

# ObliGentle: Procesrapport

Svendeprøve ved  
Kenneth Løvgren  
&  
Rasmus Ladefoged Wolffram

Aarhus Tech

Sune Koch Rønnow  
sune@kochroennow.dk

8. december 2023

# Titelblad

<b>Deltagere</b>	Sune Koch Rønnow
<b>Projektnavn</b>	ObliGentle
<b>Skole</b>	Aarhus Tech
	Hasselager Allé 2, 8260 Viby J
<b>Projektperiode</b>	13. november 2023 - 15. december 2023
<b>Afleveringsdato</b>	8. december 2023
<b>Vejleder</b>	Kenneth Løvgren & Rasmus Ladefoged Wolffram

## Underskrifter

Sune Koch Rønnow	Dato
------------------	------

Kenneth Løvgren	Dato
-----------------	------

Rasmus Ladefoged Wolffram	Dato
---------------------------	------

# Indhold

# Kapitel 1

## Formalia

### 1.1 Læsevejledning

I forbindelse med svendepróven er vi blevet pålagt at aflevere to rapporter, en om projektets proces og en om dets produkt. Dette er procesrapporten, som fokuserer på min overvejelser og refleksioner i forhold til projektet. Jeg antager, at læseren har en teknisk viden på niveau med en færdiguddannet datateknikker.

### 1.2 problemformulering

Lav en opgaveorganiserings-/kalenderapplikation, hvor brugere har mulighed for at styre forskellige typer af opgaver:

- **Aftaler:** Opgaver som bruger skal gøre på, eller inden et bestemt tidspunkt og som automatisk bliver færdiggjort på det pågældende tidspunkt.
- **Projekter:** Opgaver som brugeren selv har valgt, hvor brugeren kan notere fremgang og færdiggørelse.
- **Tjanser:** Tilbagevendende opgaver som brugeren skal have gjort ved lejlighed og hvor kun de vigtigste tjanser vises.

Applikationen skal laves som en *RESTful API* med *Crossplatform* brugerflader. Det skal være muligt for brugere at tilgå deres profil både fra et webinterface og en android app. Applikationen skal være modulært opbygget og rumme mulighed for, at jeg eller andre udviklere, kan udviklere videre på den. Applikationen skal i udgangspunktet sættes op online og have en *proof-of-concept tilgængelig*.

### 1.3 case

Tema/titel for	ObliGentle
Godkendt nummer	
Casebeskrivelsen	<p>Det er en reel udfordring for mig, at jeg på den ene side godt kunne bruge et program til at hjælpe mig med at organisere mit privatliv, men på den anden side oplever jeg, at de fleste programmer enten er udviklet til arbejdsmarkedet eller fungerer som om de var. Dette fungerer ikke for mig, fordi jeg ikke gider at føle arbejdsdagen forsætter, efter jeg er kommet hjem. Derfor er idéen bag ObliGentle at lave en opgave organisator/kalender, som fokuserer på at hjælpe brugeren med det, som man har overskud til uden at den udskammer en for det man ikke nåede. Programmet skal kunne håndtere tjanser, ting som brugeren skal gøre, projekter, ting som brugeren gerne vil gøre, og aftaler, samt ting brugeren skal gøre på eller før et bestemt tidspunkt. Programmet skal hjælpe brugeren med at organisere sin opgaver, men hverken moralisere, direkte eller indirekte, om brugeren når nok eller optimerer sin produktivitet tilstrækkeligt. Derimod skal programmeret fejre, hvad det bliver gjort, og hjælpe med at prioritere det vigtigste, fremfor at alt skal gøres hver dag.</p> <p>Programmet kommer til at være baseret på personlige præferencer og ikke nyeste forskning. Det havde selvfølgelig været præferablet, men det er ikke realistisk at orientere sig i et nyt forskningsområde ved siden af at man lave sin svendeproeve. Programmet forsøges dog at laves modulært og nemt foranderligt, så det i en potentiel fremtid ville kunne tilpasses forskningsfeltet<sup>1</sup>.</p>
Teknologier	Python & FastAPI JavaScript og React Native MongoDB
Fagområder som projektet dækker	Clientside programmering Serverside programmering GUI programmering Programmeringsmetodik Objektorienteret programmering Database programmering APP programmering
Valgfri specialefag som projektet dækker	

---

<sup>-1</sup>Eller mere individuelle behov, da folk nok organisere sig selv bedst på forskellige måder  
<sup>0</sup>Eller mere individuelle behov, da folk nok organisere sig selv bedst på forskellige måder  
<sup>1</sup>Eller mere individuelle behov, da folk nok organisere sig selv bedst på forskellige måder

## Kapitel 2

# Metode- og teknologivalg

### 2.1 Udvikling og planlægning

Valget af udviklingsmetode endte med at være en større udfordring for mig, end jeg egentlig havde antaget. På den ene side er jeg primært blevet oplært i *scrum* og har trives så godt i det, at jeg ikke rigtigt har søgt erfaringer i så mange andre; men på den anden side er det tvivlsomt, i hvilken grad en én-persons *scrum* proces overhovedet er mulig.

Hvorvidt en én-persons scrum proces overhovedet er mulig, er et forbløffende debatteret emne på nettet<sup>12</sup>, og egentlig en debat, som jeg ikke ser nogen væsentlig grund til tage del i, men jeg vidste, at jeg ikke havde i sinde at udnævne mig selv hverken til *Scrummaster* eller *produktowner*, fordi det, i hvert fald i min læsning af scrum, er ret vigtigt, at disse er separate opgaver. Jeg forsøgte at gøre op med mig selv, hvad jeg egentlig på et dybere plan ønskede at få ud af Scrum, når det nu ikke var de faste roller: Det var i virkeligheden dens rod i agile udvikling, så det besluttede jeg mig for at fokusere på, i særdeleshed Agile Alliance 101<sup>3</sup>.

Særligt fokuserede jeg på:

- Holde daglige ajourføringer
- Dele projektet op i 3 iterationer/sprint
- Hvert sprint indebærer en planlægnings- og evalueringssession
- Indstille mig på, at processen kan ændres efter behov og de daglige ajourføringer og særlig sprintevalueringen skal danne et rum for dette

Derudover forsøgte jeg også at have de 12 principper<sup>45</sup>, hvor princip 1 og 4 naturligvis blev ret frit fortolket, da jeg dårligt kan agere min egen kunde. Årsagen var, at jeg forsøgte endvidere at planlægge både for mit produkt og dokumentation og delte forløbet op i 3 sprints, hvor jeg havde 2 halve uger til start og slut, som fokuserede på henholdsvis at begynde og afslutte forløbet.

---

<sup>1</sup><https://www.scrum.org/forum/scrum-forum/36139/one-man-scrum-team-possible>

<sup>2</sup><https://pm.stackexchange.com/questions/9081/a-one-man-project-methodology>

<sup>3</sup><https://www.agilealliance.org/agile101/>

<sup>4</sup><http://agilemanifesto.org/principles.html>

<sup>5</sup><https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

Sidste men ikke mindst, så brugte jeg kanban-bræt for at holde overblik over opgaver. I slutningen af projektet, bevægede jeg mig lidt væk fra brættet og styrede det mere igennem logbogen, ajourføringerne og evalueringerne, men særligt i starten af projektet var det et vanvittigt godt redskab.

Daglige ajourføringer : Hver formiddag afholdte jeg en daglig ajourføring , hvor jeg stillede følgende spørgsmål : Daglig ajourføring Holdes hver arbejdsformiddag , helst inden 10. Skriv kun ja , nej eller stikord . Hvis noget skal uddybes , så gør det nede i loggen .  Gårsdagen log : Fuldendt? Tilstrækkelig? Renskrevet?  Arbejdsproces : Opfølgninger? Blokeringer? Forandres?  Hvad er dagens prioriteter ?
--

Disse er dokumenterede i logbogen. Ajourføringerne holdt samme form igennem hele processen. Dette skyldes, at jeg haft et svendeforløb før, hvor jeg lærte en del om, hvordan jeg arbejdede bedst med ajourføringerne.

Procesevalueringer : Ved slutningen af hvert sprint , holdt jeg en procesevaluering , som endte med at tage denne form : Evaluering ... af Sidste evaluering Seneste sprint Tidsplanen Produkt Konklusion
--

De 3 evalueringer kan også findes i bilagsmaterialet. Procesevalueringerne skifter i højere grad form, hvor den første er helt andet format end de efterfølgende Sprintplanlægning: Sprintplanlægningen endte med at blive indtastet i kanban-brættet for at sikre mig, at det var ført ordentligt ajour.

Dette er Agile development og jeg mener også, at man kan se en klar inspiration fra Scrum, om end det er underimplementeret, men det leder jo til et naturligt spørgsmål. Hvordan kan det være, at man ikke anvendte egentligvis Extreme Programming (XP), som er et agile framework, men som godt kan bruges af en person? Der er svaret simpelthen blot, at det er det her, som jeg vil have ud af det.

## 2.2 RESTful API Arkitektur

RESTful API kombinerer REST, som står for *Representational state transfer* , softwarearkitekturen med en API. RESTful API indeholder 5-6 arkitektoniske retnings-



linjer<sup>6,7,8,9</sup>

- Klient-server arkitektur igennem HTTP
- "Stateless" klient kommunikation, hvilket vil sige, at hver "*request*" er separat og der ikke gemmes data om klienten
- "*Cacheable*" data - for bedre performance for klientsiden og bedre skalering på serversiden.
- Ensartet interaktion med *API'en*, hvor data anmodes tydeligt og separat fra klientens repræsentation deraf
- *lag*-inddelt serverarkitektur, hvor data sendes igennem flere lag, usynligt for klienten.
- "*Code-on-demand*" er at serveren kan sende kode, som eksekveres af klienten. Denne retningslinje er dog bredt anerkendt, som værende valgfri og ikke nødvendig.

Med andre ord, så handler RESTful API om, at man giver klienter mulighed for let og tydeligt at anmode og indsende data igennem http-request, men man styrer selv behandlingen af dataen på serveren grundet sikkerheds- og skaleringshensyn. Data sendes frem og tilbage mellem server og klient og ansvaret for at repræsentere dataen placeres i højest mulige grad hos klienten. For at RESTful API faktisk er "*performance*-let og sikker, er det alfa-omega, at man sørger for at grænsefladerne fungerer, som de skal.

SOAP<sup>10</sup> er et ældre alternativ. SOAP få request igennem en eller flere "*application layer*"-protokoler, så som *HTTP*, *SMTP* eller *TCP*, men returnerer altid et *XML* dokument. Kontra *RESTful API* så har *SOAP* flere indbyggede standarder og servicier, eksempelvis i forhold til sikkerhed, men er også markant tungere for serveren<sup>11</sup>. Af mere moderne alternativer eksisterer også *GraphQL*, som fokuserer på kun at overføre det korrekte data fremfor at minimere server ressourcer, og *RPC*<sup>12</sup>, som fokuserer på at køre kode lokalt på klienten, men bruger serveren som en "*dependency*", som om den var lokal og man kan have logikken på serveren<sup>13</sup>. Både GraphQL og RPC er ret spændende, men jeg tror, at det er korrekt valgt at fokusere på ressourceforbrug i dette tilfælde og SOAP virker dateret i forhold til andre alternativer.

Potentielt kunne det være spændende at se på at sammenligne projektet med et ellers lignende *GraphQL*-og *RPC*-projekt, men herfra hvor jeg står nu, virker *RESTful API* som det korrekte valg.

