Bruger authorization     | 30          |
| 21.7 Bruger authentication    | 30          |
| 21.8 SSL Sikkerhed            | 30          |
| 21.9 Routing                  | 31          |
| 22 Software quality assurance | 31          |
| 23 Version control on Gitlab  | 31          |
| 24 Test                       | 32          |
| 25 Pipeline og VPS hosting    | 33          |
| 26 Docker med Nginx           | 33          |
| 27 Indsigter                  | 34          |
| 28 Den videre udvikling       | 34          |
| 29 Konklusion                 | 35          |
| 30 Bibliografi                | 36          |
| 31 Bilag                      | Vedlagt zip |

## 1 Introduktion

Denne rapport gennemgår udvikling af et digitalt ledelsessystem til planlægning af arbejdsdage og opgaver for virksomhedens medarbejdere. Udviklingen pågår over 8 uger og skal ses som et Start-up projekt, hvor der skal opnås interesse og midler til en efterfølgende udviklingsproces. Første fase har den målsætning at præsentere en solid og sikker digital udviklingsplatform, som kan bruges til test før den endelige udviklings fase begynder.

Platformen skal indeholde et admin dashboard med superbruger permissions, et administrations dashboard for administrative medarbejdere og et arbejdsplan dashboard for medarbejderne. Det skal være muligt at kommunikere mellem administrationen og andre medarbejdere. Ud over tidligere nævnt funktionaliteter skal ledelsessystemet som minimum tilbyde medarbejderne at kunne administrere deres daglige tid, som live kan følges af administrationen.

## 2 Problemformulering

Er det muligt at finde en skalerbar digital framework, der er i stand til at udnytte nutidens webteknologier, både hvad angår krav til funktionaliteter og krav til kvalitetssikring af software?

## 3 Målformulering

Målet med projektet er at starte udviklingen af en digital ledelses platform, som opfylder stakeholders krav til funktionalitet, brugervenlighed og brugergrænseflade.

Der skal udvikles et designvalg ud fra User-centered design, som er en interagere designprincipper ved at udnytte brugerdata. Der skal identificeres Pain Points, Build Personas, Ideation, Scenarios, UI Prototype, Usability Testing og den digitale produktudvikling. Der skal imødegås stakeholders holdninger under produktudviklingen ved brug af Sitemap, Wireframe og beskrivende prototyper, som forelægges til videre evaluering.

Der skal tage stilling til valg af teknologier indenfor programmeringssprog, valg af framework, databaser, og cross platform mobile løsninger, ud fra hvilke perspektiver udviklings processen herunder sværhedsgrad og konkurrence aspekter indbefatter. Hvilke muligheder er der for at oprette AIP-funktionalitet til integration af andre software udbydere.

Opsætning af server side, herunder Virtuel Personal Server Drev, er der andre brugbare server-less løsninger og hvilke no-code software er tilgængelige og har de integreret Cloud Solution deployment.

Hvilke muligheder er der for containerisation med Docker og Nginx server og hvilke andre tendenser er oppe i tiden, når det gælder Software quality assurance CI/CD pipeline, herunder brug af Git source control også kendt som Version control systems med godkendelse og tracking af ændringer i programmering af applikationen.

## 4 Projektplanlægning

Der er en tidsramme på 9 uger, hvor den første uge har fokus på den faktiske forberedelse af en brugbar projekt opsætning, herunder opsætning af online ressourcer, Trello, Google Drev, Figma, Gitlab og andre brugbare kilder og deres dokumentation. De næste tre uger bruges på at undersøge teknologier, der med fordel kan erstatte allerede kendte løsninger. Herefter følger en uges test for at sikre, at der ikke kan ske væsentlige setback i udviklingsfasen og den efterfølgende produktionsfase.

## 5 Projektets muligheder

Projektet indeholder muligheder for at skabe et solidt management system, der tager højde for en daglig tidstagnation til ens ansatte, en platform som både tager hensyn til arbejdsgiver brugeroplevelse gennem et nøje tilrettelagt dashboard og til den ansattes overblik og forståelse af de enkelte dages arbejdsopgaver og tidsforbrug ved tidtagning og akkumulering af ugens forbrug. I sidste ende skal konceptet tilbydes som en mobile app-løsning til download fra App Store eller Google Play. Et koncept som Management Systems kan udvikles i mange retninger. Der kan kommunikeres og tegnes og illustreres mellem ansatte og gives et gennemsigtigt lønregnskab, som løses direkte i systemet, på ledelsens dashboard. Et samlet produkt bør indeholde kundeliste, samarbejdspartnere og regnskab med lagerstyring og kassekladder. Platformen vil også kunne indeholde analytics af forskellige mål med interesse for at kunne imødegå ændringer på markedet og i arbejdsprocessen.

## 6 Konkurrerende virksomheder

### Konkurrerende virksomheder - Management system

Experience map

<https://www.tamigo.com>

#### Brugermål

- Bække Tamigo virksomhedens fulde behov
- Skabes der værdi for den ansatte
- Vagtplan med time opgørelse af ansatte
- Mindske administrationsomkostninger på ansatte
- Forbedre arbejdsmiljøet gennem kommunikation
- Data driven beslutningstagning
- Cross platform integration
- Mobilløsninger

#### Bruger actions

- Prøve gratis demo
- Tag stilling til hvilke behov virksomheden har
- Kontakt Tamigos kundeservice
- Vælg en pakke pris

#### Pain points

- Rettet mod retail, restauranter og hoteller
- Fungere Tamigos som brand
- Er der en stærk support mulighed
- Kan der ske forbedringer, vedligeholdelse
- Har Tamigo andre stakeholders på markedet

#### Forbedringer

- Der mangler en stærk branding især UI
- Intern løn beregning sikre data processen
- En fuldstændig FAQ sektion
- Synlig sundhed i virksomheden
- Budget og regnskab
- ip - end to end kommunikation

Experience map

<https://www.billy.dk/>

#### Brugermål

- Regnskabsprogram for selvstændige
- Send faktura, bogfør, moms og regnskab
- Lønssystem med bogføring i regnskabet
- Få en gratis konto
- Kontakte Billy, support og FAQ
- Integrationsmuligheder vs fuld dækning af administrative opgaver

#### Bruger actions

- Prøve gratis konto
- Tag stilling til hvilke behov virksomheden har
- Læs support artikler
- Kontakt Billys kundeservice
- Vælg en pakke pris

#### Pain points

- Er produktet kun for selvstændige
- Fungere Billy som brand
- Er der en stærk support mulighed, tutorials
- Kan produktet testes
- Kan produktet forbedres
- Kan produktet vedligeholdes
- Har Billy andre stakeholders på markedet

#### Forbedringer

- Branding især skræddersyet UI
- Hvad betales der efter forbrug
- Hjælp til setup
- Mangler lagerstyring
- Test periode af integrering
- Sikkerhed for videreførsel af data

#### 1. Experience map, konkurrerende virksomheder

### 6.1 Virksomhedsanalyse

Der er foretaget en virksomhedsanalyse af Tamigo og Billy, hvor der fremstår nogle pain points, som er relateret til hvilke brugere mål, hvilke interesse, der er de mest synlige i forbindelse med hvilke management der tilbydes på virksomhedens hjemmeside, og hvilke brugerhandlinger, der er valg foretages, når de bruger hjemmesidens brugergrænseflade. Også brugerens forståelse af UX ses gennem signifiers, signalværdier, call to actions, indhold og de forventninger der er til brugerens valg, på en hjemmeside, som dem i analysen.

I virksomhedsanalysen er der nogle sammenlignende pain points, som peger på nogle forbedringer som vil styrke hjemmesiderne og som kan være noget Management Systems vil kunne bruge i en udviklingsfase.

Ser vi først på Tamigo, er der tegn på, at det kan være svært at se en brugerværdi ud fra den branding som er valgt. På den anden side er der en god forståelse for brug af data fra analytics. Der er også en god brug af og forståelse for integration med andre virksomheders tjenester. De første valg kunden står over for, er at sikre at tilbuddet er det helt rigtige, at de skal bruge demo-siden og at kontakt med kundeservice, vil hjælpe med at finde det rigtige tilbud.

Ser vi på Billy, er deres tilbud et mere samlet intern regnskabsløsning med konto liste, løn og administrative opgaver. Der er mulighed for at få hjælp i et udbygget FAQ sektion.

Brugerhandlingerne er næsten de samme her, som hos Tamigo.

Gennemser vi pain points der opstår på disse sider, kan vi finde forbedringer og tage højde for dem i Management Systems. Der skal lægges vægt på branding af siden, testmuligheder, betalingens gennemsigtighed og en god hjælp sektion, f.eks. FAQ på en landing page.

## 7 Programmeringssprog og Framework

De fleste frameworks er som udgangspunkt designet med open source programmeringssprog og er derfor skalerbare og inspirere til et særligt mønster i valg af programmerings struktur.

### 7.1 Python

Python er et meget brugt gratis programmeringssprog, der skal installeres og åbnes som et version environment, et miljø der giver adgang til nemme programmerings funktionalitet. Det er et moderne web framework setup hvor Machine Learning biblioteker som numba og panda kan inkluderes. Den seneste version er Python 3.11<sup>1</sup> som kommer med en masse variabler til brug i statements, som det kan ses herunder.<sup>2</sup>

```
# Get data type
data_type = type(variable)

# Get user input
user_input = input("Enter a value: ")
```

---

<sup>1</sup> <https://python.org>

<sup>2</sup> <https://kinsta.com/blog/php-vs-python>

```

# Print output
print(result)

# Generate sequence of integers
sequence = list(range(1, 11))

# Get length of an array
element_count = len(array)

# While loop
x = 0
while x < 10:
    # Code block
    x += 1

# For loop
names = ['John', 'Jane', 'Bob']
for name in names:
    # Code block

# Get file path
file_path = os.path.realpath('path/to/file.txt')

```

Python Django udnytter pip installation af packages til at udvide funktionaliteterne, herunder one time password OTP authentication, database implementering herunder MySQL, Postgres og Software Quality Assurance bl.a. lint, pip audit. Python mest kendte frameworks er Flask og Django.

