|  |
| --- |
| [Forfatterens navn]  10-11-2022 |

|  |
| --- |
| [Firmanavn] |
| Processrapport |
| [Dokumentets undertitel] |



|  |
| --- |
| **Elev:**  Emil Ømark Jensen |
| **Firma:**  Netcompany |
| **Projekt:** |
| **Uddannelse:**  Datatekniker med special i programmering |
| **Projektperiode:**  17/10/2022 – 10/11/2022 |
| **Afleveringsdato:**  10/11/2022 |
| **Fremlæggelsesdato:**  17/11/2022 |
| **Vejledere:**  Lærke Brandhøj Kristensen |
|  |

# Titelblad



Techcollege Aalborg,

Struervej 70,

9220 Aalborg

# Læsevejledning

##### Forkortelser

MVC: Model view controller

API: Application programming interface

ORM: Object relation mapping

# Forord

Hvad skal man vide om rapporterne og projektet før man går i gang med at læse?

Indholds fortegnelse

[Titelblad i](#_Toc118858893)

[(Indeks) ii](#_Toc118858894)

[Læsevejledning ii](#_Toc118858895)

[Forord ii](#_Toc118858896)

[(Indledning) 1](#_Toc118858897)

[Intro til problemet og processen, i modsætning til forord som er intro til rapporten og processen bagved. 1](#_Toc118858898)

[Case beskrivelse 1](#_Toc118858899)

[Problemformulering 1](#_Toc118858900)

[Projektplanlægning 1](#_Toc118858901)

[Estimeret tidsplan 1](#_Toc118858902)

[Arbejdsfordeling 1](#_Toc118858903)

[Metode- og teknologivalg 1](#_Toc118858904)

[Backend 2](#_Toc118858905)

[Hvorfor 2](#_Toc118858906)

[Beskrivelse 2](#_Toc118858907)

[Alternativ 2](#_Toc118858908)

[Frontend 3](#_Toc118858909)

[Hvorfor 3](#_Toc118858910)

[Beskrivelse 3](#_Toc118858911)

[Alternativ 3](#_Toc118858912)

[Server 3](#_Toc118858913)

[Hvorfor 3](#_Toc118858914)

[Historie 4](#_Toc118858915)

[Alternativ 4](#_Toc118858916)

[Væsentlige elementer fra produktrapporten 4](#_Toc118858917)

[Realiseret tidsplan 4](#_Toc118858918)

[Konklusion 4](#_Toc118858919)

[Diskussion 4](#_Toc118858920)

[(Referencer) 6](#_Toc118858921)

[(Bilag) 7](#_Toc118858922)

# (Indledning)

# Intro til problemet og processen, i modsætning til forord som er intro til rapporten og processen bagved.

# Case beskrivelse

Næsten alle folk har flere forskellige deadlines, events og/eller møder de skal deltage i, som skal deles med venner, familie, kollegaer eller måske kunne være private reminderes for en kommende deadline. Dette kan være svært eller næsten umuligt for nogle folk at kunne holde styr på i hovedet, uden at komme til at glemme eller overse en deadline, eller komme forsendt i gang med forberedelse til et event. Dette kan før til at man føler sig meget mere stresset, spildt tid brugt på at huske hvad var det nu lige jeg skulle eller går hele ugen og tænker over hvad man skal nå enden eventet.

(flyt til design)  
”Dette ville kunne fikses med en simpel kalender system hvor i brugeren logger ind og kan oprette farvekoordineret deadlines, møder og events. Derefter kan oprettede element vil kunne deles med andre bruger i systemet og sætte e-mail/webside notifikationer op om reminder.”

## Problemformulering

Hvordan kan det gøres nemmer for folk at holde styr på deres deadlines, møder og events i forhold til arbejde og private arrangement med venner og familie.

# Projektplanlægning

## Estimeret tidsplan

Tidsplanen må gerne ligge i bilag hvis det fylder meget.

Se Biag A: Estimeret Tidsplan.

Lad være med at have bilag du ikke skriver om og referer til.

## Arbejdsfordeling

Hvordan har du fordelt arbejdet på programmet og rapporterne.  
har du fyldt din tidsplan?  
Er der noget du har brugt forlangtid på og hvordan har det påvirket din planlægning?

# Metode- og teknologivalg

Dette problem ville kunne fikses med en simpel kalender system hvor i brugeren logger ind og kan oprette farvekoordineret deadlines, møder og events. Derefter kan det oprettede element deles med andre bruger i systemet og sætte e-mail/webside notifikationer op om reminder.  
For at bygge sådan et system har jeg valgt at bruge disse teknologier og metoder.

## Backend

Til backend system har jeg valgt at bruge Microsofts meget populære web framework ASP.Net core til mine API og entity framework core til styring af database

ORM, MVC, mediator, programmerings paradimer

### Hvorfor

Til Backend er der valgt at brug C# som sprog og det er der for at ASP.Net core og entity framework core er valgt.  
Grunden til at brug af samme sprog til backend er smart over at vælge andre teknologier. Er at det gør det meget nemmer at integrere de forskellige frameworks og få dem til at snakke sammen.

Til backend arkitektur bliver der brugt MVC design patterns til API sammen med mediator design patterns mellem services og handleres.