## 2.3 Software design: SOLID-principper

**SOLID** er en række principper for at lave den bedst mulige *objekt-orienterede programmering* som muligt og inkluderer fem principper.

---

<sup>6</sup>i ingen særlig rækkefølge

<sup>7</sup><https://www.redhat.com/en/topics/api/what-is-a-rest-api>

<sup>8</sup><https://en.wikipedia.org/wiki/REST>

<sup>9</sup><https://www.ibm.com/topics/rest-apis>

<sup>10</sup>**S**imple **O**bject **A**ccess **P**rotocol

<sup>11</sup><https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>

<sup>12</sup>**R**emote **P**rocedure **C**all

<sup>13</sup><https://blog.bitsrc.io/not-all-microservices-need-to-be-rest-3-alternatives-to-the-classic-41cedbf1a907>

**SOLID**-principper kan fuldt ud implementeres i *Python*<sup>14</sup> og *JavaScript*<sup>1516</sup>, men ingen af dem har egentlige interface implantationer, som man eksempelvis finder C#. Begge bruger såkaldt ”*duck typing*” i stedet for normativ eller strukturel typesystem. Ikke desto mindre kan interfaces sagtens implementeres, men det har jeg lige forklaret i dets eget afsnit.

Alle **SOLID**-principper kan der skrives og siges meget om, hvilket er blevet gjort. Jeg vil forsøge at forklare dem så simpelt som muligt, hvilket vil medføre, at jeg flere steder vil være reduktiv og oversimplificerende i min gennemgang. For en længere og væsentlig mere grundig gennemgang kan jeg anbefale følgende ressourcer<sup>171819</sup>.

### 2.3.1 Single-Responsibility Principle

Nok det simpleste af principper og kan i bund og grund reduceres til, at en klasse skal gøre en ting og ikke flere.

### 2.3.2 Open-Closed Principle

Det skal være muligt at udvide en klasse/module/software entitet, men ikke ændre den. Så det er godt og positivt at bygge videre på andet software, men vores klasser skal ikke kunne ændre hinanden.

### 2.3.3 Liskov Substitution Principle

Reglerne og bestemmelserne for en ting, skal også gælde for dens undertyper af samme ting. Hvad vigtigere er, så er det, at tingen ikke skal have en masse regler og bestemmelser, som ikke også er gældende for undertyperne.

### 2.3.4 Interface Segregation Principle

”interfaces” tilhører klasserne og omvendt. Derfor skal klasser ikke gøres afhængige af funktioner, som de ikke bruger. I stedet for skal interfaces underinddeles og gøres mere specifikke, så det passer til klasserne.

### 2.3.5 Dependency Inversion Principle

Dette er desværre et lidt misvisende navn, for det handler ikke om at vende afhængighederne rundt, men afkoble dem og have abstraktioner i mellem.

### 2.3.6 ”Duck typing” og interfaces

I normativ typesætning, som man finder vi blandt andet C++, C# og Java<sup>20</sup>, eksplicit deklarerer deres type og så er den såkaldte ”*duck typing*” kendt for, at hvis ”*det går*

---

<sup>14</sup><https://realpython.com/solid-principles-python/>

<sup>15</sup><https://dev.to/denisveleae/5-solid-principles-with-javascript-how-to-make-your-code-solid-1kl5>

<sup>16</sup><https://blog.logrocket.com/solid-principles-single-responsibility-in-javascript-frameworks/>

<sup>17</sup><https://stackify.com/solid-design-principles/>

<sup>18</sup><http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>

<sup>19</sup><https://realpython.com/solid-principles-python/>

<sup>20</sup>[https://en.wikipedia.org/wiki/Nominal\\_type\\_system](https://en.wikipedia.org/wiki/Nominal_type_system)

som en and og ligner en and, så er det en and", så programmet bestemmer typen efter opførelse og hvad det ligner og ikke deklARATIONER. "Duck typing" adskiller sig fra strukturel typesystemet, som man finder i eksempelvis typescript, Go og OCaml<sup>21</sup>, ved kun noget, af den "duck type"-ede struktur evalueres, hvorimod der det strukturelle typesystem evalueres på hele strukturen for at se, om to er kompatible. Så selvom Python og Javascript ikke rigtigt har interfaces, så kan man lave klasser, som opfører sig som et interface, og som også vil virke som et interface. Så man kan lave en klasse, som man aldrig initierer og på den måde vil de agere som en abstrakt klasse.

**SOLID**-principperne står ret stærkt i *Objekt-Orienteret Programming (OOP)*. Personligt, sådant helt anekdotisk, er min oplevelse, at programmører oftere kritiserer **OPP** end **SOLID**. Et direkte alternativt til **SOLID** er der dog i *GRASP*<sup>22</sup>, som er et ældre **OPP** principsæt, men så kan man vel næsten lige så godt bruge **SOLID**. **SOLID** er også blevet bearbejdet, således, at det angiveligt også skulle omfavne funktionelt programming og multiparadigme *microservices*, men der er ikke helt enighed, om det rent faktisk er det mest fornuftige og Poul Merson har eksempelvis foreslået hans egne *IDEALS*-principper til *microservices*<sup>23</sup>.

Personligt er min erfaring, at **SOLID** i langt højere grad indeholder nogle mål at aspirere i mod, men at man altid kan få ens kode til at blive mere og bedre **SOLID** og det særligt er anvendt som målestok, når man refactor ens kode.

## 2.4 Tech Stack: FAR(n)M

**FARM**-tech-stack står **FastAPI**, **React** og **MongoDB**. Jeg har tilføjet et lille n efter R'et, fordi jeg bruger **React Native**. **FARM**-techstack'en er langt fra at være bredt etableret, men er eksempelvis nævnt af *MongoDB*<sup>24</sup>. Ikke desto mindre er det en valid *tech-stack*, som tillader mig at arbejde med **Python** og **JavaScript** i et *MEVN/MERN* lignende stack. Stack'en er også valgt for at prøve en nyere og spændende tech-stack, som ikke er helt så etableret og som jeg har lidt mindre erfaring med.

### 2.4.1 Sprog: Python

**Python** bliver blot ved med at stige i popularitet og selvom det kan blive anset lidt som *datascience*- og begynderfokuseret sprog, har det en bred palette af anvendelighed.

På trods af, jeg selv er lidt af en **Python** fortaler, har jeg ikke brugt sproget i *full*- eller *web*-stack sammenhæng og derfor ønskede jeg at lave *backend*'en i **Python**.

Til projektet har jeg lavet et min nyt environment med *Anaconda*<sup>25</sup>, som min package manager, for at undgå, at projektet løber ind i nogle problemer fra tidligere. Jeg endte dog med at reinstallere det, fordi jeg troede, at der var gået noget galt med miljøet, hvilket vidste sig at være en nedarvingsfejl fra *FastAPI*-users' *beanie* modul.

### 2.4.2 FastAPI

**FastAPI**<sup>26</sup> er både F'et og A'et i **FARNM**, så vigtig er den for stack'en.

<sup>21</sup>[https://en.wikipedia.org/wiki/Structural\\_type\\_system](https://en.wikipedia.org/wiki/Structural_type_system)

<sup>22</sup>[https://en.wikipedia.org/wiki/GRASP\\_\(object-oriented\\_design\)](https://en.wikipedia.org/wiki/GRASP_(object-oriented_design))

<sup>23</sup><https://www.infoq.com/articles/microservices-design-ideals/>

<sup>24</sup><https://www.mongodb.com/developer/languages/python/farm-stack-fastapi-react-mongodb/>

<sup>25</sup><https://www.anaconda.com/download/>

<sup>26</sup><https://fastapi.tiangolo.com/>

**FastAPI** er et framework til at bygge *API'er* i *Python* hurtigt, og det er blevet ret populært.

**FastAPI** er den primære grund til, at jeg gerne ville arbejde med denne stack.

### 2.4.3 Anvendte Python og FastAPI moduler

Modul og link	Beskrivelse	Anvendelse
Beanie <sup>27</sup>	Asynkron ODM for MongoDB baseret på pydantic	FastAPI-users dependant
FastAPI <sup>28</sup>	API framework	Til at lave API'en. Alt fra fouter, Reponse, exception, json encoding, CORS og Middleware med mere
Motor <sup>29</sup>	MongoDB driver	Sammen med asyncio til at oprette asynkron forbindelse MongoDB
Pymongo <sup>30</sup>	Endnu en MongoDB driver	Til at lave synkrone forbindelser.
uvicorn <sup>31</sup>	Asynchronous Server Gateway Interface web server	Til at kører min API
BSO <sup>32</sup>	Et BSON kodeks uafhængig af MongoDB	Til at generere ObjectId kompatible med MongoDB
datetime <sup>33</sup>	Inkludere en Datetime	Til at sætte dato og klokkeslæt på oprettelser og opdateringer
typing <sup>34</sup>	Type hints	til datamodeller, således de kan opdateres uden alle værdier sættes.
Pydantic <sup>35</sup>	Data validerings type hints	Datavalidering til datamodellerne
FastAPI_users <sup>36</sup>	Registrering og autentifikation til FastAPI	Håndtere bruger registrering og autentifikation
HTTPX OAuth <sup>37</sup>	Asynkront OAuth	Til at lave 2faktor autentifikation
os <sup>38</sup>	Indbygget os modul	Til at få 2fa miljøet fra systemet

## 2.4.4 Sprog: JavaScript

Jeg brugte **JavaScript** som mit frontend sprog. Oprindeligt designet til at være et letanvendeligt *object scripting language* med fokus på HTML<sup>39</sup>. Det blev født, da Netscape bad en en scheme udvikler, Brendan Eich, om at skrive et scripting programmeringssprog, hvis syntaks mindede om Java's, som Netscape allerede havde fået implementeret i deres browser *Navigator*<sup>40</sup>. **JavaScript's** navn har ofte været en kilde til forvirring, men dets forbindelse til Java er ikke tættere end denne.

*JavaScript* er dog et af de, hvis ikke det, mest populære programmeringssprog til webudvikling og var derfor et oplagt valg til at udvikle min *frontend*.

## 2.4.5 Framework: React Native

Er et open-source UI-framework udviklet og driftet primært af Facebook/Meta<sup>41</sup>. **React Native** er baseret *React*, facebook's UI-framework til webudvikling, men målrettet mod at kunne køre "native" på mobile platforme.