### 7.1.1 Flask

Flask ses som et minimalt Python framework<sup>3</sup>, der ikke bruger eksterne biblioteker under opsætning, men standard form validation, object-relational mapping og open authentication. Flask har også Unit testing system intelligence, som giver en væsentlig feedback under debugging. Der kan også indhentes extensions i et omfang som vil dække de fleste krav under webudvikling.

---

<sup>3</sup> <https://kinsta.com/blog/flask-vs-django>



## 7.1.2 Django

Django er et framework, dedikeret til hurtig opsætning af webløsninger. Django ses som et skalerbart framework, ved at kunne integreres med en lang række applikations pakker. Django tilbyder en simpel SQLite database, out of the box med et admin panel med database oversigt, samt en troværdig opsætning af gruppe permissioner, ud fra de apps der er tilgængelige. I Djangos settings.py kan databasen nemt ændre til f.eks. Postgresql eller InnoDB fra MySQL også XAMPP med PHPMyAdmin kan sættes op ved brug af eksterne packages. Med Django er sikkerheden sat højt og under udvikling sikre Django, at de mest sårbare programmeringsfejl, som SQL-injection og cross-site request forgery, ikke er muligt.

Samlet set kan det siges, at Django tilbyder et hurtigt framework med fokus på sikkerhed og scalability, mens Flask fremstår som et framework til mindre løsninger, som især kan bruges til one page webapplikationer. Se Bilag - PentestTools-SqlInjection-report.pdf

## 7.2 PHP

PHP Hypertext Preprocessor<sup>4</sup> er et andet udbredt objektorienteret open source programmeringssprog, som tilbyder et php biblioteket, der gør kodningen hurtig og nem. Den seneste version af PHP er 3.8, der kan bruges til de fleste større frameworks og har multiple database support herunder MySQL og Postgres. PHP er kendt for at koden starter med `<?php` og kan slutte med `?>`. Der bruges brugerdefinerede funktioner<sup>5</sup> til at mindske duplikering af kodning og super variabler i statements, som kan ses herunder<sup>6</sup>.

```
// Get data type
$dataType = gettype($variable);

// Get user input
$userInput = readline("Enter a value: ");

// Print output
echo $result;

// Generate sequence of integers
$sequence = range(1, 10);
```

---

<sup>4</sup> <https://kinsta.com/blog/php-vs-python>

<sup>5</sup> <https://www.geeksforgeeks.org/what-are-user-defined-functions-and-built-in-functions-in-p>

<sup>6</sup> <https://www.codeconvert.a>

```
// Get length of an array
$elementCount = count($array);

// While loop
$x = 0;
while ($x < 10) {
    // Code block
    $x++;
}

// For loop
$names = ['John', 'Jane', 'Bob'];
foreach ($names as $name) {
    // Code block
}

// Get file path
$filePath = realpath('path/to/file.txt');
```

I PHP bruges regulære expressions til tekstsøgning og PDO klasser php dataobjekter til “try” expression og “catch” data error funktionalitet og for at interagere med eksterne web servere ved hjælp af protokoller som IMAP, POP3 og HTTP. Npm-biblioteket<sup>7</sup> og at PHP er et at de tidlige programmeringssprog udviklet til framework er nogle af hovedårsagerne til at PHP står stærkt som e-handels løsninger, hvor PHP kan udnyttes maksimalt. Det er vigtigt at vurderes om systemet skal skaleres bredt eller dybt, hvis PHP skal vise sin virkelige styrke<sup>8</sup>. Management Systems er udviklet som en dyb løsning, som implementeres i enkelte virksomheder, gerne langvarigt med stor indvirkning fra stakeholders.

### 7.2.1 Lavarel

Lavarel er testet til at være et af de hurtigste php frameworks, der bruge de mest populære SQL databases og kun kræver en installation, for at komme i gang. Lavarel tilbyder sikkerhed på følgende områder: Bruger authorization, kryptering, Hashing. Derudover

---

<sup>7</sup> <https://www.npmjs.com>

<sup>8</sup> <https://kinsta.com/blog/php-frameworks/>

tilbyder Lavalab brug af Email confirmation og Forny password. Det lyder ikke så lidt fancy. Derudover tilbyder Pack Lyst mere end 15000 software stumper.

## 7.2.2 Symfony

Symfony er kendt for at have professionel support samt en fuldt dokumenteret e-læringsplatform. Symfony er skalerbar ved at understøtte de fleste SQL databaser. Det er også muligt at installere mere end 4000 Symfony packages, hvor seriøsiteten kan ses af det faktum at CMD platformen Drupel udnytter dem.

## 7.3 Dashboard

I de fleste webløsninger har i dag en eller anden form for bruger authorization, hvor det skal være muligt at tilføje indhold til sitet i eget brugernavn, et dashboard med forklaring af indhold og input felter, der kommunikerer med en opsat database. Et dashboard kan være et genereret layout i forbindelse med installationen af et framework designet til det valgte formål. Det er vigtigt at gøre sig klart, hvilke kravspecifikationer og funktionalitet der søges i forbindelse med løsningen af projektet.

## 8 Mobile Framework

Hvis der stilles krav om deployment på App Store eller Google Play vil der oftest tages brug af udvikling software som Xcode og Android Studio, her kommer Vue og React<sup>9</sup> ind i billedet. Vue arbejder med en visuel DOM, der duplikeres og præsenteres som en JavaScript konstruktion. Vue er også kendt for at have en todelt data binding, Event binding og property binding, som giver nogle særlige muligheder for at anvende værdier på HTML-elementer og klasser. Vue-objekter er komponenter, der indeholder genanvendelige elementer, som kan interagere med hinanden gennem events. VUE udnytter også den virtuelle DOM til at genskabe brugergrænsefladen baseret på den virtuelle DOM, hvilket giver en kortere vej til komponenter. Der sker en løbende forøgelse af plugins, som sammen med VUE CLI giver en kort indlæsnings tid.

React bruger en envejs data binding, som gør det muligt at rette en tilbagekaldes funktion efter en handling. React JS gør det muligt at udvikle HTML lignende byggeklodser nemt ved brug af J VX, et hurtigt framework med mange komponenter, som nemt kan udvides og forbedres med eksisterende kodning. React er skiftet til Hooks variabler som importeres og kaldes i function components, i stedet for tidligere class components.

---

<sup>9</sup> <https://kinsta.com/blog/vue-vs-react/>

## 9 CSS

Når der arbejdes med simple HTML som skal layouts enten som former, overskrift eller brødtekst, uden inline styling, er det muligt at bruge class less css styling som er en simpel grid styling af input former. For at opnå et responsivt layout kan Flexbox bruges til at style unordered list tags ved at tildele en column direction eller float left eller right. For at sikre en solid, responsive layout, skal der bruges @media screens(max-width:xxx px) og specifik styling af form felter til mobile web applikationer, for ensartethed med sammenlignelige applikationer.

## 10 Databaser

ACID<sup>10</sup> reglerne kommer i spil når vi udvikler backend. At få frameworket feedback med advarsler og error fra systemet til debugging er en af de vigtigste ting at overveje, når der skal overvejes en opsætning. Ved atomicity ses en handling som enestående, hvilket betyder, at tabellerne vedbliver intakte. Consistency sikrer, at databasen kun kan bevæge sig fra en konsistens, til en anden, herunder begrænsninger, cascades og triggere. Isolation bruges, når en transaktion udføres på flere tabeller på samme tid, og sikrer, at udførelsen er skjult, indtil den har fundet sted. Durability sikrer, at transaktionen bevares som en afsluttet transaktion. Holdbarhed opnås ved at lave en non-volatile registrering af transaktionen. Grunden til, at det er værd at nævne, er at der ved brug af python environment, er så stor sikkerhed for, at disse regler opretholdes, at systemet fremstår robust efter migration. SQL har den unikke mulighed for at kunne bygges med joins, hvilket på SQL niveau betyder at konteksten er etableret, før data tilgår applikationen.

### 10.1 Relational databaser

For at komme lidt længere med forståelsen af fordele og ulemper mellem forskellige SQL databaser<sup>11</sup>, er der læst artikler med sammenligning og her er, hvordan databaserne fremstår.

MySQL er en meget brugt open source database. Open source database gør det muligt at arbejde med data uden at dele dem med andre og kan derfor bruges, hvor sikkerhed er særlig vigtig. MySQL er kendt for at være brugervenlig, pålidelighed og rimelig hurtig. Udover at kunne håndtere store mængder data er den også den mest kompatible database, når det kommer til at flytte data og omstrukturere til andre systemer. MYSQL kan skaleres til

---

<sup>10</sup> Databases Things we do in the class 13-10-2022 af Santiago Donoso, Lecturer at KEA

<sup>11</sup> <https://kinsta.com>

1000-vis af forespørgsler i sekundet. Den overholder ACID-principperne og er sikker ved f.eks. at kunne indsætte callback-transmission når der sker kritiske transaktioner, og dermed tilgå et tidligere stadium, hvor der var consistency i datasættet. Derfor bruges MySQL ofte til e-handelsplatform, administration af kundedata, transaktioner og lagerstyring.

PostgreSQL er en objektrelation database open source med gode udvidelsesmuligheder og understøttelse af brugerdefinerede datatyper. PostgreSQL er en af de mest brugte databaser i dag, lige efter MySQL<sup>12</sup>. PostgreSQL kan bruges af forskellige brancher. Den er ACID-kompatibelt med sikker redundans og har næsten ingen nedetid. PostgreSQL er en relationel database, men kan også fungere som en dokumentdatabase. På grund af gode replikations muligheder kan hjemmesider nemt skaleres ud til et paraply af mange databaseservere.

SQLite er, som navnet antyder, let med hensyn til opsætning, administration og lagring, hvilket sker i projektet. De fleste databaser kræver en server proces, hvor SQLite er serverløs<sup>13</sup>, hvilket betyder, at applikationer kan læse og skrive data direkte uden klient-server-arkitektur. Derudover kræver SQLite ikke installation eller konfiguration, hvilket gør den selvstændig og mindre afhængig af operativsystemet. Den bruges i stor udstrækning på personlige maskiner som smartphones og scannere.