---

<sup>3</sup><https://pypi.org/project/beanie/>

<sup>4</sup><https://pypi.org/project/fastapi/>

<sup>5</sup><https://pypi.org/project/motor/>

<sup>6</sup><https://pypi.org/project/pymongo/>

<sup>7</sup><https://pypi.org/project/uvicorn/>

<sup>8</sup><https://pypi.org/project/bson/>

<sup>9</sup><https://pypi.org/project/DateTime/>

<sup>10</sup><https://pypi.org/project/typing/>

<sup>11</sup><https://pypi.org/project/pydantic/>

<sup>12</sup><https://pypi.org/project/fastapi-users/>

<sup>13</sup><https://pypi.org/project/httpx-oauth/>

<sup>14</sup>[https://www.w3schools.com/python/module\\_os.asp](https://www.w3schools.com/python/module_os.asp)

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

<sup>16</sup><https://pypi.org/project/fastapi/>

<sup>17</sup><https://pypi.org/project/motor/>

<sup>18</sup><https://pypi.org/project/pymongo/>

<sup>19</sup><https://pypi.org/project/uvicorn/>

<sup>20</sup><https://pypi.org/project/bson/>

<sup>21</sup><https://pypi.org/project/DateTime/>

<sup>22</sup><https://pypi.org/project/typing/>

<sup>23</sup><https://pypi.org/project/pydantic/>

<sup>24</sup><https://pypi.org/project/fastapi-users/>

<sup>25</sup><https://pypi.org/project/httpx-oauth/>

<sup>26</sup>[https://www.w3schools.com/python/module\\_os.asp](https://www.w3schools.com/python/module_os.asp)

<sup>27</sup><https://pypi.org/project/beanie/>

<sup>28</sup><https://pypi.org/project/fastapi/>

<sup>29</sup><https://pypi.org/project/motor/>

<sup>30</sup><https://pypi.org/project/pymongo/>

<sup>31</sup><https://pypi.org/project/uvicorn/>

<sup>32</sup><https://pypi.org/project/bson/>

<sup>33</sup><https://pypi.org/project/DateTime/>

<sup>34</sup><https://pypi.org/project/typing/>

<sup>35</sup><https://pypi.org/project/pydantic/>

<sup>36</sup><https://pypi.org/project/fastapi-users/>

<sup>37</sup><https://pypi.org/project/httpx-oauth/>

<sup>38</sup>[https://www.w3schools.com/python/module\\_os.asp](https://www.w3schools.com/python/module_os.asp)

<sup>39</sup><https://web.archive.org/web/20070916144913/https://wp.netscape.com/newsref/pr/newsrelease67.html>

<sup>40</sup><https://web.archive.org/web/20200227184037/https://speakingjs.com/es5/ch04.html>

<sup>41</sup><https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>

Mit oprindelige valg var *Ionic*, men endte med at fravælge det, da ionic er ret langsomt og, selvom det formentlig er brugbart til at lave et *proof-of-concept* som dette, så tænkte jeg, at det formentlig var bedre bruge et mere industriegnet framework – også selvom jeg havde mindre erfaring.

#### 2.4.6 Anvendte JavaScript og React Native modul

Modul og link	Beskrivelse	Anvendelse
react <sup>42</sup>	Bibliotek til at lave UI	Brugt det til at lave UI
react-native <sup>43</sup>	React Native biblioteket til at lave native UI	Brugt det til at lave UI
axios <sup>44</sup>	"Promise"-baseret http-klient	Brugte det til mine frontend API kald
react-native-modal <sup>45</sup>	Modal komponent, som håndterer pop-ups	Til når der opdateres eller skabes et nyt "task"-element
expo-constants <sup>46</sup>	Giver informationer om konstanter	Til at få højde på statusbaren på android enheder
react-navigation/bottom-tabs <sup>47</sup>	bottom tab navigator	Bottom-tab navigationsbaren.
react-navigation/native	React native's navigations håndtering	native Til at lave min "navigations container"
react-navigation/stack <sup>48</sup>	App navigation ved at "stack" sider oven på	Til at lave min "stack" navigation

## 2.4.7 MongoDB

Jeg har arbejdet med **MongoDB**<sup>50</sup> før og føler også, at *NoSQL* database imødekommer projektets behov for at kunne blive udbygget modulært og over længere tid. Det er bare en kæmpe fordel ikke at skulle have defineret en database, hvis man senere beslutter sig for, at produktet skal kunne rumme noget helt andet.

Desuden er **MongoDB**'s *NoSQL* også mere fleksibel og selvom ObliGentle ikke kommer til at blive stort nok til, at jeg for alvor kan gøre brug af *NoSQL*'s skaleringsmuligheder, så er det rart at arbejde med og ville være vigtig, hvis jeg beslutter mig for senere at ville færdigudvikle ObliGentle.

Der er åbenlyse alternativer til *FARM*-stack'en, så som *MEVN*<sup>51</sup> - eller *MERN*<sup>52</sup>-stack. De kunne begge have fungeret godt, og som nævnt før, så skyldes det i højere grad en præference for at bruge Python og FastAPI, end at der er noget i vejen med de andre stack'er. *LAMP*<sup>53</sup> -stack er også oplagt at nævne, men vil helst ikke arbejde i PHP. Et spændende alternativ kunne være *PERN*<sup>54</sup>-stack'en, men ville gerne arbejde med en *NoSQL* database. *Ruby on Rails* havde også været en spændende mulighed.

Alt i alt er jeg dog tilfreds med stack'en som den blev. *MEVN* var den anden store mulighed for mig, men med *Python* vandt *FARM* bare for mig.

## 2.4.8 GitHub

Helt fra starten af projektet har jeg benyttet **GitHub** til versionering. Jeg brugte VS code's indbyggede værktøjer til at forbinde til **GitHub** det meste af tiden, men da jeg skulle ligge to repositories sammen, så måtte jeg også bruge noget *Git Bash*.

---

<sup>26</sup><https://www.npmjs.com/package/react>

<sup>27</sup><https://www.npmjs.com/package/react-native>

<sup>28</sup><https://www.npmjs.com/package/axios>

<sup>29</sup><https://www.npmjs.com/package/react-native-modal>

<sup>30</sup><https://www.npmjs.com/package/expo-constants>

<sup>31</sup><https://www.npmjs.com/package/@react-navigation/bottom-tabs>

<sup>32</sup><https://www.npmjs.com/package/@react-navigation/>

<sup>33</sup><https://www.npmjs.com/package/@react-navigation/stack>

<sup>34</sup><https://www.npmjs.com/package/react>

<sup>35</sup><https://www.npmjs.com/package/react-native>

<sup>36</sup><https://www.npmjs.com/package/axios>

<sup>37</sup><https://www.npmjs.com/package/react-native-modal>

<sup>38</sup><https://www.npmjs.com/package/expo-constants>

<sup>39</sup><https://www.npmjs.com/package/@react-navigation/bottom-tabs>

<sup>40</sup><https://www.npmjs.com/package/@react-navigation/>

<sup>41</sup><https://www.npmjs.com/package/@react-navigation/stack>

<sup>42</sup><https://www.npmjs.com/package/react>

<sup>43</sup><https://www.npmjs.com/package/react-native>

<sup>44</sup><https://www.npmjs.com/package/axios>

<sup>45</sup><https://www.npmjs.com/package/react-native-modal>

<sup>46</sup><https://www.npmjs.com/package/expo-constants>

<sup>47</sup><https://www.npmjs.com/package/@react-navigation/bottom-tabs>

<sup>48</sup><https://www.npmjs.com/package/@react-navigation/>

<sup>49</sup><https://www.npmjs.com/package/@react-navigation/stack>

<sup>50</sup><https://www.mongodb.com/try/download/community>

<sup>51</sup>MongoDB, Express.js, Vue.JS & Node.js

<sup>52</sup>MongoDB, Express.js, React.js & Node.js

<sup>53</sup>Linux, Apache, MySQL & PHP

<sup>54</sup>PostgreSQL, Express.js, React.js & Node.js

Det var målet at få brugt det flittigt og det lykkedes ikke for mig. Men jeg mistede ikke noget data, så den del er ok.

### 2.4.9 MongoDB Compass

**MongoDB Compass** har jeg brugt til at interagere direkte med databasen. Den er ikke fantastisk, men er rar at have til at fejlfinde, teste og gøre ting, som jeg ikke ønsker at skrive ind i programmet.

Jeg brugte **MongoDB Compass** til at oprette databasen og collections samt inspicere databasens indhold.

**MongoDB Compass** er **MongoDB's native GUI**, og jeg vidste, at de havde de begrænsede funktioner, som jeg havde brug for og derfor valgte jeg blot at bruge dem.

### 2.4.10 OneNote

Til at føre og holde styr på min noter brugte jeg **OneNote**<sup>55</sup>. Det er en fantastisk hjælp at kunne holde styr på tanker om projektet på farten.

Logbøgerne er siden hen blevet skrevet over i  $\text{\LaTeX}$ , men det har været guld værd i arbejdsprocessen.

### 2.4.11 Swagger UI

Jeg brugte **Swagger UI**<sup>56</sup> til at teste min API.

SwaggerUI kommer som en del af **FastAPI** og kan teste ens **API** fra browseren. Jeg har tidligere brugt *Postman* og har haft gode erfaringer med den.

### 2.4.12 TexMaker og MikTek

En *opensource*  $\text{\LaTeX}$ -editor (**TexMaker**<sup>57</sup>) og package-manager (**MikTeX**<sup>58</sup>). Der er et væld af alternativer, men disse er jeg vandt til og glade for at bruge.

### 2.4.13 Visual Studio Code

**VSC** er nærmest en industristandard og den vi har brugt mest på skolen. Den kan godt føles lidt bloateret til tider, men det er klart den, som jeg har mest erfaring med og fordi jeg eksperimenterede med andre ting, ønskede jeg ikke også at gøre det med mit *editor*-valg.

Jeg kunne formentligt med fordel af gjort brug af en fuld IDE, særligt i forhold til *React Native*, men kan simpelthen ikke lide at programmere på den måde. Så for mig er **VSC** en god mellemvej.

---

<sup>55</sup><https://www.onenote.com/Download>

<sup>56</sup><https://swagger.io/tools/swagger-ui/>

<sup>57</sup><https://www.xmlmath.net/texmaker/>

<sup>58</sup><https://miktex.org/>





## Kapitel 3

# Tidsplan

Tidspunkt	Projekt	Dokumentation	Realiseret
13-21/11 Opvarmning Præsprint		Lav <i>case</i> , <i>problemformulering</i> , <i>Kravsspecifikation</i> , <i>tidsplan</i> , <i>mockup</i> og send dem til godkendelse	Realiseret, samt også lavet et flowchart
13-21/11 Startsprint Sprint 1	Lav API og Test API'en Lav en rudimentær frontend	Begynd begge rapporterne og skitser en overordnet struktur. Skriv første udkast til metode- og teknologiafsnittet	Realiseret, dog uden rudimentærfrontend
22-28/11 Fremstød Sprint 2	Gør frontend færdig API på nettet og sørg for kan tilgås af programmet Sørg for frontend kan gøres fra både tablet og web	Beskriv løbende program features, til produktoprapporten, og erfaringer og udfordringer til procesrapporten	Bagud på dokumentation, frontned åd alt tiden, begyndte dog på features liste
29/11-5/12 Slutsprint Sprint 3	Program testning Ryd op i koden og sikre der er kommenteret tilstrækkeligt Lav en liste over mulige <i>features</i> Frasorter fra feature-listen, 17 hvad der ikke kan nås, implementer, hvad der kan	Beskriv testene, features (både kasserede og implementerede)	Kom efter det på dokumentation, fik lavet fraktioneret frontend
6-8/9 Dokumentationsr Postsprint	Sidste test af programmeret kører Eksporter filer og sikre dig, at de	Færdiggør og aflever rapporterne	realiseret dokumentation, problemer med at samle fron-

## Kapitel 4

# udvalgte refleksioner

### 4.1 AI

Det var min tanke, at jeg ville have udviklet noget AI. Der kom forskellige ting i vejen, navnlig tidspres. Det gik det også op for mig, at det kunne være fedt for mig at have udviklet, men det formentlig ikke vidste mine bedste kompetencer frem

AI udvikling har en tendens til at bestå nogle relativt anonyme API kald og handler om noget teknologi, som vi heller ikke forstår særligt godt.

Derfor kan det være svært egentlig at vise meget teknisk formåen med det medmindre, at man laver rigtig meget af det og det var der nok aldrig tid til. Jeg var tæt på at udvikle en funktion til at populere de forskellige opgaver.

Sidst er der også en nogle etiske udfordringer for den måde, som AI bliver udviklet på nu, som jeg ville skulle forholde mig til på en anden måde, end jeg synes, at jeg har haft tid til.