## 10.2 Dokument databaser

Dokumentdatabaser er ikke-relationelle databaser, f.eks. MongoDB, ArangoDB, Firebase, Firestore, SurrealDB. Dokumentdatabasen er lagret i en fil og har en anderledes struktur og retorik, hvert objekt kaldes et dokument, og et dokument indeholder flere columns key: value pairs. Der bruges forespørgsler, skemavalidering, sammenligning og referencedata. SurrealDB tillader brug af SQL.

## 10.3 Research af databaser

I forbindelse med udviklingen af Management Systems på en Django framework er følgende database muligheder gennemgået: PostgreSQL, MySQL, SQLite hvor test af benchmark<sup>14</sup> af insert, write, update, delete fra form, viser en stor fordel i brug af Postgresql. Et andet aspekt er opsætning og integration af funktioner og triggers, hvor MySQL har nogle klare fordele, som det ses i valgmuligheder af software applikationer udviklet til at understøtte MySQL,

---

<sup>12</sup> <https://kinsta.com/blog/postgresql-vs-mysql/>

<sup>13</sup> <https://kinsta.com/blog/sqlite-vs-mysql/>

<sup>14</sup> <https://github.com/sarathak/django-db-benchmark/tree/main>

herunder Workbench og PHPMyAdmin. Python tilbyder en enkel opsætning i settings.py i Django projektet med package fra PyPi<sup>15</sup> mysqlclient 2.2.4. MySQLClient gør det muligt at migrere Django's modeller til tablets og columns i databasen, som om det var SQLite eller PostgreSQL. Innodb er en grundlæggende MySQL cluster database, som nemt kan bruges til app udvikling fra CLI, alle MySQL commands kan inkluderes. Senere i produktionen vil det naturlige valg ved brug af MySQL være Mariadb databasen, som er en kommercielt understøttet gren af MySQL, som er et gratis open source-software under GNU General Public License. I udviklingsfasen af Management Systems er det valgt at arbejde med postgresql, fordi det er Django standard og fordi der er de samme muligheder for at eksportere databasen til MySQL. Export kan komme på tale hvis en mere komplekst opsætning af algoritmer og data miljø skal tages i brug, herunder kalkulerings af datasæt med IA modeller.

## 10.4 Entities Relation diagram

Entity Attributter er definerede datatyper som string, boolean og arrays. I ER Diagram viser primære og foreign keys, data types og relationer, one to one, one to many eller many-many, forholdet mellem tabellerne. Se Bilag - Database.pdf.

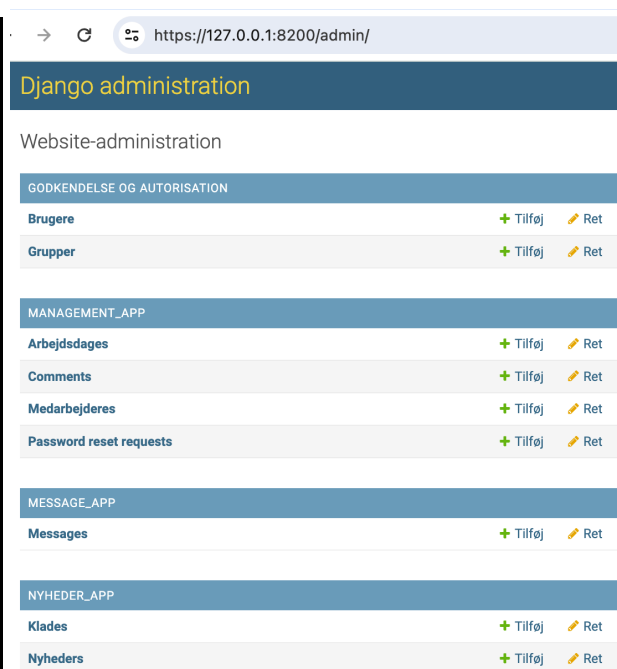
```
peter@kea-dev: ~/2024.04.10/management-systems
(main) % docker-compose exec db psql -U root -d db_management

psql (14.12)
Type "help" for help.

db_management=# \l
db_management=# e
db_management=# \l

      List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
db_management | root | UTF8 | en_US.utf8 | en_US.utf8 | 
postgres | root | UTF8 | en_US.utf8 | en_US.utf8 | 
template0 | root | UTF8 | en_US.utf8 | en_US.utf8 | =C/root;
template1 | root | UTF8 | en_US.utf8 | en_US.utf8 | =C/root;
(4 rows)

db_management=# \dt
      List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
public | auth_group | table | root
public | auth_group_permissions | table | root
public | auth_permission | table | root
public | auth_user | table | root
public | auth_user_groups | table | root
public | auth_user_user_permissions | table | root
public | authtoken_token | table | root
public | django_admin_log | table | root
public | django_content_type | table | root
public | django_migrations | table | root
public | django_session | table | root
public | management_app_arbejdsdage | table | root
public | management_app_buttonclick | table | root
public | management_app_comment | table | root
public | management_app_medarbejdere | table | root
public | management_app_passwordresetrequest | table | root
public | message_app_message | table | root
public | nyheder_app_klode | table | root
public | nyheder_app_nyheder | table | root
(19 rows)
```



### 2. Database tables og django admin panel

<sup>15</sup> <https://pypi.org>

## 11 Kravspecifikationer

Ved brug af kravspecifikationer<sup>16</sup> gennemgås alle dele af den applikation der skal udvikles. I et sheet oplistes alle krav baseret på handlinger på siderne, herunder muligheder, som at læse eller skrive, ændre data gennem request til serveren. Der kan være krav til teknologien, syn på interaktioner og oplevelsen af løsningen. Når de mest åbenlyse handlinger er på plads, kan der tilføjes detaljer, som kan indgå i teknologien ud fra et UX-syn på brugeren og stakeholders.

For at opbygge en stærk forståelse af produktet skal mulighederne i kravspecifikationen opdele i funktionelle krav f.eks. en knap der trigger en event og non-funktionelle, at handlingen skal eksekveres på f.eks. under en vis tid. Når funktionaliteten er på plads, skal der gives en prioritering fra 1 til 4 baseret på, om elementet skal indgå i projektet. Moscow-prioriteringen opdeler elementerne i Must have, should have, could have eller won't have. Se Bilag - Kravspecifikationer.pdf.

## 12 Business Model Canvas

Der er udarbejdet et Business Model Canvas, som gennem syn på samarbejdspartnere, aktivitet og ressourcer på den ene side og kundesegment, kundeforhold og kanaler på den anden side kommer frem til et fundament af værditilbud, omkostninger og indtægter. Selve Web applikationen har stor betydning i værdiansættelsen sammen med integration og unikke udviklingsmuligheder og firma hensyn. I første omgang vil der være betydelige omkostninger til udvikling, rådgivning, ansatte, samt den digitale infrastruktur. På indtægtssiden er det primært salg af abonnementer og særlige tjenester. Se Bilag - Business Model Canvas.pdf.

### 12.1 Stakeholders

Når et koncept som Management Systems skal udvikles, er det nødvendigt at inddrage ekstern finansiering. Det kan ske ved at gengælde procenter af senere indtægter eller tilbagebetaling med tillæg af renter. Rådgivning og support vil også blive inkluderet, ligesom direkte involvering af betalende teams, mindre og mellemstore virksomheder.

---

<sup>16</sup> GDD 3 Requirements Definition.pdf af Arturo Mora Rioja Lecturer at KEA

## 12.2 User Stories

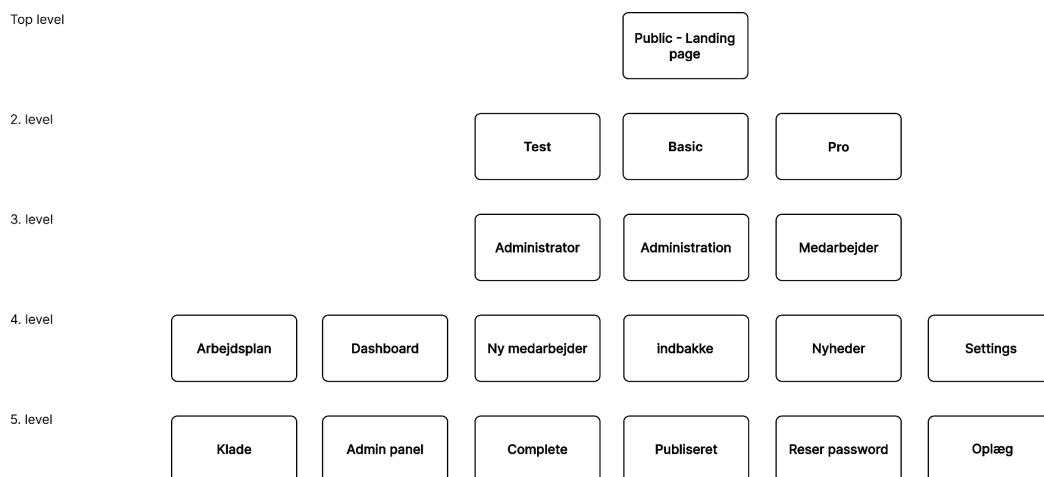
Til markedsføringsformål og for bedre at kunne forstå brugen af applikationen er der gennemtænkt to User Stories. De kan findes som et kort indlæg midt på Landing page.

## 12.3 Virksomhedstype

Som udgangspunkt har der kun været to virksomhedsformer at vælge imellem, EVS enmandsvirksomhed og ApS Anpartsselskabet, det skyldes at der kræves flere end en stiftere for at starte en interesse selskab og at der kræves en væsentlig startkapital ved stiftelse af et A/S Aktieselskab. Det er valgt at stifte et anpartsselskab med en startkapital på 40.000,- kr og med en ejer<sup>17</sup>. Dette valg kræver stor commitment fra start af, hvad regnskab angår og markedsføring. Markedsføringen er omfattende og vil kræve løbende kontakt til testgrupper. Som del af udviklingen af virksomheden skal der ansættes kvalificerede medarbejdere, som under ansættelsen kan købe sig ind som medejer i selskabet, hvis der skabes en særlig god dynamik fra medarbejderen.

## 13 Sitemap

En oversigt over sider og hvilket af de fem levels i brugergrænsefladen, de befinder sig på.



### 3. Sitemap af Management Systems

## 14 Wireframes

Som en vigtig strukturel analyse af systemet er der udarbejdet et wireframe, der afspejler projektets funktionelle krav, der stilles til projektet. Figma tilbyder et teambaseret miljø til at

<sup>17</sup> <https://www.nemadvokat.dk/blog/selskabsstiftelse/de-4-virksomhedstyper>



visualisere komplekse webløsninger med mulighed for at præsentere direkte fra den sky hvor det blev udarbejdet.

Fordelene ved at inkludere et wireframe af løsningen er, at det giver et særdeles godt overblik over hvad løsningen indeholder, både til at forstå funktionaliteter, design og senere i den videre udvikling og i test fasen. Se Bilag - Wireframes.pdf.

## 15 Scenarios

Der er valgt fem Key-Path scenarios som der udvikles på.

Functional requirements i grupper og hierarchically

### Authentication

#### Login

- Fra landing page click på login i hovedmenuen, du ser -----**Key-path scenario**  
login form Necessary-use scenario
- Er det første gang der logges ind gå til Nyt password -----**Key-path scenario**
- Ellers skriv dit brugernavn og password og click login Necessary-use scenario

#### Forny password

- For at forny password angiv brugernavn og Email, derefter nyt password to gange Necessary-use scenario

#### Logout

- for at logout gå til settings i hovedmenuen og tryk på logout Necessary-use scenario

### Medarbejder

#### Arbejdsplan

- Click Arbejdsplan og se dine arbejdsdage og opgaver Necessary-use scenario
- Skriv til administrationen på valgt arbejdsdag -----**Key-path scenario**
- Click på Start tid og efter opgaven er løst Stop tid Alternative-use scenario

#### indbakke

- Skriv en besked til en anden medarbejder Alternative-use scenario
- Modtag en besked fra en anden medarbejder Alternative-use scenario
- Åben valgt besked og læs indhold Alternative-use scenario

#### Nyheder

- I Nyheder kan du læse seneste nyheder fra ledelsen Alternative-use scenario

#### Settings

- I Settings har du tre valg Nyt password, Theme, Logout Necessary-use scenario
- I Settings har du tre valg Nyt password, Theme, Logout Alternative-use scenario
- Nyt password kræver et request med brugernavn og Email Alternative-use scenario

- |                                                   |                          |
|---------------------------------------------------|--------------------------|
| - Logout redirecter til Login page                | Necessary-use scenario   |
| - Theme skifter toggle mellem mørkt og lyst theme | Alternative-use scenario |

## Administration udover Medarbejder

### Dashboard medarbejderliste

- |                                                                   |                          |
|-------------------------------------------------------------------|--------------------------|
|                                                                   | Necessary-use scenario   |
| - Søg efter medarbejder i index søgning                           | Alternative-use scenario |
| - Click på den valgte medarbejder og se Profil og arbejdsdage     | Alternative-use scenario |
| - Profil oplyser alt om medarbejderen og oplysninger kan fornyes  | Alternative-use scenario |
| Arbejdsdage er en liste med dage, opgaver, beskeder, skriv        |                          |
| - besked, start tid, slut tid og brugt tid                        | Alternative-use scenario |
| - I skriv besked kan du skrive en besked på en valgt dags opgaver | Alternative-use scenario |
| - Tilføj arbejdsdag med mødetidspunkt opgaver                     | Alternative-use scenario |

### Ny medarbejder

- |                                                                                                     |                          |
|-----------------------------------------------------------------------------------------------------|--------------------------|
| - Opret medarbejder med brugernavn, for og efternavn, password, Email, telefon, personligt id ----- | <b>Key-path scenario</b> |
| - Efter medarbejder er oprettet, er der mulighed for at oprette arbejdsdage -----                   | <b>Key-path scenario</b> |

### Nyheder

- |                                                          |                          |
|----------------------------------------------------------|--------------------------|
| - I nyheder er der tre valg Klade, Nyheder, Published    | Alternative-use scenario |
| - I Klade kan du skrive en klade i et fælles forum       | Alternative-use scenario |
| - I Nyheder kan der skrives et indlæg til vurdering      | Alternative-use scenario |
| - I Published kan du se hvilke nyheder som er publiceret | Alternative-use scenario |

## Administrator udover Medarbejder og Administration

### Nyheder

- |                                                      |                          |
|------------------------------------------------------|--------------------------|
| - I Klade kan du prioritere klader i et fælles forum | Alternative-use scenario |
| - I nyheder kan der publiceres indlæg                | Alternative-use scenario |

### Settings

- |                                             |                          |
|---------------------------------------------|--------------------------|
| - I settings er der adgang til admin panel  | Alternative-use scenario |
| - I Admin panel kan du redigere i alle data | Alternative-use scenario |

## Landing page

### Landing page

- |                                                                                   |                          |
|-----------------------------------------------------------------------------------|--------------------------|
| - Det er fire valg Home, Kontakt, FAQ, Login                                      | Necessary-use scenario   |
| - På Home siden kan du se hvem Management Systems er                              | Alternative-use scenario |
| - På Kontakt kan du sende en besked til administrator med navn, Email             | Alternative-use scenario |
| - På FAQ kan du index søge og slå op i fire kategorier af spørgsmål til konceptet | Alternative-use scenario |
| - På Login går du til login siden til Management Systems                          | Alternative-use scenario |

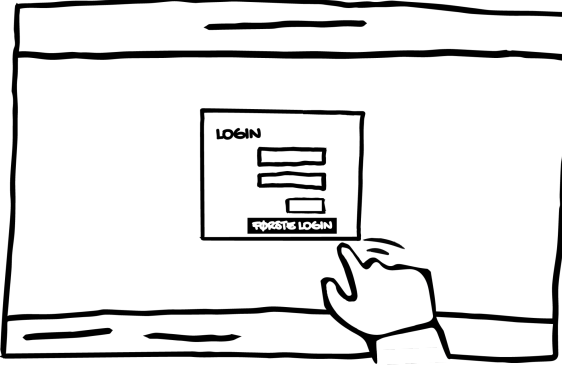
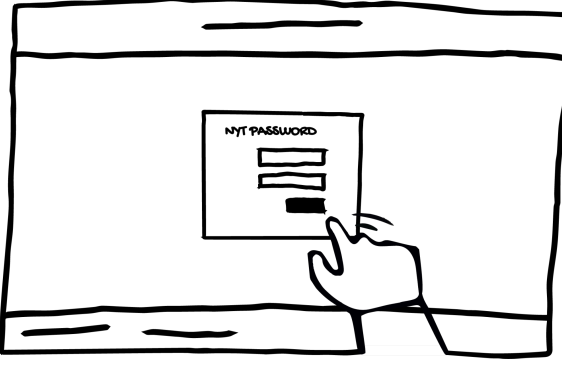
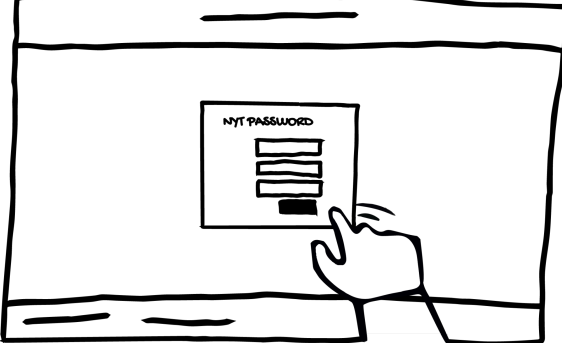
Key path scenarios er ofte et hvordan spørgsmål, der forklares tekstuel i form af storyboards. Storyboardet kan forklares ved brug af en række illustrerende billeder. Under udviklingen af wireframes blev der stillet to spørgsmål til Key Path Scenarios.

En medarbejder ønsker indsigt i næste dags arbejdsopgaver og sender en besked til lederen?

En ansat i administrationen vil tilføje en ny medarbejder og de første arbejdsdage?

## 15.1 Key-path scenarios storyboards

Fra første login til at se arbejdsopgaver til at sende en besked til Lederen

|                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Login</b></p>  <p>På første arbejdsdag skal Eva logge ind, ser hun at der står skrevet på login siden at hun skal ændre sit password første gang hun logger ind, ved at trykke på Første "Login".</p> | <p><b>Nyt Password</b></p>  <p>Efter at have trykket på "Første Login" kommer hun ind på Nyt Password siden, hvor hun skal angive brugernavn og Email for at få mulighed for at ændre password. Eve trykker på "Submit" og sender hendes oplysninger.</p> |
| <p><b>Nyt Password</b></p>                                                                                                                                                                                 | <p><b>Arbejdsopgave</b></p>                                                                                                                                                                                                                              |

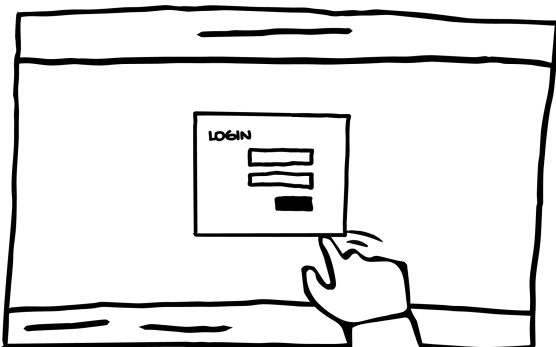
En ny side med tomme input felter venter på at hun skriver brugernavn og sit nye password to gange.

Efter Eva har brugt sit nye password til at logge ind, kan hun se sit navn og en liste med arbejdsdage, opgaver, tomt, text input. Hun vælger at skrive en besked til administrationen på beskeder til første dag. Hun skriver at hun fornyede sit password og er klar til at starte på første opgave.