Med det sagt er det grundet tidspres, at det ikke er blevet en del af opgaven, fordi selvom det potentielt er relativt nemt, så kunne det have være spændende at vise lidt frem af.

### 4.2 ”Comment first”-kodning

Jeg besluttede mig at forsøge at lave ”Comment first”-kodning, som for de uindviede er, at man kommenterer først og koder bagefter. Det skulle give mere meningsfyldte kommentarer, og tvinger også én til at tænke over koden før i stedet for siden. Det gik rigtigt godt i starten, men i forbindelse med, at tiden blev knappere og jeg oplevede, at frontend'en drillede, mistede jeg fokus på det. Jeg oplever dog stadig, at det var et fornuftigt valg, fordi API'en er bedre kommenteret end jeg er vant til.

### 4.3 Modulærudvikling

Det var en del af idéen med opgaven, at programmet skulle udvikles så modulært, som muligt, og have mulighed for ikke blot at blive bygget videre på, men også at bygge det i forskellige retninger. For at opnå dette, har jeg i særdeleshed gjort to ting: Først og fremmest har jeg valgt en NoSQL database, således at det ikke er nødvendigt

at ændre database-strukturen. Desuden har jeg sørget for, at både min frontend og backend "task"-operationer er bygget op efter "solid" principper og med en "interface", som andre opgavetyper kan nedarve fra. Jeg havde en idé om, at det kunne være fedt, hvis brugerne selv kunne vælge at oprette den slags opgaver, som passede til dem – eventuelt ved hjælp af noget AI.

## 4.4 React Native

Som tidligere skrevet i metodeafsnittet , så endte jeg med at vælge React Native over Ionic, som egentlig havde været min første valg. Grunden hertil var, at Ionic er langsomt og ikke lige så anvendt i industrien. Jeg overvejede også flutter i stedet for React Native. Modargumentet var, at jeg skulle fokusere på, hvad jeg allerede kendte. Grundet min begrænsede erfaring og entusiasme for Ionic og frontend udvikling generelt; samt at jeg gerne ville forbedre mine full-stack kvalifikationer, gik jeg med React Native. Det var en fejl. Det har været rigtig lærerigt at lære et helt nyt framework fra bunden af og havde det ikke været en svendeproeve, så kunne jeg nok også have hygget mig med det, men det var simpelthen et dårligt valg til en svendeproeve, så jeg skulle have holdt fast i Ionic. Det har simpelthen taget for lang tid og alt for meget arbejdstid er gået med frontend udvikling i stedet for noget der kan vises frem, så jeg skulle have fokuseret mere på at fremvise mine kvalifikationer end at tillære mig nye.

## Kapitel 5

# Mangler og videreudvikling

### 5.1 Mangler

Grundet tidspres og mange problemer med React Native, har jeg måtte aflevere et program med nogle mangler.

- Det har problemer med at fetch'e på task menu'en, selvom API virker og det har virket tidligere
- Den tager sin database for access-token i stedet for brugeren.
- Jeg har ikke fået implementeret 2faktor autorisering i frontend, så derfor har jeg også fjernet den fra backend
- Jeg har ikke fået lagt API'en på nettet, så applikationen kan kun køres lokalt
- jeg fik ikke lavet noget AI til programmet
- fordi appointments, chore og project skærme er udvidelser af task skærmen, så de ikke implementeret færdigt endnu, selvom de er klar til
- Der mangler et brugervenligt og pænt design
- forgotten password kan produce den relevante token, men ikke sende en mail til brugeren.

Mange af disse er jo ikke store ting, men da jeg skulle samle komponenter af flere mindre programmer, som umiddelbart virkede, så endte jeg med alle disse fejl.

Jeg vil dog prøve at få nogen af dem inden det mundlige forsvar. Jeg notere ændringer, som jeg laver her<sup>1</sup>

### 5.2 Videreudvikling

Der var dog flere idéer, som jeg tidligere i processen indså, at jeg ikke kunne nå, men stadig er gode idéer at kunne implementere i programmet:

- Flere OAuth forskellige muligheder
- Billedhåndtering

---

<sup>1</sup><https://github.com/SuneKR/obliGentle/blob/main/Efter%20aflevering/efterAFlevering.md>

- Tag profil direkte fra enheden
- Muligheder for sprog
- Del opgaver med andre profiler
- Email påmindelse
- Mulighed for at vælge design temaer
- Mulighed for at oprette andre slags opgaver med dets egne regler
- Ai som styrer notifikationer
- AI som tilføjer flere opgavetyper for en
- Stemme til AI

## Kapitel 6

# Konklusion

Det har været et lærerigt projekt og selvom jeg er ret frustreret over at aflevere noget, som ikke helt virker, som det skal, så er jeg egentlig ok tilfreds med meget af koden og jeg tror med lidt finpudsning, så kan det blive et godt projekt.

Da jeg skrev case, kravspecifikationen og problemformuleringen var tanken, at jeg ville vælge noget, som var realiserbart og med mulighed for at bygge videre på det.

Den helt store fejl, har dog været at vælge at bygge frontend'en i React Native. Det var det forkerte valg. Som nævnt tidligere, er det ikke fordi, at jeg ikke har fået noget ud af det, men det har ikke været tidseffekt og har efterladt mig en situation, hvor jeg fået en del ud af det selv, men som ikke gav mig optimale betingelser for at demonstrere min egen formåen.

Angående proces var jeg egentlig ok tilfreds, men man kan selvfølgelig ikke være helt tilfreds med den, når det braser sammen i sidste øjeblik. Det har været meget hektisk her til sidst og den sidste del af processen har også været meget frustrerende.

Men jeg synes egentlig, at jeg har fået noget fornuftigt ud af det i alt roden og produktet, på trods af manglende finjusteringer, kan godt noget, ligesom jeg også er godt tilfreds med mit skriftlige produkt, omend den del også har holdt hårdt.

Hvis jeg slut skal fremhæve to ting, så er det mine metodiske overvejelser i metodeafsnittet og implementeringen af i min arkitektur. Det er selvfølgelig tydeligere tidligere i processen, men i det er der og skaber en god bund for, hvad der ellers er usleppent projekt.

# Bilag A

## Literaturliste

Disse er formentlig ikke dækkende for alle henvisning. Find flere henvisninger i materialet.

### A.1 Internet ressourcer

- [Agilemanifesto.org](http://Agilemanifesto.org)
- [Agileit.dk](http://Agileit.dk)
- [Agilealliance.org](http://Agilealliance.org)
- [atlassian.com](http://atlassian.com)
- [back4app.com](http://back4app.com)
- [baeldung.com](http://baeldung.com)
- [code.visualstudio.com](http://code.visualstudio.com)
- [conda-forge.org](http://conda-forge.org)
- [christophergs.com](http://christophergs.com)
- [codevoweb.com](http://codevoweb.com)
- [colorhexa.com](http://colorhexa.com)
- [ctan.org](http://ctan.org)
- [developer.android.com](http://developer.android.com)
- [docs.parseplatform.org](http://docs.parseplatform.org)
- [fastapi.tiangolo.com](http://fastapi.tiangolo.com)
- [flaticon.com](http://flaticon.com)
- [devwithdave.co.uk](http://devwithdave.co.uk)
- [geeksforgeeks.org](http://geeksforgeeks.org)
- [github.com](http://github.com)
- [howtogreek.org](http://howtogreek.org)
- [ionicframework.com](http://ionicframework.com)



- [npmjs.com](https://npmjs.com)
- [medium.com](https://medium.com)
- [merriam-webster.com](https://merriam-webster.com)
- [mongodb.com](https://mongodb.com)
- [multimediedesigneren.dk](https://multimediedesigneren.dk)
- [nodejs.org](https://nodejs.org)
- [ordbogen.com](https://ordbogen.com)
- [ordnet.dk](https://ordnet.dk)
- [overleaf.com](https://overleaf.com)
- [panabee.com](https://panabee.com)
- [pandoc.org](https://pandoc.org)
- [purepng.com](https://purepng.com)
- [pythontutorial.net](https://pythontutorial.net)
- [pypi.org](https://pypi.org)
- [reactnative.dev](https://reactnative.dev)
- [scrum.org](https://scrum.org)
- [scrumguides.org](https://scrumguides.org)
- [shortcut.com](https://shortcut.com)
- [stackexchange.com](https://stackexchange.com)
- [thefreedictionary.com](https://thefreedictionary.com)
- [thesaurus.com](https://thesaurus.com)
- [realpython.com](https://realpython.com)
- [vuejs.org](https://vuejs.org)
- [whois.domaintools.com](https://whois.domaintools.com)
- [wikipedia.org](https://wikipedia.org)

# Bilag B

## Logbog

Dette er min logbog og inkluderer også min daglige ajourføring. Logbogen er skrevet i en uformel tone og jeg har i høj grad også selv brugt logbogen til at reflektere.

De er navngivet efter [årstal][måned][dato]

### B.1 231113

Brugte dagen på at blive introduceret til svendeprøve samt at lave case og problemformulering. Samt forberede meget af det andet materiale til godkendelse.

Brugte noget tid på at komme på et navn. Jeg endte med at vælge "obligentle", som en sammentrækning af "Oblige/Obligation" og "Gentle", hvilket flugter ret godt med, hvad jeg gerne vil have programmet til at kommunikere, at det er en nænsom måde at få ordnet ens forpligtelser. Jeg burde teste navnet, men nu må vi se, om det er noget, som jeg får tid til. Desuden er både obligentle.com og obligentle.dk ledigt i skrivende stund, hvilket også er godt tegn. Igennem whois.domaintools.com kan jeg også konstatere, at det også er tilfældet for: • Net • Org • Info • Biz • Us

Jeg kommer ikke til at købe så mange domæner, men det betyder nok, at jeg ikke bruger et etableret firmas navn.

Jeg lavede et første udkast til case og problemformulering og fik det sendt til Kenneth og Rasmus. Umiddelbart blev det godkendt, men Kenneth, som mine primære vejleder, ville gerne lige mødes og snakke om det. Umiddelbart advarer han om, at jeg kan have slået et for stort brød om. Personligt, tror jeg i højere grad, at det handler om, at jeg ikke får beskrevet noget for åbent, men nu må vi se.

Jeg mangler at beskrive, hvorledes jeg vil håndtere projektstyringen. Det har jeg i sinde at beskrive i morgen, men det bliver noget morgen-opsummering med nogle faste spørgsmål.

Jeg fik også aftalt korrektur læsning med min kusine, men der mangler stadig lidt detaljer.

### B.2 231114

Mødtes med vejleder i morges om projektet. På den ene side, bekræftede Kenneth, at der var kød nok på projektet, men senere på dagen efterlyste han noget mere teknisk

dybde og en ting som 2fa eller at kunne sende mails. Mit take-away er, at jeg i så fald skal vælge 2-3 af de ekstra features, som er "dybere" teknisk - for ikke blot at vise det, men for at vise det på en overbevisende måde. Jeg har oprettet en liste i OneNote med feature-ideer, hvor jeg har skrevet en note om det - det er alligevel lettere end at designe frontend. Mockup farvevalget blev også diskuteret og jeg var alene i min entusiasme. Jeg må have en blind mands sans for farver.

Jeg lavede tidsplanen, mock-up, krav spec og rettede gårsdagens arbejde. Mock-up blev ret detaljeret. Det var en hjælp for mig.

Kenneth anbefalede, at jeg lavede et flowchart. Det havde jeg egentlig håbet at slippe for, for jeg synes ikke, at de er særlig anvendelige og tager lang tid at lave, men han har en pointe, så det vil jeg lave i morgen og så komme det senere i gang med at programmere. Muligvis giver det mening, eventuelt sammen med nogle database datamodeller.