#### 4. Storyboard - Key-path scenarios

Fra Login til at oprette en medarbejder og arbejdsdage

##### Login

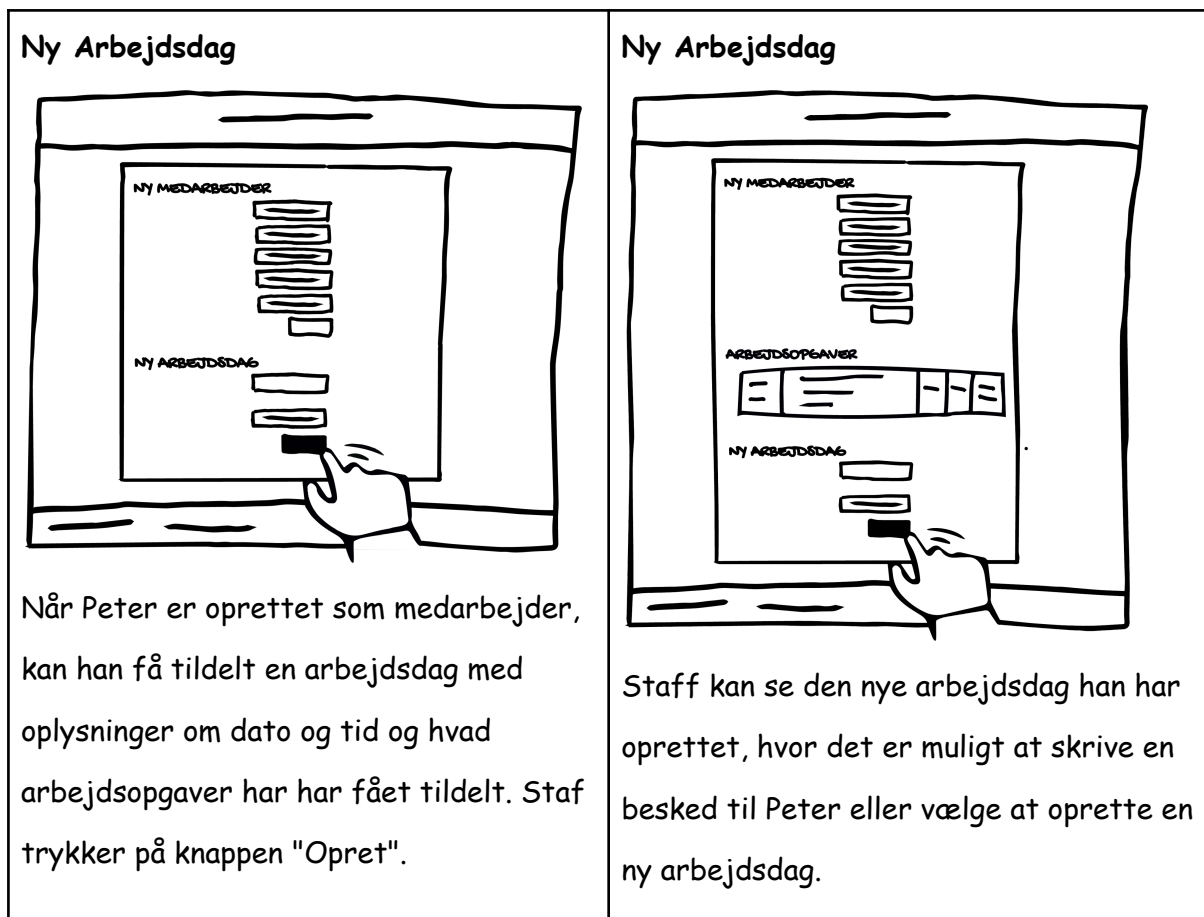


Staf er ansat i administrationen og har fået besked om at Peter er blevet ansat. Staf skal oprette Peter som medarbejder og tildele ham de første arbejdsdage i kalenderen. Staff logger ind som administrations ansat.

##### Ny Medarbejder



Efter at være logget ind, lander han på siden Ny Medarbejder. Staf ser en liste med input felter som skal udfyldes før Peter kan få tildelt sit brugernavn og midlertidige password.



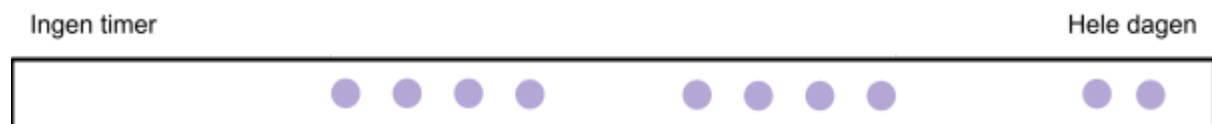
5. Storyboard - Key-path scenarios

## 16 Adfærds variable

Management Systems adfærds variable

- Hvordan bruges tiden ved skærmen

read



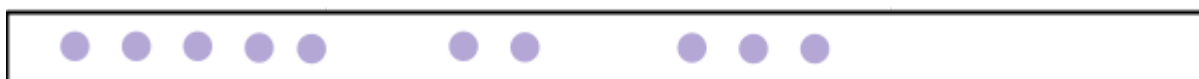
search



write

Ingen tid

Hele dagen



## 6. Adfærds variables, UX opgave KEA 2023

På grund af manglende brugerdata og tid til at generere seriøse analyser ved at opsætte Machine Learning modeller, er det valgt at vise hvordan indsamlede brugerdata kan se ud. Det sker gennem interview, kvalitative data eller brugerundersøgelser, kvantitative data, for at få en dybere forståelse af, hvordan brugeren prioriterer deres tid. Se Bilag - Brugerundersøgelse.pdf.

I fremtiden vil det blive testet på produktet for at sikre analyse af krav og mangler, bl.a. ved at bruge scenarier fra manuskript. Testpersoner vil kunne kommentere på spørgsmål. Disse data vil give et gennemsnitligt syn på mangler og vil kunne hjælpe med at komme med løsninger til forbedringer.

## 17 Persona

Det er på baggrund af indhentede indsigter udarbejdet tre persona'er<sup>18</sup>, to mænd, en kvinde, kvinden er vist herunder. De har alle det fællestræk, at de har deres daglige gang på kontoret. Se Bilag - Persona.pdf.

|                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Navn:</b> Eva Johnesen, Hovedvejen 40, 3480 Fredensborg                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                                                                                     | <b>Hverdag</b><br>Jeg arbejder for Alles Tag Aps. Alles Tag tilbyder taglægning på tage med tegl, samt rådgivning og gennemførelse som byggeleder af mellemstore tag projekter. Mine arbejdsopgaver består i at tilbyde personlig rådgivning til kunderne. Når jeg ikke sidder i telefonen med kunder er jeg på farten for at mødes med en kunde. Jeg har gjort det i 6 år og er oplært på dette område som var det i mesterlære. Jeg har altid deltaget, hvis der har været sociale sammenkomster og mener det er væsentligt at vedblive sine familiebånd. Jeg arbejder ofte med ansvar mellem det aftalte og udførelsen, opgaver på flere planer, det er et stort ansvar. |
|                                                                                     | <b>Mål</b><br>Arbejde for min familie og give Alles Tag en mulighed for at udvikle størrelsen til et niveau, hvor det er nemmere at opfylde løbende arbejder, uden at skulle trække på eksterne håndværkere.<br>Et andet mål er at bruge mest muligt tid på min søn Aske, som er i en alder, hvor han formes meget og derfor er meget modtagelig og skal udvikle sig mentalt med gode værdier.                                                                                                                                                                                                                                                                              |
|                                                                                     | <b>Paint points</b><br>Hvordan begrænser jeg den tid, jeg er på arbejdspladsen, og får mere tid hjemme? Jeg føler at mit arbejde fylder meget, også når jeg har fri?<br>Mit engagement i kultur og samfund, som enlig, har jeg haft svært ved at forfølge.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>KØN:</b><br>Kvinde                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>ALDER:</b><br>37 år gammel                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>SAMLEVERSKER:</b><br>Ingen                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>BØRN:</b><br>Eske, 14 år gammel                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Citat:</b><br>Jeg arbejder gerne, hvis andre ikke gør                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

## 7. 1 af 3 Personas

<sup>18</sup> GDD 2 Modelling.pdf af Arturo Mora Rioja

## 18 UX og Design regler

Når der skal udvikles et design til brug af mindre til mellemstore virksomheder, er det værd at overveje, at der vil forekomme en bred vifte af farvevalg og layout, som er en del af det enkelte firmas brand. Derfor udtrykker designet fra Landing page, en forståelse af designprincipper og skal gerne vedblive at eksistere hos stakeholders, det sker ved b.la. at bruge Micro interaktioner som htmx.org, npm/sortable.js, herunder feedback med Swal og Weetalert.

Selve Management Systems er holdt i et enkelt design, der er let at forholde sig til og kan bruges som en gratis test løsning i udviklingsfasen og senere test før køb. Der var mulighed for at integrere et bootstrap design med javascript og css bibliotek, eller et simpelt flexbox setup som gennem klasser ville kunne fremstå simpelt og responsivt. Begge løsninger af UX indeholder eksterne biblioteker og skal tilføjes klasser, som kan besværliggøre ændringen af designet til det endelige valg. Hvorfor ikke et simpelt css classless<sup>19</sup> setup i en intern fil, der opfylder de UX krav som webløsningen skal tilbyde. Simpel css og holliday er to små filer som udnytter grid opsætning af form felter og samtidig er responsivt. Holiday.css har en dropdown-styling af knapper i navigationen og gør web applikationen mobil venlig.

### 18.1 Farve og font

Farverne og font typerne er valgt ud fra hvordan Landing page er designet, skrifttypen er Nunito og farven er grå. Det valgte udtryk afspejler forsigtighed og familieliv og er et prototype design, der nemt kan evalueres på. Landing page har samme farveskema, men bruger Call To Action og branding til at skabe værdi for kunden.

### 18.2 Ikoner

Ikoner vælges ud fra principperne om, at de skaber genkendelighed i forhold til andre webløsninger og øger overblikket. Her kommer favicons til sin ret og gør det nemt at finde applikationen imellem andre browser tabs. Det er også muligt at udvikle layout ud fra principper om at brugeren opnår ekspertviden, som kan ses som genveje i layoutet og medvirke til at mindske stress.

## 19 Udviklingsstrategi

Som udgangspunkt for denne opgave, at udvikle et Management Systems, der skal markedsføres over for startups, små og mellemstore virksomheder, skal der lægges en

---

<sup>19</sup> <https://github.com/dbohdan/classless-css>

udviklingsstrategi, der afspejler krav til funktionalitet. Udgangspunktet er, at tilbyde en brugbar prototype, som stakeholders kan forholde sig til så hurtigt som muligt. Prototypen skal fremstå som et konkurrencedygtig produkt på markedet. Det vil kræve udvikling i et team, hvor særlige kompetencer kan komme på tale. De skal være fortrolige med aspekter som Software Quality Assurance, Version Control, Containerisation og sikkerhed især for databasen. Der er ligeledes udfordringer i markedsføring, content management og branding.