I forbindelse med tidsplanen designede jeg også processen: • Dagligt laver jeg en ajourføring, som tager udgangspunkt i en række faste spørgsmål • Onsdag formiddag planlægger jeg sprint og skriver det ind i et kanban board • Tirsdag eftermiddag evaluerer jeg det foregående sprint med særligt fokus på proces og tidsplan Dette lyder for meget som en scrum/agile, men det er det jo ikke helt. Så derfor burde jeg overveje muligvis at ændre ordvalget lidt - ligesom jeg har gjort med den daglige ajourføring.

## B.3 231115

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Flowchart, Oplader\*, nej

Hvad er dagens prioriteter? Flowchart & Kanban\*\*

(\*) Jeg mangler min oplader. Nhat og Morten sagde, at det var til at låne. Ellers må jeg tage hjem og hente min egen. Så det var til at ordne. Det ordnede sig ret nemt.

(\*\*) Foruden at lave sprintopgaver, vil jeg også afsøge hvilket kanban board er bedst. Dagens prioriteter er i rækkefølge: • Flowchart • Afsøge bedste Kanban løsninger • Planlægge sprint • Begynde at kode Det er hensigt at komme til at kode i dag. Med fordel kunne man også beskrive datamodeller, men det bliver kun, hvis der er tid til det, fordi det kan også gøres senere.

Nhat foreslog i øvrigt, at jeg brugte Flutter eller React native i stedet for Ionic. Det må afsøges, men lyder som en god idé.

Jeg kodede flowchart'et i LaTeX. Det oplever jeg personligt som at være lettere, særligt fordi jeg synes flowcharts er uforholdsvis besværlige i forhold til udbytte, så det hjælper altid, at man får lov at programmere lidt. Jeg skulle lige skifte bekendtskab med TikZ igen og havde lidt problemer med at få dvipnames fra xcolor-pakken til at spille sammen med TikZ-pakken, men endte med en løsning (umiddelbart er det lavet så smart, at rækkefølgen af importerne har betydning i denne sammenhæng og det var vigtigt, at xcolors skulle indlæses først). Jeg valgte at fokusere på login processen, som blev ret minutiøs. For "aftale", "projekter" og "tjanser" fokuserede jeg meget på processerne og ikke så meget de enkelte knapper. Det kommer jeg dog til at gøre senere. Angående "indstillinger", "profil" og "kalender" blev de nærmest kun nævnt. Jeg satser dog på at lave et opdateret flowchart senere, når der er mere programmeret

Desuden fandt jeg fodnote problem i tabellerne, navnlig at de ikke blev lavet. Jeg fiksede det ved at importere tabelfoodnote. Den lavede dog 3 dublet fodnoter, men jeg besluttede, at det var mindre vigtigt at fikse nu og det var vigtigere at få mit materiale godkendt.

Jeg endte med at vælge shortcut som Kanban board. Det brugte vi også på H3. Jeg har i mellemtiden brugt trello hjemme, men jeg synes ikke helt, at den kunne det samme, så derfor føles valget rigtigt - endvidere tillader det mig at sammenligne 1-1 med vores forhenværende projekt, hvilket var rart, da jeg satte det op. Det kan ikke undgå at slå mig, at det er lidt underligt at bruge et produktivitsredskab til et program, som også er lidt en kritik af produktivitsredskaber. Men det giver jo mening i den forstand, at shortcut og andre lignende værktøjer virker for godt som arbejdsredskaber, men jeg ikke gider at opgøre min fritid i produktivits sprints og lignende.

Jeg fik oprettet en masse opgaver. Helt sikkert ikke alle, men nok til, at der er en del at tage fat i. Det fungerer i hvert fald allerede godt og giver en del overblik.

Jeg gik i gang med at oprette projektet og tog den beslutning, at jeg vil oprette et helt nyt python miljø i anconda, som jeg vil bruge til at under svendepreven. Jeg kaldte miljøet for "svendeprove" og oprettede det som python 3.11.0. Føler ikke, at det var strengt nødvendigt, men at det giver mig en rene oplevelse og på sigt gør, at jeg muligvis kan undgå bøvl, fordi jeg er ikke er den bedste til at gøre rent eller holde orden i mine python miljøer. Jeg oprettede projektet og forbandt det til github. Jeg besluttede mig for, at jeg vil forsøge at bruge "Comment first", altså man skriver alle kommentarer og måske lidt pseudo kode, før man begynder at kode - i modsætning til at gøre det efter. Jeg er spændt på, hvor længe jeg kan holde det, men nu forsøger jeg. Det har også den fordel, at selvom jeg skal vente på godkendelse, før jeg egentlig må kode, så er der ingen regler om at skulle vente med at kommentere på ting. Jeg gik i gang med at lave "comment first" af main.py filen - det var ikke så meget, men det hjalp lidt.

Det sidste af dagen gik med at researche 2FA og bruge best practice og fandt et python modul, FastAPI-users, som understøttede både 2FA og hjalp med brugerhåndtering. Jeg fandt også en artikel til at implementere 2FA selv. På den ene side vil jeg godt demonstrere, at jeg selv kan finde ud af det. På den anden side, så er det formentlig sikkerhedsmæssigt mere forsvarligt og jeg kan formentlig finde nogle andre steder at vise noget teknisk formåen.

I dag var en lang dag, så i morgen har jeg planlagt at korte lidt af dagen af til gengæld. Jeg er den heldige situation, at jeg burde kunne fokusere på lige at få lavet API'en i morgen, så det kunne jo være mægtigt rart. Alt i alt en god og produktiv dag, hvis jeg selv skal sige det.

## B.4 231116

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, Grøntlys(\*), Nej

Hvad er dagens prioriteter? Metode og Datamodels

(\*) Mangler grønt lys med at gå i gang.

Jeg tager det roligere i dag, har overarbejdet meget i starten af ugen, men kigger lidt på metodeafsnittet. Hvis jeg får grønt lys, så skal jeg dog i gang og få lavet mest

muligt. Jeg orkede ikke lige at rette gårsdagens log, da jeg kiggede på det. Det gør jeg senere i dag, men lader den del af punktet stå som ikke-udfyldt indtil jeg har udfyldt det.

Jeg har besluttet mig for at prøve FastAPI-user og se om det ikke giver mening i forhold til sikkerhed. Det er sikrere og om det viser nok teknisk dybde, må vi se, men ellers får jeg nok mere tid. Jeg bør lave en liste over alle anvendte moduler, som jeg bruger.

I samme henseende besluttede jeg mig for, at jeg vil prøve at bruge flutter på mandag. Så har jeg en hel dag til at kigge på det, men kan stadig nå at lave en helt rudimentær front i iconic, hvis det ikke dur, om tirsdagen. Jeg ved også, hvor meget jeg mangler inden review, så jeg bruger dagen på noget, hvis nødvendigt. Det kunne være fedt at nå så langt med de andre ting inden da som muligt, da jeg i så fald ville udvikle i flutter og eventuelt også komme længere end tidsplanen og virkelig for fordybet mig i det.

Dagen gik med at kigge på metodeafsnittet, men det var ærligt talt ikke min mest produktive dag, men jeg har også lagt mig i selen de 3 første dage, så derfor tænker jeg egentlig, at det går.

## B.5 231117

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, nej

Hvad er dagens prioriteter? API

Jeg startede dagen med at rydde op i kanban boardet og sørget for, at det var opdateret. Jeg endte dog også med at smide en del ekstra opgaver ind i backloggen udenfor iteration. Mange af dem er opgaver, som skal laves på et tidspunkt, men som jeg ikke nødvendigvis vil presse ind i dette sprint. Jeg endte også med at justere mine "space" på shortcut for at optimere min effektivitet. Jeg arbejdede primært på API'en og fik også lavet noget, men blev afbrudt af noget familiemæssigt og kom ikke tilbage til det senere på dagen. Jeg fik dog koordineret rette-muligheder sammen med Kusine og vi aftalte, at jeg sendte til hende i word format og ventede med at opsætte det i LaTeX til efter hun har haft mulighed for at rette det.

## B.6 231120

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, ja\*

Hvad er dagens prioriteter? API og frontend

(\*) Jeg kan se på mit arbejdsflow sidste uge, at jeg arbejdede en del over, hvilket gik ud over torsdag og fredag, hvor jeg simpelthen var mindre oplagt og effektiv. Det må jeg tage med retrospektivt i morgen.

Snakkede med Kenneth, som anbefalede mod flutter og anden ukendt frontend, samt at skifte MongoDB ud med firebase. Over weekenden har jeg tænkt over det med teknisk dybde og lave noget med AI. Jeg besluttede mig for, at jeg laver app'en færdig

som beskrevet, og så bagefter forsøger at lave noget AI ovenpå. Det ville gøre, at jeg endte med at have et færdigt produkt, som jeg lavede det mere usikre ovenpå. Det bærer faren med, at jeg muligvis skal lave noget færdigt, men på den anden side, så er det ikke nødvendigvis entydigt skidt og jeg tror ikke, at det bliver helt grundlæggende, at jeg må lave noget om. Kenneths kritik mod flutter handler om, at jeg bør fokusere på, hvad jeg kan. Men Nhats kritik er også gyldig. Jeg laver API'en færdig, så overvejer jeg det lige igen.

Jeg arbejdede også med min anaconda environment, som virkede, men meldte fejl på pakkerne. Det handlede om, at jeg skulle, installerede dem ved at bruge conda-forge kanalen(<https://conda-forge.org/#about>), som er et "community-drevet "repository", som tilbyder mange pakker, som ikke er anaconda default "repositories". Jeg installerede dem først bare direkte, men endte med at tilføje conda-forge til environmentets channels, således det bare skulle virke fremover. Hans kritik mod flutter handler om, at jeg bør fokusere på, hvad jeg kan. Men Nhats kritik er også rigtigt. Laver api'en færdig, også kigger jeg på det.

Endte med at få lavet datamodellerne færdige, men må indrømme, at mit "comment first"princip snarere blevet en iterativ udvidelse af skiftevis kode og kommentarer, men det hjælper mig til at skrive bedre kode og kommentarer, så jeg holder fast i den.

Jeg endte med at sidde lidt fast i, hvordan jeg bedst muligt kunne skrive hjælpefunktionerne toDict funktionerne. Endte med ikke at ville sidde fast i det og ikke fokusere på at skrive den optimalt i forhold til "inheritance", men bare skrive den grimt.

Jeg endte med også at være lidt mentalt brugt, imens jeg skrev mine routers og endte med at tage en pause. Oplevede at jeg gik i gang med at skrive det dumt i stedet for at holde mig til SOLID principperne og det gik op for mig, at hvis jeg ikke passede på, så endte jeg formentlig med at skulle skrive det helt om. Jeg har generelt set oplevet, at jeg var ret brugt og måtte tvinge mig selv i gang.

## B.7 231121

Jeg prøvede flere forskellige ting for at komme i gang og få lavet noget, men for at være ærlig, så var hovedet ikke bare til det i dag. Jeg prøvede at koncentrere mig om forskellige opgaver, om at skrive, om at kode, prøvede at tage pauser og prøvede at tvinge mig selv. Det gav ikke alverden og jeg besluttede mig til sidste for, at det ikke gav alverden og jeg måtte prøve igen om aftenen eller tage fat igen i morgen. Uanset om det fungerer i aften, så må procesevalueringen udskydes til i morgen.

Det fungerede heller ikke om aftenen.