## 19.1 Delmål

For at komme så langt som muligt på kort tid er delmål planlagt, så der ikke forekomme væsentlige iterationer mellem punkterne i processen. De delmål som har været i spil frem til levering, er i kronologisk rækkefølge

- Planlægning, hvilke værktøjer der skal bruges til hvilke formål, opsætning og struktur.
- Målsætning, en introduktion til projektet og de underliggende problemer og muligheder.
- Research, der dokumenterer, at markedet er klart til Management Systems, herunder Business modeling og UX design elementer fra User centered og goal directed design.
- Development, gennem udvikling af Wireframes, prototype, backend og frontend samt Continuous integration og deployment.
- Funktionel prototype til test og fundraising.
- Evaluering, er målet med projektet nået, kunne tingene være gjort bedre og selve konklusionen.

## 19.2 Endeligt mål

Det endelige mål med projektet er at udvikle et nutidig digitalt fundament, der kan stå intakt uden store ændringer det næste årti. Der er udviklet en webløsning, som kan tilbydes kunden, og som er fuld funktionsdygtig, også på mobile enheder. Derudover er der opbygning af et dygtigt team af medarbejdere. Forretningsformen er et anpartsselskab, hvor stakeholders kan skabe vækst og inspirere til samarbejde med nye ansatte. I næste fase skal anpartsselskabet tilbyde digitale løsninger, der kan skabe overskud for stakeholders og for Management Systems.



## 20 Valg af Software

Der er et bredt udbud af software at tage stilling til. Hvor kan der spares tid under udviklingen af konceptet, webapplikationen. Der er brugt Trello, Google Drev, Figma, Smartdraw, VS Code og Vim.

No-code software, udviklet til at sætte fart på programmering og færdiggørelse af et sammenhængende produkt. Værktøjer som Grid til prototyping, FlutterFlow<sup>20</sup> og Webflow, som tilbyder de samme moduler, rest API og DNS integration. FlutterFlow som er et framework fra Google, udviklet med Dart programmering hvor parent, child af body tag særligt layoutet som app deployment på cross platform, App store og Google play. Webflow har næsten den samme brugergrænseflade, men er baseret på et HTML template, hvor enkle API curries og DNS submission er tilgængeligt out of the box. Grid er et Google sheet produkt til simple udvikling af prototype App løsninger, med download fra QR kode.

I dette projekt er Landing page designet og udviklet ved brug af et gratis Webflow theme template Velocity<sup>21</sup>. Velocity udnytter micro-animationer og interactions til det yderste. FlutterFlow vil være oplagt til at udvikle en mobil løsning til medarbejdere, som del af deres personlige mobil installation eller en tildelt mobil arbejdsenhed, der kan indeholde flere nødvendige applikationer til brug på jobbet.

### 20.1 Planlægningsværktøjer

Planlægningen kræver et layout, der er overskueligt og nemt at kommunikere og dele dokumenter i. Der er et stort udvalg, og flere fungerer gratis, herunder Asana og Trello. Trello blev valgt til denne opgave, fordi det er nemt at overskue, opgaverne er ensartede objekter. Hvorimod Asana har mange tilføjelser herunder diagrammer, tegneværktøjer og analytics. I Trello indsættes de enkelte ugentlige sprints i tags Todo, Done, Problemer eller til Rapportskrivning, hvor hver enkelt tag har en farver indikation. Gennem sorterings muligheder kan der ændres løbende og prioriteres. Sker der udskydelse af opgaver eller genoptages tidligere opgaver, er det nemt at anskueliggøre. Se Bilag - Trello.pdf.

### 20.2 EDI

Ved udvikling af et framework, skal der tages højde for hvilke EDI der er bedst til formålet. Apples Xcode og Android Studio er nødvendige for at kunne udvikle til Mobile enheder og Visual Studio EDI bruges til at udvikle ASP .NET Core frameworks, derudover er der det

---

<sup>20</sup> <https://webflow.com/> <https://flutterflow.com>

<sup>21</sup> <https://webflow.com/templates/html/velocity-multipurpose-website-template>

velkendte VS Code, som har en bred vifte af tilføjelser, dog ikke custom SwiftUI popcorn intelligence. I dette projekt var det oplagt at udvikle i VS Code med tilføjelser som programmering formater, ci pipeline til Gitlab, CLI terminal og Vim terminal EDI.

## 20.3 CI/CD

I dette projekt er der opsat en direkte ssh terminaladgang til VPS Alpine Linux drev hos Linode, der ikke er et custom operativsystem, men et installeret setup. Alpine er med til at gøre det nemt og overskueligt at ændre filer, permissions og placering. CI, Continuous Integrations med såkaldt build af layers, i et image gennem brug af Docker, Nginx server Containerization ved brug af Runners på Gitlab.

Software Quality Assurance sker automatisk på Gitlab efter push af commits, som er hashede (SHA-1) for dataintegritet. I forbindelse med selve commit opsættes gitignore og dockerignore for ikke at pushe unødvendige filer.

## 20.4 AI & ChatGPT

Der er nogle oplagte muligheder for at udnytte AI teknologi ved brug af machine learning. I et management system kan der ved brug af tags indikere hvilke type opgaver der udvikling på. Disse data kan optimere arbejdsprocessen ved at blive gennemanalyseret i et netværk, som sammensætter opgaver efter prioritering, tid, profil af udvikleren og data fra tidligere opgaver.

ChatGPT er et færdigt udviklet NLP tekst-generering<sup>22</sup>, som kan bruges til at gennemlæse kompliceret tekst og kodning og ud fra tidligere tekst, generere struktureret forståelse af teksten indhold og hvor der er fejl i f.eks kode gennem regel forståelse, herunder funktions udvikling af scripts, samt generere test af disse scripts, samt videreudvikling og forslag til opsætning i digitale frameworks, er der en rimelig hjælp at komme efter.

## 21 Framework

Som tidligere beskrevet blev Python framework Django brugt til Management Systems, primært fordi det er et nemt framework at starte og nemt at udvikle på.

Django frameworket kræver nogle få trin for at kunne starte op, først skal der installeres et Python environment fra terminalen og det er fra det drev, projektet skal startes. Python Environment gemmer PyPi installationer som importeres til projektet. Når pip har installeret

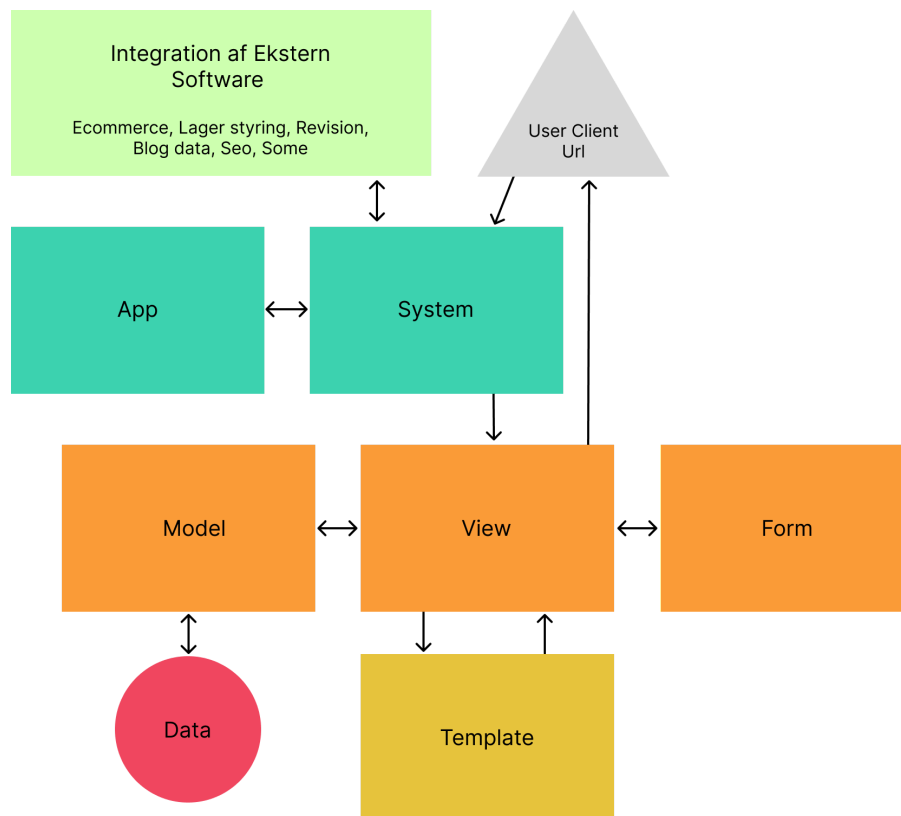
---

<sup>22</sup> AAI lectures af Henrik Strøm Lecturer at KEA

django i Python-enviromentet<sup>23</sup> og Django>=3.2,<5.0.6, kan der startes et projekt fra terminalen med (py-env)% django-admin startproject <projekt navn>.

Nu har vi en projektmappe, der indeholder en systemmappe og en manage.py fil, der udføre administrative opgaver, herunder runserver, management filer, tests og oprettelse af superuser, samt nogle importerede biblioteker. Herefter mangler kun at freeze pip installationer til en fil og udføre python manage.py startapp <app navn><sup>24</sup>, så har vi et framework, hvor vi kan udvikle dynamiske web løsninger med urls pach() routing, models, forms, views og templates.

## 21.1 Fil struktur



### 8. Django Projektstruktur

Ser vi på applikationens filstrukturen, som med fordel kan opdeles i backend models.py database tables, som migreres ved at angive en class Xxx(models,Model) med de respektive columns med angivet datatype, genereres der en migration fil som er med til at skabe forbindelse til databasen gennem DATABASES i settings. I forms.py kan der skrives

<sup>23</sup> Development environments lectures af Henrik Strøm Lecturer at KEA

<sup>24</sup> <https://pypi.org/>  
<https://kinsta.com/blog/php-vs-python/>

class Xxx (forms.ModelForm) med model, fields og labels. I views.py har vi def xxx(request): som gennem if, try, else, except, binder data sammen til et responsivt frontend sammen med form indføre der data fra models backend.