## B.8 231122

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Rensket? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Gårsdagen\*, nej, gårsdagen\*

Hvad er dagens prioriteter? App skitse\*\*, & retrospektiv

(\*) Det åbenlyse svar er, at en dag inddrages fra weekenden og alt udskydes med en dag. Jeg tror, at jeg kørte mig selv for hårdt i sidste uge og weekenden og havde

brug for en hviledag. Men må lige tage mere under retrospektivet senere i dag. Det her ikke optimalt, men til gengæld føler jeg mig mere ovenpå i dag.

(\*\*) lave API'en færdig, lav en rudimentær frontend og få dem til at spille sammen. Hvis der er tid, så begynd at implementere fastapi-users

Jeg stødte ind i nogle udfordringer med at lave klasser med inheritance i routerne. Jeg fejlsøgte ved at lave en metode som fungerer uden for en klasse, men det drillede stadig at få oprettet det rigtigt med inheritance. Jeg endte med at bruge en del tid på at research'e problematikken og endte med at udskyde evalueringen.

## B.9 231123

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Routers\*,routers\*, ja\*\*

Hvad er dagens prioriteter? App skitse\*\*

(\*) Problemet med nedarving mellem de forskellige "router"klasser, skal der styr på i dag. Føler, at jeg er tæt på at finde og løse problemet og derfor lader jeg det tage topprioritet i dag.

(\*\*) Jeg vil have styr på "appskitzen", en rudimentær frontend koblet på API. Det blokerer lidt processen, så derfor udskyder jeg lige procesevalueringen og planlægningen af næste forløb indtil der er styr på den.

Jeg føler mig egentlig ok med det, selvom selvfølgelig er irriterende at have tekniske problemer. Jeg tror også, at jeg skal huske, at jeg har en ekstra dag i denne uge - så planen sikkert ikke er helt så skidt ud, som jeg først var bange for.

Fik styr på "router"ne og fik nedarving til at virke, sådan da. For POST og PUT nedarvingen, blev jeg nødt til at skrive metoderne fra bunden for alle klasser. Det vil jeg fikse senere, lige nu har jeg brugt nok tid på det. Ellers fungerer det godt. Jeg har tilføjet endnu en "read"metode, som finder alle af en gruppe, som bruges til alle "tasks"af en type. Lige nu bruger jeg router interfacen både, som interface og som router for task. Det skal jeg også gøre noget ved, men python lader mig gøre det, så det er ok for nu, indtil jeg lige får styr på nogle andre ting. Begge ting er tilføjet til backloggen. Lige nu sætter jeg databasen i routerne. Det kommer til at ændre sig, så lige nu er det bare en midlertidig løsning.

Jeg testede også API'en og den virkede. Jeg venter dog med at dokumentere det, indtil jeg har fikset de to overstående problemer, fordi så har jeg brug for at test'e det igen.

Jeg besluttede mig for, at det reactNative i stedet for iconic. Det besværlige er Android Studio og ikke iconic kontra reactNative, så derfor kunne jeg lige så godt køre med det. Jeg installerede reactNative og opdaterede android studio, hvilket tog en krig, så endte med at evaluere sidste sprint og planlægge næste (se Nederste på siden). Jeg fik gennemført installation, opdateringer og initieret en app. Desværre havde jeg problemer med gradle. Jeg forsøgte at fikse på selv og ved at bruge reactNative doktoren. Det hjalp dog ikke og da react-native doktoren ville have en genstart, opgav jeg for i aften.

Havde også brugt tid på at studerede hosting muligheder og købte et webhotel ved AzeHosting, som jeg vil bruge til API og web-applikationen. Til databasen vil jeg lige se, om der er mulighed for også at ligge den der, det lignede det ikke, men ellers påtænker jeg at bruge mongodb atlas.

## B.10 231124

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, gradle\*, nej

Hvad er dagens prioriteter? Frontend

(\*) Gradle drillede i går. Det skal jeg lige fikse

Det var lidt optimistisk at tænke, at man bare kunne lave en rudimentær frontend. Mere realistisk er det nok, at man bruger lang tid til at sætte frontend op og få det til at fungere, men selve programmeringen er rimelig overkommelig. Så målet er en funktionel frontend, så kan jeg gøre den pæn senere.

Det tog en del tid at arbejde i gradle og android studio, men det gav ikke megen resultat. Fandt dog ud af, at den åbenbart havde installeret to versioner, så jeg slettede alt android studio fra computeren for kun at installere den nyeste version. Det var godt nok helt sært med Android Studio. Det fyldte over 150 GB i min appData mappe og en af versionerne kunne kun afinstalleres derinde fra. Uanset om det giver pote, så var det nok ret fornuftigt at få rydde op i det. Det får mig til at tænke på, om alle de problemer, som jeg har haft med Android Studio og lagt til grund for min dybfølte ringeagt af det produkt, handler om en installationsfejl, som bare har gjort det unødvendigt besværligt og langsomt. Jeg installerede den også på mit data drev, så det kan fylde mere. Dog kunne det ikke længere starte og fandt endnu en gemt mappe i min APP data, hvor den også fyldte. Det var heldigvis løst ved at slette en txt fil i app-data mappen, som jo er det mest naturlige for et program at genere. Efterfølgende skulle jeg installere SDK'erne igen.

Kenneth anbefalede, at jeg brugte expo i stedet for CLI med android studio. Jeg fik det umiddelbart til at virke, så det tror jeg er vejen fremad. Det endte med at have taget dagen. Jeg fik også sat tablet op, men havde ikke tid til at teste, om expo også kunne gøre på den. Kenneth anbefalede at bruge Expo for at undgå at skulle bruge

## B.11 231127

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, nej

Hvad er dagens prioriteter? App

I dag skal jeg have APP'en op og helst også fastAPI-users. Weekendten gik mest med at holde fri og der blev udviklet meget ekstra, desværre. Jeg fik dog undersøgt lidt. Der er ingen pointe i at slå mig selv i hovedet over det. Det er, hvad det det.

Jeg fik tablet til at tage virke med expo.

Jeg fik også den web applikation til at køre og gik i gang med at skrive en rudimentær frontend og jeg fik sat nogle routes op.

Gik i gang med at få den til at fetch'e data, men blev ikke færdig med det. Det var sgu overmodigt bare at tage fat i react native, når jeg ikke rigtig kendte det. Det er jo til at lære, men det hele tager bare lidt ekstra tid. Jeg må indrømme, at jeg ikke er sikker på, at iconic havde været bedre, men stadig er det værd at have med.

Lavede i øvrigt en aftale med Nhat om, at han vil hjælpe mig i gang med at programmere noget AI på onsdag.



## B.12 231128

Denne opfølgning er skrevet dagen efter, d. 29/11-23

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, arbejdsrytme\*, Evaluering\*\*

Hvad er dagens prioriteter? App

(\*) Jeg endte med ikke at få lavet så meget i løbet af dagen, men så til gengæld at sidde og programmere til 3 i nat. Det er sgu lidt træls i forhold til døgnrytme, men nu er der 9,5 dag tilbage, så tror jeg, at jeg med fordel bare kan give lidt kast på døgnrytmen og bare arbejde mest muligt. Det er alligevel mere en test af, hvor meget jeg kan lave på 4 uger, end en simulering af en arbejdsmarkedssituation.

(\*\*) Evalueringen planlægningen i morgen. Det skyldes, at dette er skrevet dagen efter, så derfor ligges evalueringen, hvor den rent faktisk laves. Nu er det sket begge gange, så det skal forandres.

Routing skulle laves lidt om og jeg endte med at tilføje TaskScreen og en HomeScreen. Jeg havde i første omgang brugt camel case (eller lower camel case, om man vil), men react ville have mig til at bruge pascal case, så det lavede jeg dem om til. Jeg lavede også en index fil for "screens"mappe, således jeg lettere kunne hente.

Jeg laver også nogle components, hvor jeg skrev DataFetcher.js, som jeg bruger til at hente og vise dagen for alle de forskellige slags task. Lavede det således, at jeg kunne DataFetcher i hver screen med en string, som modererer api kaldet. Lige nu kører alt igennem DataFetcher. Det er er ikke realistisk, at det kan komme til det. Lavede også en styles fil. Jeg lavede ikke en index fil til components mappen, fordi jeg synes, at components er lidt misvisende. Så jeg vil lige finde på et bedre navn.

Jeg stødte ind i lidt problemer med CORS (Cross origin resource sharing), hvilket jeg så måtte implementere i API'en. Det var heldigvis lige til.

For at tjekke, om jeg kunne få vist en task's forskellige egenskaber, så lavede jeg et tooltip, som viste task'ens type og beskrivelse, når jeg holdt over tasken. Det gav nogle udfordringer, da det tooltip, som jeg brugte fra material-ui krævede, at "wrapper"det eller medsender en reference og "prop". Det viste sig at kunne løses med at ligge det hele ind i <div>, hvilket var meget nemt, men det tog mig lidt tid at komme dertil. Primært fordi Material-ui foreslog nogle andre og mere besværlige løsninger(<https://mui.com/material-ui/react-tooltip/#main-content>)(<https://mui.com/material-ui/guides/composition/#wrapping-components>). Jeg er dog i tvivl, om det var lidt spild af tid. Jeg retfærdiggjorde det, at det ikke skadede at få styr på wrappers, men med løsningen, som det endte med at blive, altså bare sætte den i <div>, tror jeg muligvis bare er lidt specifik for denne pakke, at den er skrevet til at have brug for det. Jeg tror med fordel, at når jeg arbejder til sent, så skal jeg sætte nogle klare mål. Fordi jeg har nået en del og det er sgu fint, hvad jeg har lavet, men det her med tooltip'et er nok lidt spildt arbejde og havde jeg haft en klar målsætning, når jeg arbejder sent, så kunne jeg nok have undgået det.

## B.13 231129

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja  
Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Evaluering\*, Træt\*\*, evaluering\*

Hvad er dagens prioriteter? App & AI\*\*\*

(\*) Evalueringen laves i dag og derefter planlægges næste sprint. Evalueringen er begge gange blevet udskudt og lagt umiddelbart før planlægningen. Det skal også evalueres.

(\*\*) Jeg kan godt mærke, at jeg fået 4 timers søvn. Jeg havde tænkt mig at arbejde sent i dag, da det passer med, at jeg har en aftale i aften i nærheden af skolen, men lad os se, om jeg holder til det.

(\*\*\*) Jeg har en rudimentær frontend, som kan hente data fra API'en. Nu skal den også kunne alt det andet og være til at navigere i. Design og brugeroplevelse er dog ikke med i dagens mål. Klokkeren 1 vil Nhat komme forbi og hjælpe mig i gang med AI. Det er vigtigt, at jeg lige er klar til det. Jeg skal ikke nødvendigvis gå i gang med at udvikle der - jeg er der for at få noget viden om, hvordan jeg går i gang med det, når jeg kommer dertil.