## 21.2 Engine

I settings opsættes DATABASE engine, som kan læse data fra standard SQLite database, derudover Postgresql, MariaDB, MySQL og InnoDB. At skifte SQL database er normalt ikke noget problem når opsætningen ellers fungerer, men hvis der skiftes til en dokumentdatabase kræves der tredjepart udviklet packages, der ikke med sikkerhed er vedligeholdt, herunder MongoDB, som PyPi pymongo package med engine djongo i settings. Også opsætning af Multiple Databases Instances er muligt<sup>25</sup>.

I dette projekt var der ingen grund til at ændre Django fra en standard engine set, da der ikke før senere i udviklingsfasen skal indgå komplicerede SQL queries til at strukturere en cloud solution med Multiple Access Key. Til at starte med out of the box SQLite, blev på et tidspunkt ændret til Postgresql, da der skulle indgå større sikkerhed og skalering. InnoDB MySQL blev også testet, men da Postgres har samme mulighed for at eksportere data til MariaDB og det kræver særlig opsætning på VPS, blev det ikke valgt til.

## 21.3 Billeder

Billeder i nyheder hentes med denne views.py def nyheder

```
16         image = request.FILES.get("image")
```

Ved opdatering triggers en elif method, af objektet ved brug af dens unikke pk, primary key.

```
28         elif request.method == "PUT":
29             pk = request.PUT.get("pk")
```

ved at importere fra django.views.decorators.http require\_http\_methods.

Billeder gemmes i mappen images fra models.py klassen Nyheder

```
10         image = models.ImageField(upload_to='images', blank=True, null=True)
```

som får en path gennem URLs debug i settings.py MEDIA\_URL og MEDIA\_ROOT = os.path.join giver en url som gemmes i databasen.

---

<sup>25</sup> <https://docs.djangoproject.com/en/5.0/ref/databases>

## 21.4 Pagination

Der er indsat Pagination paginator script på medarbejder list view, som er en Python super klasse<sup>26</sup>, der indsætter sider med et forudbestemt antal objekter fra en liste. Html siden indeholder to if statements `page.has_previous` og `page.has_next`, samt `{{page.number}}` of.

```
244 @login_required
245 def medarbejder_list_view(request):
246
247     medarbejder_list = Medarbejdere.objects.all()
248     items_per_page = 3
249     paginator = Paginator(medarbejder_list, items_per_page)
250     page_number = request.GET.get('page', 1)
```

ved at importere fra `django.core.paginator` `Paginator`, `EmptyPage`, `PageNotAnInteger`.

## 21.5 Rest API og json fetch

Representational state transfer handler om at tillade forskellige systemer at kommunikere og udveksle data via HTTP metoder. Vi sender en request til vores API og får et respons tilbage. Gennem Django Rest Framework<sup>27</sup> kan vi teste API endpoints. Django custom applikation, sikre et godt testmiljø, med adgang til form GET POST PUT DELETE og PATCH i html form eller raw data. Med brug af serializers, kan kompliseret data ændres til forståeligt Python datatyper, som den returnerer i json format eller andet, herunder XML. Det minder om forms filer i projektet med model, fields, labels og widgets input. Her udtrækkes datatyper til specifikke formål.

I management Systems foretages nogle af de samme valg, men med en middleware opsætning, hvor datatyper er specificeret i models og middleware opretter et PUT request, hvis der ikke er tale om et json object, hvilket betyder, at den eksisterende post data eller fil delete, request POST metoden for at saven den, hvor den var, en PUT request metoden opdaterer et objektet og bevare det intakt i databasen.

På Landing page er der en javascript index søgning af FAQ og show FAQ, som minder om rest API, men som er en lokal fetch json data. Derfor `.then if no error, return response.json();`, `.then` tilføj data object variable til root div og `.catch(err)`

```
53 .catch(err => {
54     const errorMessage = document.createElement("marquee");
55     errorMessage.textContent = "Arg.. Json is not working!";
56     document.getElementById("rootShow").appendChild(errorMessage);
57 });
```

<sup>26</sup> <https://docs.djangoproject.com/en/5.0/topics/pagination>

<sup>27</sup> <https://www.django-rest-framework.org/>

med en “marquee” error text message. Er data fra en MySQL server endpoint API cont variable og med return respocse.json();, næsten ens, men er et Rest API call til et remote endpoint. Se Bilag - Faqlssues.pdf

## 21.6 Bruger authorization

Bruger authorization er en vigtig del af Django standard backend. Som standard tilbyder Django en tredelt opdeling af brugeren under oprettelse. Disse roller er boolean columns i database tabellen, Super User, Staff og andet. Out of the box tilbydes permission, som er opdelt i superuser, som har permission til alt, staff har permission til admin panel dog uden se, skriverettigheder, andre brugere har kun client browser rettigheder. Permission er vigtige, når det drejer sig om udvikling af applikationen i teams, hvor teamlederen kan sikre at teammedlemmer kun har adgang til mindst muligt, af frameworket og af databasen. I Management Systems bruges der ikke Permissions til at starte med, da der kun er et team medlem. Der er skabt en deling kaldet binding i navigationen og andre steder, til de tre forskellige roller. Derudover bruges authorization til at logge brugere ind og til at sikre adgang til objekters id'er med @login\_required.

## 21.7 Bruger autentifikation

Når det gælder bruger autentifikation, er der normalt tre måder at sikre, at den bruger som logger ind, har samme identitet som forventet. Biometrics og token nøgler<sup>28</sup> ses ikke så ofte, det mest almindelige er password baseret login, som er den metode der bruges på Management Systems, her tildeles medarbejderen et brugernavn og et password, som de opfordres til at ændre, før første login. En anden mulighed er 2 factor authentication, som udnytter flere af brugerens enheder til at sikre identiteten optimalt. 2 faktor autentifikation vil nøje blive overvejet i den videre udvikling fase. MitID er et kendt og tillidsvækkende 2 factor system som Staten har udviklet i samarbejde med Signicat<sup>29</sup>.

## 21.8 SSL Sikkerhed

SSL Secure Certifikat Layer er et certifikat baseret key certificate, der ligger på servere. Et gyldigt certifikat giver en krypteret forbindelse mellem klient og server. SSL<sup>30</sup> er et sikkerhedskrav til e-handel for at sikre, at transaktionen og kundens kontooplysninger forbliver private. I dag ses SSL HTTPS som standard i alle dynamiske webløsninger. Under udviklingsfasen kan det være vigtigt at have adgang til SSL, da der ofte stilles krav til SSL

---

<sup>28</sup> <https://www.miniorange.com/blog/different-types-of-authentication-methods-for-security>

<sup>29</sup> <https://dashboard.signicat.com>

<sup>30</sup> <https://github.com/teddziuba/django-sslserver>

ved cross platform integration, herunder Facebook, Ecommerce. Da Management Systems kun er i udviklingsfasen, er SSL certifikat, kun prioriteret til den videre udvikling lokalt.

## 21.9 Routing

For at forstå, hvordan routing fungerer i Django-frameworket, kan den opdeles i flere faser<sup>31</sup>, først er der ROOT\_URLCONF indstillinger, der leder http request objectet mod root url, hvis der ikke forinden er fastsat en attribut, sat af middleware. Herefter router App'en requested til urls.py variabelen

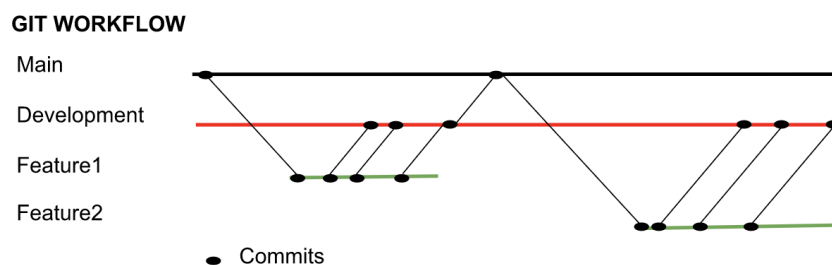
```
6 urlpatterns = [  
7     path('', views.index, name='index'),
```

som udvælges ved første URL match, herefter kalder django et views argument som er en klassebaseret funktion. Opstår der fejl i processen, tilgås en fejlhåndtering med intelligent feedback.

## 22 Software quality assurance

Der dukker stadig flere og flere værktøjer op, som sikrer at kodenstandarder overholdes og at packages og dependencies er opdateret. I dette Django setup er der brugt djlint til django test, pylama til Python test, ymallint til test af yml filer, shellcheck som tester alt shell kode. Pip-audit tjekker packages for nyere versioner og om der er konflikter mellem bibliotekerne. Der er ignore mulighed for de enkelte værktøjers error kode, som undlader fejlfinding på netop det punkt.

## 23 Version control on Gitlab



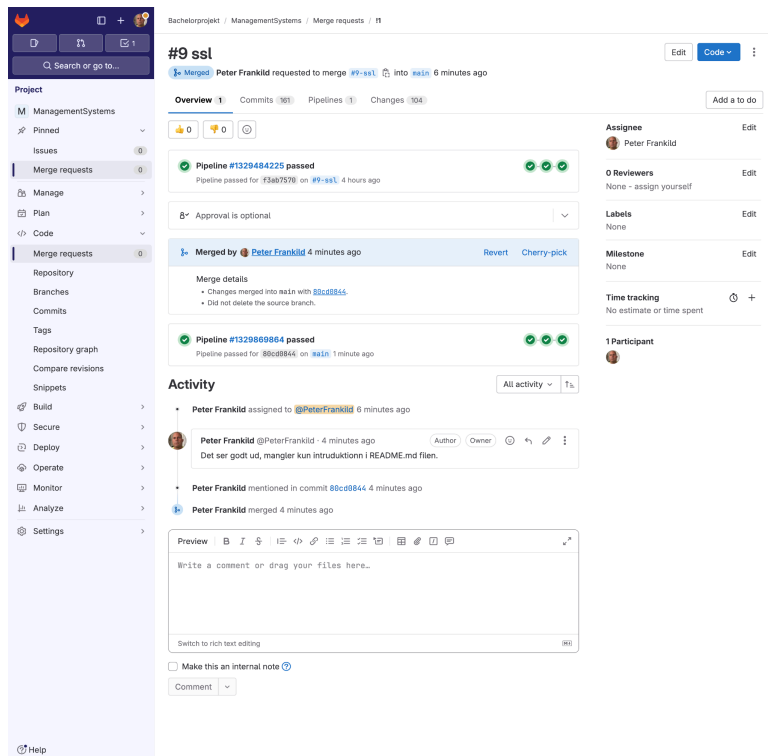
### 9. Git software quality assurance

Kort forklaret er det der sker at de løbende ændringer bliver gemt i blobs med unikt timestamps som altid peger bagud til et tidligere gennemført commit, dette kaldes en acyclic graph<sup>32</sup> og kan ske fra et lokalt drev til et remote, ved hjælp af Personal Tokens fra Gitlab

<sup>31</sup> <https://docs.djangoproject.com/en/5.0/topics/http/urls>

<sup>32</sup> <https://docs.gitlab.com>

access Token. Vi kan også give autorisation mellem Gitlab og VPS eller bruge Lokale Tokens når vi git push og beder om en Pull Merge Request til produktion leder. Derved sikres kodningen mest muligt for fejl, før den går i produktion.