Jeg foretog evalueringen og laver en nu tidsplan. Dokumentationsdage: fredag, søndag, tirsdag Programmeringsdage: torsdag, lørdag, mandag, onsdag

Dokumentationsdage skal jeg nå at skrive 10 sider minimum. Programmeringsdage skal jeg nå en af følgende ting: • Funktionel APP og brugerhåndteringer • AI funktionalitet • Design og brugeroplevelse • Lagt på nettet og test på tablet

Det var det vigtigste. Jeg fik også lavet planlægningen. Som jeg også har noteret i procesevalueringen, så har jeg simpelthen glemt at bruge git. Det er sgu ikke så smart. For at gøre det lettere, har jeg lavet et nyt repository, hvor både front- og backend er i samme mappe - skal dog stadig huske at få det gjort. Det tog lidt ekstra tid, fordi jeg kom til at dumme mig, da jeg lavede et samlet repository-

Nhat var syg, så intet møde om AI. Jeg må lige selv læse op på det. Det er godt nok lidt noget rod det her. Jeg må hitte ud af det, men jeg tror, at jeg har ladet mig vejlede til at gøre det projekt væsentligt mere kompliceret og besværligt end nødvendigt. Jeg holder mig til planen og det skal nok gå, men jeg er blevet talt langt væk fra det originale projekt, som trods alt er blevet godkendt og man må kunne forvente, at det kan bestå uden alle de her ekstra ting, som hele tiden kommer ind fra højre.

## B.14 231130

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, nej

Hvad er dagens prioriteter? Operationer

Det var egentlig meningen, at jeg ville tage det mest af dagen fri, fordi der ligesom er så kort tid tilbage, at jeg ikke får tid til at se familie eller alverden venner inden afleveringen på fredag. Familien endte dog med at måtte aflyse grundet vejret, men tog det stadig afslappet.

Jeg implementerede lidt operationer i app'en, men det var ikke alverden. Fordi jeg gik lidt i stå i hvilken måde, som jeg helst ville gøre det på. Jeg endte med at lave noget research, men også bare få holdt fri og hvilet, således at jeg kan komme i gennem den næste uge.

Jeg blev så ikke færdig med APP og brugerhåndtering i dag, så det må jeg gøre færdig i morgen efter jeg har skrevet 10 sider (hvilket jeg formentlig næsten har i forskellige dokumenter, som lige skal samles og revideres, før de kan sendes til vurdering).

## B.15 231201

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, nej

Hvad er dagens prioriteter? Skriv 10 sider

Jeg startede med at bruge pandoc til at konvertere min latex til word, således jeg nemmere kunne konvertere og sende, hvad jeg allerede har lavet til retning, som jeg har lovet min korrekfløse. Jeg kom dog overraskende hurtigt op i nærheden af de 10 sider med hvad jeg allerede havde lavet, men besluttede mig for at forsøge at få lavet så mange sider, som jeg lige kunne overskue nu her, sådan at jeg have det mindre bekymring og der blot var bedre tid til korrekturlæsningen og for mig til at genoverveje det.

Jeg mødtes desuden med Kenneth og talte lidt forventninger og lignende. Jeg følte det projekt, som jeg havde lavet, var blevet talt en del ned i det omfang og vi fik talt ind på, hvad der menes. Han kan selvfølgelig ikke sige noget specifikt, men jeg fornemmelsen af, at jeg havde ret i min vurdering i, at det sådan set er fint projekt, men det selvfølgelig mangler noget større teknisk, hvis det skal ligge helt op i den gode ende. Så det er noget mere til at forholde sig til og er jo mere end muligt, men det er godt nok, at det ikke står og falder med AI implementeringen, for nu overvejer jeg bare at lave noget andet end AI til at give den tekniske tyngde, som givet er nødvendig.

Jeg havde sidst på eftermiddagen et kaffemøde med en holdkammerat og endte med at få lavet 21 sider på det tidspunkt og mangler stadig logbøgerne, som lige skal skimtes og flyttes til et andet format. Jeg føler, at jeg godt kan presse lidt flere sider ud i dag. Selv bare nogle mindre og kedelige ting, så som læsevejledning og indledninger, ville være rart at få styr på. Meget af det, som jeg skriver, er sammenfatninger af ting af kladder, som skulle skrives færdig. De er færdige, så derfor ville det være fedt at få lidt mere, men natten er ung, så mon ikke, at det bliver muligt. Jeg mangler også at skrive om mine anvendte Python moduler og noget om JavaScript og mine anvendte JavaScript moduler.

Jeg skrev logbogen og procesevalueringen over, samt fik skrevet den procesvejledningen og rettet det lidt til. Det endte med at være 6 sider til produktrapporten, 13 sider til produktrapporten og 15 siders log- og evalueringsbog. Hvilket jeg er ret godt tilfreds med.

## B.16 231202 & 231203

Fordi det er weekend, laver jeg ikke daglig ajourføring, men opsummerer blot begge dage sammenlagt.

Begge dage gik med at arbejde på operationer.

Påopfordring fra min holdkammerat installerede jeg også flere react native extensions. Jeg tog udgangspunkt i anbefalinger fra denne side: <https://reactnativecentral.com/top-vscode-extensions-react-native/> Jeg har også godt vidst, at der var fordele ved at in-

stallere flere extensions eller muligvis bruge en IDE, sådet var også bare for at få gjort.

Jeg var blevet træt af, at den hele tiden åbnede nye faneblade, så jeg lavede et nyt script i package.json og kaldte det "webnb".

Jeg endte med at være glad for syntaksen i "Parse" og skrive min kode op i mod det, men det var en fejl. Da jeg endte med at finde ud af, at det ikke virkede med parse servere. Der var gået en del tid med at skrive koden og forsøge at fikse diverse problemer. Det var ikke kun med parse, som der var problemer, så derfor var det svært at finde den egentlige fejl.

Fik også lavet flere andre dele, så som navigation, tab menu, screens og diverse designing. Flere af disse ting, kan formentlig kan godt reddes. Men det må tiden vise.

## B.17 231204

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, ja\*

Hvad er dagens prioriteter? Operationer

(\*) I weekenden arbejde jeg meget målrettet uden at tænke så meget over proces og deslige. Det fortsætter jeg med. Jeg har målene, som jeg satte mig selv i fredags, men arbejder eksempelvis ikke med kanban bræt - det må jeg få styr på i evalueringen.

Jeg fik lavet alle operationerne til task-klassen, men ikke til de andre endnu. Jeg fik også lavet hele backend'en med fastapi-users inklusiv oauth2, men der mangler en frontend login og oauth2 mangler formentlig lidt opsætning.

Det meste af FastAPI-users er baseret på deres kode eksempler: <https://fastapi-users.github.io/fastapi-users/12.1/configuration/full-example/> <https://fastapi-users.github.io/fastapi-users/12.1/configuration/oauth/>

Hvilket skal beskrives i rapporten. Alternativt hvis der er tid, så kunne man også med fordel skrive det om, så det passer mere ind i koden.

Det er lige nu ret rodet og koden for håndteringen af task er fordelt på app.js og TaskModel, hvilket skal ryddes og gøres meget mere SOLID.

Der er også en del ting, som muligvis kan bruges.

## B.18 231205

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, Evaluering\*

Hvad er dagens prioriteter? Skrive 10+ sider

(\*) Egentlig skulle evalueringen være i dag, men de sidste gange har jeg flyttet det til onsdag. Det ville give mening at gøre det samme i dag, men desværre på den anden side, så er i dag sidste dag, at der kan blive sendt til korrektur.

Jeg forsøger at skrive så meget muligt i dag, overskuden tid vil jeg kigge på en bruger frontend og sidst vil jeg lave en evaluering, så må vi se, hvad jeg kan nå.

Jeg kommer til at skrive nogle ting i dag, som konkluderer på hele projektet, selvom dele ikke er færdigt. Det kan selvfølgelig kun være udkast, men jeg er ret sikker på, hvad jeg vil konkludere om 2 dage.

Jeg fik skrevet 14 sider. Hovedsageligt på processrapporten. Jeg tror, at den er ved at være klar.

Jeg arbejdede på API'en, men stødte ind i et problem med at uvicorn op og køre. Jeg endte med at sove på, men det er potentielt ret skidt og skal fikses hurtigst muligt.

## B.19 231206

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, miljøfejl\*, slutevaluering\*\*

Hvad er dagens prioriteter? Program færdig

(\*) Der er pludselig kommet en fejl i python miljøet, hvor den ikke genkender alle mine imports og derfor ikke vil køre min backend. Jeg opdagede fejlen i går, efter jeg var færdig med at skrive. Forsøgte at løse det, men jeg var simpelthen for træt. Nu må jeg løse det i dag. Hvis jeg ikke får styr på det, så er det potentielt ret skidt, men der er ingen grund til at stresse over det.

(\*\*) Jeg skal have lavet slut evalueringen i dag. Jeg vil blot have styr på programmet først. Men laver det seneste ved aftensmadstid.

Jeg prøvede forskellige ting, men endte med at gå i gang med at afinstallere nogle gamle python miljøer, som lignede roede emd det. Før jeg kunne få det tjekket røg mit net dog, som jeg lige måtte fikse. Jeg endte med at installere et nyt miljø, men det vidste sig ikke at være fejlen - hvert fald ikke udelukkende.

Jeg endte også med at skrive oauth ud af router. Efter megen fejlfinding, så endte det med at være mine modeller, som jeg havde skrevet ind i mine app startup forbindelse.

Jeg venter med at skrive de ting ind igen, så nu får jeg styr på brugerfrontenden, så skriver jeg 2faktor autoriserignen ind og derefter skriver jeg det hele sammen. Så genindfører vi det, som vi rent faktisk kan tjekke det.

Jeg aftalte med også med min korrekfleuse, at hun muligvis kunne rette lidt i morgen også, så hvis jeg får tid til det. Nu vil jeg lige tage et kvarters pause, før procesevalueringen og derefter har jeg hele aftenen til at klø på med at få programmet færdigt.

Jeg endte med at sidde og lave controllers, som arvede fra hinanden.

## B.20 231207

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Ja, ja, ja

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Nej, nej, nej

Hvad er dagens prioriteter? Præsentabelt

Det prioriteter lidt mere udpenslet: Programmet skal gøres præsentabelt og let tilgængeligt. Derefter skal jeg ligge det op på nettet og tjekke det på tablet'en. Derefter skal jeg sætte rapport op.

Herefter skal jeg rydde op i kode og afpudse. Hvis der er mere tid, så tager vi det, når vi kommer tættere på.

Jeg arbejder blot fra toppen af, indtil jeg kan mærke, at jeg har brug for at en pause. Så tager jeg hjem, holder en kort pause og vurderer, hvad jeg har tid og prioriteter. Hvis det værste er ude, kan jeg altid skippe at sætte opgaven op i LaTeX og aflevere i word og går det helt skidt, så må jeg undvære at ligge programmet op på nettet.

Jeg mødtes med først Kenneth og derefter Jeanette og lige gennemgå, hvor jeg var. Kenneth nævnte, at jeg først skulle aflevere fredag 23.59 i morgen. Jeg talte også om, hvad jeg realistisk ville kunne justere ved programmet efter aflevering. Det kommer ikke med i rapport vurderingen, så det er mest oplagte er nogle fin pudninger og desgin. Jeg aftalte med Jeanette at holde et oplæg for hende mandag 1230.

Jeg gik i gang med at justere min controllers, men blev lidt distraheret af, at jeg gerne ville undgå i relative stier i mine imports. Det justerede jeg efter dette eksempel (<https://stackoverflow.com/questions/43681091/how-to-use-import-with-absolute-paths-with-expo-and-react-native>) og det virker. Jeg fik også billedet til at virke. Det var dog nogle fjollede prioriteringer og jeg vender tilbage til at fokusere på controllers.

Jeg fik styr på login skærmen og mangler bare resten, men metoden er der. Jeg tog hjem til frokost og holdt derefter en kort pause. Efterfølgende lavede jeg lige et overblik over prioriteterne.