10. Gitlab pull merge request

## 24 Test

Der er brugt test i test.py med Python test scripts, der er en integreret del af Django framework og som dermed bliver genbrugt som en del af CI/CD pipeline, med live output af testresultater ved brug af Coverage PyPi package. I Management Systems testes, som en start, nyheder\_app ændringer i Databasen, samt HTMX ændringer af ID rækkefølgen med sortable.js<sup>33</sup>. Nyhederne er testet på default rækkefølge, create, update og convert user object til en f-string og formater, 13 test tilsammen, som er under udvikling til at dække alle python statements, also linier i views.py filerne, som kræver over 50 tests. Test.py aktiveres i enntrypoint.sh filen, i to af tre RTE=env variabler, kun RTE=prod er uden coverage report og fail minimum procent størrelse.

Ved at gennemgå testresultaterne med f.eks Coverage report, fås en detaljeret oversigt, hvor koden linje for linje er highlightet og derved viser hvilken statement der er testes på. Kode som ikke er testet, kan skyldes at funktionen ikke er i brug eller at funktionen kan optimeres. Se Bilag - Coverage report.pdf.

<sup>33</sup> <https://htmx.org/events/#htmx:afterRequest>



En anden mulighed for at teste backend og API endpoints er Postmann eller VS Code Thunder Client<sup>34</sup>. Både Postmann og Thunder Client er smart test software med et behageligt layout, især brugbart til pre-production test af især API endpoints, men også HTTP, GET PUT POST eller DELETE test af backend. Test kan gemmes og genbruges i en bagud sikring af backend-funktionalitet i en videre udviklingsfase.

## 25 Pipeline og VPS hosting

Der er oprettet en pipeline til Linode med tokens fra lokal maskine som er placeret i authorization filen i root .ssh mappen, der sikrer en hurtigere workflow fra lokal terminal login uden brug af password. Den samme form for token forbindelse er også implementeret på det Gitlab repository der indeholder projektet.

Linode er et serverless tilbud, med maskiner placeret forskellige steder i verden. Management Systems har drev i Frankfurt. Fordelene er at der tilbydes nogle hurtige bits hastigheder, plus at der remote, kan tages custom backup af ens drev også udover den automatisk daglige sikring af en backup kopi. Den anvendte løsning er 1 GB VPS drev, hvilket betyder at Linux Alpine som operativsystem passer fint, da det er et lille funktionelt linux open source OS. Management Systems har derved fået en remote public adresse at udgive fra.

## 26 Docker med Nginx

Docker er software, der installeres på operativsystemet. Docker bygger et images, som kan starte server containeren. Med docker compose up<sup>35</sup> finder det passende image i app mappen og dermed tilføjes kun de layers der er sket ændringer i. Et image er gemt hvor det bliver bygget f.eks hos Gitlab. Git eksekverer Dockerfile og bliver konfigureret af docker-compose.yml filen eksekveret af .gitlab-ci.yml filen ved et push til Gitlab.

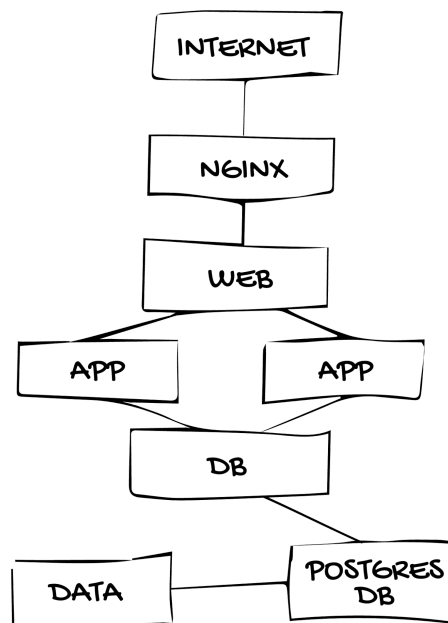
En Docker in Docker configuration i Gitlab bygger et image efter et git push er eksekveret og bygger backend datatyper, efter databasen er skabt, for derefter at migrere databasetabeller og columns.

---

<sup>34</sup> Web Development Things we do in the class 23-08-2022 af Santiago Donoso, Lecturer at KEA

<sup>35</sup> <https://www.docker.com>

Der er flere layers i en container, af væsentlige lag kan nævnes Nginx server med config fil og andre services, herunder et Docker image der er konfigureret til operativsystemet, db med database placering. Derudover kaldes et Nginx image static volumes, samt web apps med data objektet, i Management Systems skifter Nginx mellem to apps til serveren, som er sat op i Nginx config filen.



11. Docker-compose.yml struktur

## 27 Indsigter

Det har været et spændende forløb, hvor udviklingstempoet har været højt. Der blev tænkt meget over teknologiske rammer og muligheder, på den afsatte tid. Der blev testet på AVS E2 instances med Linux Ubuntu, men kom frem til, at der var nogle besværligheder med løbende backup, volumes og det at Ubuntu ikke understøtter PyPi packages. Alt i alt vil der kunne spares meget tid på at udvikle på sikker grund, og derfor faldt valget på Linode med Alpine. Det ligner et projekt med nogle gode muligheder i fremtiden, men det kræver yderligere konceptudvikling og en samlet løsning med et mobile app tilbud. Den største indsigt er at afsætte god tid og at planlægge daglige arbejdsopgaver baseret på korte sprints.

## 28 Den videre udvikling

For Management Systems er hensigten, at stå med et robust fundament, der er test klart for en kundegruppe af små og mellemstore virksomheder. Det betyder også, at der skal skabes fællesskaber, hvor udviklingsmulighederne ses værdifuldt og konkurrenceskabende på markedet. Facebook profil er første skridt mod at kommunikere til et marked.

Som udgangspunkt har der kun været to virksomhedsformer at vælge imellem, EVS enmandsvirksomhed og ApS Anpartsselskabet, det skyldes at der kræves flere end en stiftere for at starte en interesse selskab og at der kræves en væsentlig startkapital ved stiftelse af et A/S Aktieselskab. Det er valgt at stifte et anpartsselskab med en startkapital på

40.000,- kr og med en ejer<sup>36</sup>. Der kan ansættes kvalificerede medarbejdere, og hvis dynamikken fra medarbejderen er særlig funderet, kan der tilbydes medejerskab i virksomheden.

Der er en del TODO's som skal udarbejdes før applikationen kan indgå som et seriøst testmiljø i et firma, bl.a. Databeskyttelsesloven GDPR<sup>37</sup> skal overholdes og en privatlivspolitik i virksomheden kan hjælpe til at opfylde disse krav. Herudover mangler det endelige valg af sikkerhed i settings, SSL, preventdefault scroll og timetal for hver uge, køb og opsætning af domæne m.m.

## 29 Konklusion

Der har været flere udfordringer, bl.a. udvælgelse af koncept. Et andet var at vælge et realistisk mål for projektet. Jeg mener at have fundet den rigtige teknologiske løsning med Django, og at jeg faktisk har opnået de afsatte mål, der blev sat fra starten.

Jeg ved hvilke digitale udviklingsmiljøer som vil fungere i fremtiden, hvor CI, CD og Software Quality Assurance kommer i sin ret og hvilke problemstillinger, der vil opstå, hvis der skal udvikles til et bredt marked af små og mellemstore virksomheder.

---

<sup>36</sup> <https://www.nemadvokat.dk/blog/selskabsstiftelse/de-4-virksomhedstyper>

<sup>37</sup> <https://gdpr.dk>

## 30 Bibliografi

[https://docs.gitlab.com/ee/ci/cloud\\_deployment](https://docs.gitlab.com/ee/ci/cloud_deployment)  
<https://medium.com/@iamalisaleh/nginx-and-django-a-beginners-guide-to-ssl-certificate-setup-6da47bd2ba1f>  
<https://medium.com/@millienakiganda/creating-an-ssl-certificate-for-localhost-in-django-framework-45290d905b88>  
<https://www.billy.dk>  
<https://www.interviewbit.com/blog/python-commands>  
<https://www.geeksforgeeks.org/what-are-user-defined-functions-and-built-in-functions-in-php/#builtin-functions>  
<https://pypi.org>  
<https://www.npmjs.com>  
<https://www.codeconvert.a>  
<https://kinsta.com/blog/php-vs-python>  
<https://kinsta.com/blog/flask-vs-django>  
<https://kinsta.com/blog/php-frameworks>  
<https://kinsta.com/blog/vue-vs-react>  
<https://github.com/dbohdan/classless-css>  
<https://docs.djangoproject.com/en/5.0/ref/databases>  
<https://github.com/sarathak/django-db-benchmark/tree/main>  
<https://htmx.org/events/#htmx:afterRequest>  
<https://www.freepik.com/>  
<https://github.com/teddziuba/django-sslserver>  
<https://webflow.com/>  
<https://docs.gitlab.com/>  
<https://www.miniorange.com/blog/different-types-of-authentication-methods-for-security>  
<https://dashboard.signicat.com>  
<https://www.nemadvokat.dk/blog/selskabsstiftelse/de-4-virksomhedstyper>  
<https://chatgpt.com>  
<https://www.facebook.com/profile.php?id=61560507946455z>  
<https://mailchimp.com>  
<https://gdpr.dk>