1. Få sammenflettet resten af frontenden 2. Få lagt API'en på nettet og sikre mig app'en kører på tablet 3. Få sat opgaven op i LaTeX og skrevet det sidste 4. Ryd op i kode og gør frontend pænere 5. Sørg for alt er eksporteret, i rette formater og kan afleveres

1 har første prioritet Når den er færdig, så prioritere jeg opgave nummer 2 og 3. Ideelt set, tager jeg det i den rækkefølge, men 3'eren kan potentielt vente til efter aflevering, så hvis jeg er sent på den.

4 og 5 bliver i morgen, hvor pænere frontend delen godt kan vente til efter aflevering - hvis nødvendigt. 5 bliver nødt til at være noget af det sidste, som jeg gør, men bør heller ikke gøres senere end 12, så jeg er sikker på, at jeg har 2 timer til at få styr på det.

## B.21 231208

Daglig ajourføring Holdes hver arbejdsformiddag, helst inden 10. Skriv kun ja, nej eller stikord. Hvis noget skal udbydes, så gør det nede i loggen.

Gårsdagen log: Fuldendt? Tilstrækkelig? Renskrevet? Nej, nej, nej

Arbejdsproces: Opfølgninger? Blokeringer? Forandres? Fejl\*, nej, nej

Hvad er dagens prioriteter? Aflevering\*\*

(\*) Der er to fejl i programmet og den vil ikke hente rigtigt. Den vil ikke fetch'e tasks rigtigt og den henter den forkerte bruger database. Begge ting burde være overkommeligt, men det lykkes mig ikke at fikse i morges. Desuden er der kun for en type task. Alle virker som mindre ting.

(\*\*) Der skal afleveres i dag. Det inkludere rapporterne og at gøre programmet parat.

Jeg når forhåbentligt både at få lave aflevering og rettet de fejl, men prioritere afleveringer, også må jeg skrive mig ud af det og ændrer det inden forsvaret, hvis jeg ikke når det. Udover de fejl, så er der også et par andre ting, som jeg gerne vil lave, men nok ikke når. Det inkludere at få API'en på nettet, applikationen på table't og leget med AI - men det tænker jeg, at jeg kan have med til svendepróven.

Jeg skriver rapporter i LaTeX og jeg har lagt dem i samme programmeringsprojektet, så rapporter burde også komme på GitHub.

Jeg har skrevet til Kenneth, om han har tid til at snakke i dag, sådan han også lige er forberedt på det. Der var travlt indtil sidste minut, fik lov til at udskyde deadline til 15.

## Bilag C

# procesevalueringer

Dette er mine procesevalueringer. Ligesom logbogen er de af forskellige længde og karakter; og er først og fremmest brugt til egen refleksion og derfor muligvis lidt interne.

De er navngivet efter [årstal][måned][dato]

### C.1 231123

Procesevaluering: Den umiddelbart største ting procesmæssigt er, at jeg har ladet mig stresse lidt over, at jeg oplevede en kritik af, at projektet manglede teknisk dybde. Det var planen at lave produktet og skrive det meste af procesrapporten, så meget af den jeg kunne, i de 2 første uger og bagefter bygge en eller flere fede features på. Det er nu blevet til, at den feature er noget AI, og jeg har skubbet flere ting ind i grund applikationen. Jeg kan stadig nå planen, selvom jeg nok må se de her to første sprint lidt mere samlet. Det er også den stress, som har ført til, at jeg har arbejdet meget over og bagefter været lidt smadret. Så det skal jeg være mere opmærksom på. Desuden var det en hård omgang at rejse til København og spille rollespil i weekenden. Det var for ambitiøst, men heldigvis har jeg ikke planlagt det for andre weekender.

### C.2 231129

Procesevaluering: Min sidste evaluering var ikke så struktureret, så laver det lidt om. Sidste gang læste jeg dagbøger og evaluerede det lidt. For at få mere ud af det, deler jeg evalueringen op i 4 dele, sidste evaluering, sprintprocessen, tidsplan og produkt, og drager nogle konklusioner heraf.

Sidste evaluering: I sidste evaluering fokuserede jeg på projektets tekniske dybde. Tildeles at det havde stresset mig at få den kritik og jeg derfor ville tilføje AI i projektet. Desuden konkluderede jeg, at jeg skulle arbejde mere regelmæssigt, således jeg ikke var udmattet efter at have arbejdet meget i en periode. Sidst men ikke mindst slog jeg mig selv i hovedet for at være taget til rollespil i København i weekenden.

Det er noget pjat med, at jeg ikke skulle være taget væk i weekenden. Det var fornuftigt at ændre projektet, også for at få AI med, men jeg skal huske, at det kan være i større eller mindre grad.



Sprintproces: Jeg havde et kortere sprint grundet, at jeg havde flyttet sidste evaluering til torsdag eftermiddag. To dage gik med problemer android studio og senere at sætte expo op. Desuden har det bare taget længere tid end beregnet at lave front-end'en, fordi jeg ikke rigtigt kendte react native.

Jeg bør dog overveje at ligge evaluering umiddelbart før planlægning, da det reelt har været udfaldet de sidste to uger. Desuden har jeg haft gode erfaringer, egentlig, med at arbejde mindre regelmæssigt og blive sent oppe og programmere. Jeg bør overveje klart at definere nogle mål, når jeg gør det. Det var dumt, at jeg ikke tog et framework, hvor jeg havde mere erfaring og selvom jeg kun har lidt erfaring med ionic er det bedre end intet. Det har til gengæld været lærerigt.

Tidsplan: På Produkt siden er jeg lidt bagud, men ikke mere, end det er realistisk nok, at jeg komme efter det. På dokumentationsfløjen er jeg meget bagefter. Jeg har lagt en del buffer ind og den blev lidt ædt af at skulle lave noget AI. Jeg har dog også weekenden, som jeg vidst, at jeg ville komme til at bruge.

Det vigtigste er at komme i mål, så selvom næste uge står i AI's tegn, så står det faktisk mere dokumentation og få lavet produktet færdigt tegn. Jeg skal have lagt en plan med Iben, min korrekturlæsende kusine, for hvornår jeg kan aflevere til hende. Af erfaring skal jeg stadig have lidt buffer.

Produkt: Det går ok og det skal nok blive til noget. Jeg skal have lavet app færdig, implementeret FastAPI-users og lagt det op på nettet. Expo kan køre på app'en, men lige nu udvikler jeg på web-applikationen. Jeg har et møde om AI udvikling med Nhat senere i dag og derefter skal jeg have lavet en plan for det.

Prioritet er at blive færdigt med noget og bygge videre derpå. Så prioritet er: • APP færdig (bare funktionelt - ikke design) • Brugerhåndtering • Noget AI • Lagt på nettet • Refactor koden • APP design/UX (alt ikke funktionelt) • Test på tablet • Mere AI Det burde godt kunne lade sig gøre

Jeg har glemt at bruge Git. Det skal jeg huske at gøre.

Konklusion: Jeg skulle ikke have valgt react native og jeg skulle ikke slå mig selv i hovedet for at være mere demotiveret i midt forløbet, særligt når det er der, hvor det allermest kedeligste er. Men nu kan jeg også mærke, at der skal ske noget. Jeg tror dog, blot at jeg skal læne mig ind i det med døgnrytme og få arbejdet, når energien er til det i stedet for at tvinge mig selv til at arbejde så regelmæssigt som muligt.

Dokumentation er det største problem, så jeg laver fredag, søndag og tirsdag til dokumentationsdage. Formålet er få brugt dem til at lave mest muligt dokumentation, mindst 10 sider (efter 10 sider må jeg gerne programmere), og sendt dem til Iben.

Programmeringsdage i dag, i morgen, lørdag, mandag og onsdag skal jeg programmere. Prioriteringer er følgende: • Funktionel APP og brugerhåndteringer • AI funktionalitet • Design og brugeroplevelse • Lagt på nettet og test på tablet

Desuden vil jeg gerne nå at refactor koden og muligvis også nå andre features. Formentlig bliver det første AI ikke så imponerede. Der er 5 dage til 4 opgaver, så udover i dag skal jeg nå en per dag. De er på ingen måde lige de opgaver. Det er bare vigtigt, at det fremgang og de behøves ikke tages i rækkefølge.

Torsdag og fredag friholdes i udgangspunkt til opsætning af opgave, skrive det sidste samt rette og finde fejl. Det er også min buffer. Forhåbentlig kan jeg holde tidsplanen, som den er nu, men nu er det ikke noget, som projektet står og falder på.

Husk at bruge Git hver dag.

## C.3 231206

Procesevaluering: Procesevalueringen er delt i 4 deler, sidste evaluering, sprintprocessen, tidsplan og produkt, og drager nogle konklusioner heraf.

Sidste evaluering: Jeg var noget optimistisk om, hvad det vil kræve at få programmet færdigt, men i bund og grund havde jeg delt det op i disse 4 hovedpunkter.

• Funktionel APP og brugerhåndteringer • AI funktionalitet • Design og brugeroplevelse • Lagt på nettet og test på tablet

Jeg har ikke noget at lave det med AI og det bliver knippet med at få programmet færdigt i aften og jeg skal realistisk set, nok bruge lidt af morgen dagen også.

Sidste uge konkluderede jeg også, at reactNative var en fejl. Det er stadig min overbevisning, men så det vil jeg ikke bruge mere tid på her.

Sprintproces: På dokumentationsfronten går det godt. Processrapporten tror jeg, at jeg har skrevet næsten det hele af. Jeg har sendt det til retning, men det er ikke rettet. Jeg forventer, at der mangler mellem 1 til to sider, som skal skrives. Jeg ved eksempelvis der mangler at skrive AI og indledningen, men der kommer formentlig også andre ting frem, når der bliver skrevet sammen og ikke bare er i småt bider. Produktrapporten mangler der nok en cirka 5-7 sider, herunder en gennemgang af programmet, installation og erhvervelse af programmet og en rækketest. Det tager selvfølgelig noget tid, men er ret håndterligt.

Arbejdsmæssigt har det været en hård uge og været 14-16 timers arbejde dagligt fra siden mandag og typisk til ret sent om aftenen. I weekenden var tættere 12 timers arbejde dagligt. Det har sgu været en hård omgang, men også nødvendigt. Heldigvis, er vi også snart færdigt.

Tidsplan: Tidsplanen er jo røget af vinduet, for den 4 punktsplan. Det er også blevet blandet sammen.

AI funktionaliteten må vi se på senere, men de 3 andre ting er realistisk nok kan blive implementeret. Om end det med at få den på nettet og testet på tablet er længere væk, end at få app'en op, som bare lige skal laves færdig.

Produkt: Produktet sejler, men holder sig flydende. Det skal forstås på den måde, at der er ret rodet lige nu, men næsten alle dele virker til at være der. Så det skal bare samles og gøres klar. Jeg er dog i tvivl, om det er realistisk at nå noget AI.

Programmet har været præget af fejl i flere af miljøerne. Backend er færdig. Frontend skal lige udvides til at passe med det hele. Hovedproblemet er, at fejlene, så derfor skal app'en lige samles systematisk. Jeg starter med login, så 2 faktor, så de forskellige opgaver typer.

Konklusion: Jeg skal samlet programmet og udvidet det der, hvor der mangler lidt. Jeg starter med samle login Derefter 2faktor Derefter task Derefter appointments, chores og projects Derefter hvad end der er tid og overskud til.

At få det på nettet og tablet kan vente til i morgen. Hvis jeg kan nå at få skrevet noget brugervejledning til produktrapporten, ville det være perfekt.

AI er lavest prioritet. Målsætning er, at få det andet næsten færdigt, også se på, om det er muligt og ellers skrive mig ud af, hvordan jeg ville gøre det.