Et smart valg ved at bruge et design patt,erns som MVC og Mediator er at det hjælper koden med at være blive mere abstract, separate og læs bar.

Til database styring er der valgt entity framework csore som brug et ORM design til at snakke med en MSSQL-database

Dette er valgt fordi det også gør at man kan bruge C# til at snakke med databasen i forhold til at skrive sql.

### Beskrivelse

ASP.Net core er et web framework lavet af Microsoft som en vider udvikling på deres ASP.Net.  
Et web framework er et framework der lader en developer bygge applikation, API og eller services ved at bruge web protokoller

Entity framework core er et ORM framework lavet af Microsoft som en vider udvikling på deres Entity framework  
Et ORM framework er et mapping framework som mapper din kode i det her tilfælde C# om til et databasesprog and på den valgt database i det her tilfælde en MSSQL-database.

Det giver mulighed for at bruge de samme modeller i koden og til bygning af databasen

MVC-design patterns bliver brugt til API i denne applikation. I MVC opdeler man koden i 3 forskellige komponenter, model, view og controller. Det vil sige at når man eksempel vis laver et kald til et backend endpoint for at hente kalender information ned i et tids periode, vil man sende en kommando som indeholder information eksempel: {Type: KalenderType.Week, Date: ”27-11-2022”} hvor type er en enum som forklar hvad for en type kalender vi vil havde tilbage og date er den uge hvor den dato ligger i. Denne kommando vil virke som en model som er sendt til API endpointet som er controlleren og vil returner daten om hvad events der ligger i denne uge som et view der bliver brugt af fronten til at display daten i kalenderen

Mediator design patterns bliver brugt mellem service og handlers det fungere ved man laver et mellem led som tager en kommando ind eksempel vis modellen fra MVC og sender den til dette mellem led som kalder den associeret handler og authorisation.

### Alternativ

Der er mange alternative web framework man kan bruge til at lave en backend API som Django eller Ruby on Rails. De andre her kan meget det samme som ASP.net core men bare i et andet sprog

## Frontend

Til min frontend har jeg valgt at bruge Googles web application framework angular og mattlewis92’s angular calendar for nogen simple kalender UI component i angular

### Hvorfor

Til frontend har jeg valgt angular for at separere front og back -end. Dette hjælper med at man i fremtiden ville kunne udskifte hele frontend uden at tinge på at der er brugt samme kode til modeller brugt af både front og back -enden

Mattlewis 92 angular calendar er brugt da det er den nyre version a bootstraps kalender ui elementer

### Beskrivelse

### Alternativ

Blazor

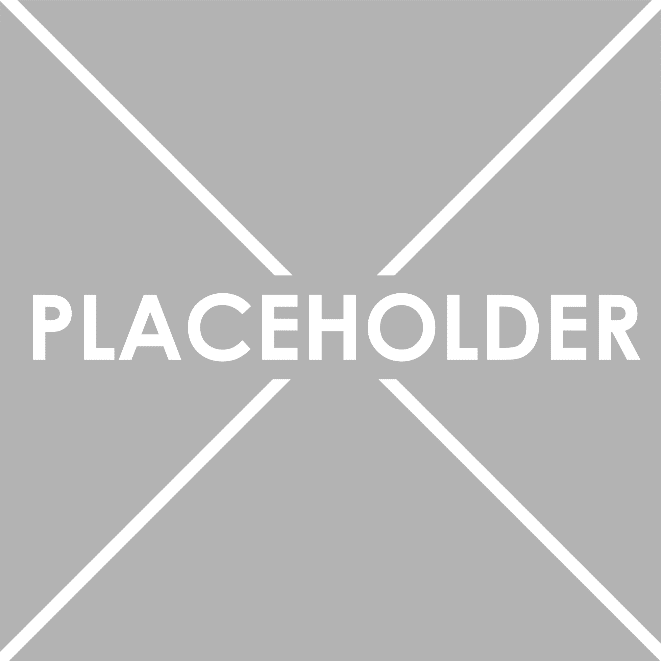
## Server

Til min server har jeg valgt at bruge en Microsofts IIS web server og MSSQL database some ville bliver hosted gennem en Docker container server

### Hvorfor

### Historie

### Alternativ



Figur 1 PLaceholder billede

# Væsentlige elementer fra produktrapporten

Eks. henvisning til afsnit og punkter i produktrapporten

Find relevante overskrifter baseret på indhold

# Realiseret tidsplan

# Konklusion

Skal hænge sammen med case beskrivelsen og problemformuleringen.

## Diskussion

Diskutere forskellige side af løsningen, fordele/ulemper.

Hvis der var mere tid, hvor kunne produktet udvides eller forbedres.

Reflekter over projektet

Hvad har du lært?

Valgte du de rigtige teknologier?

# (Referencer)

Denning, T. (2018, July 17). 15 Typical Life Problems And How To Solve Them. *Mission*, 4. Retrieved Januar 6, 2021, from https://medium.com/the-mission/15-typical-life-problems-and-how-to-solve-them-c56838f49738

<https://www.tutorialspoint.com/hibernate/orm_overview.htm>  
<https://www.tutorialspoint.com/mvc_framework/>

<https://github.com/mattlewis92/angular-calendar>

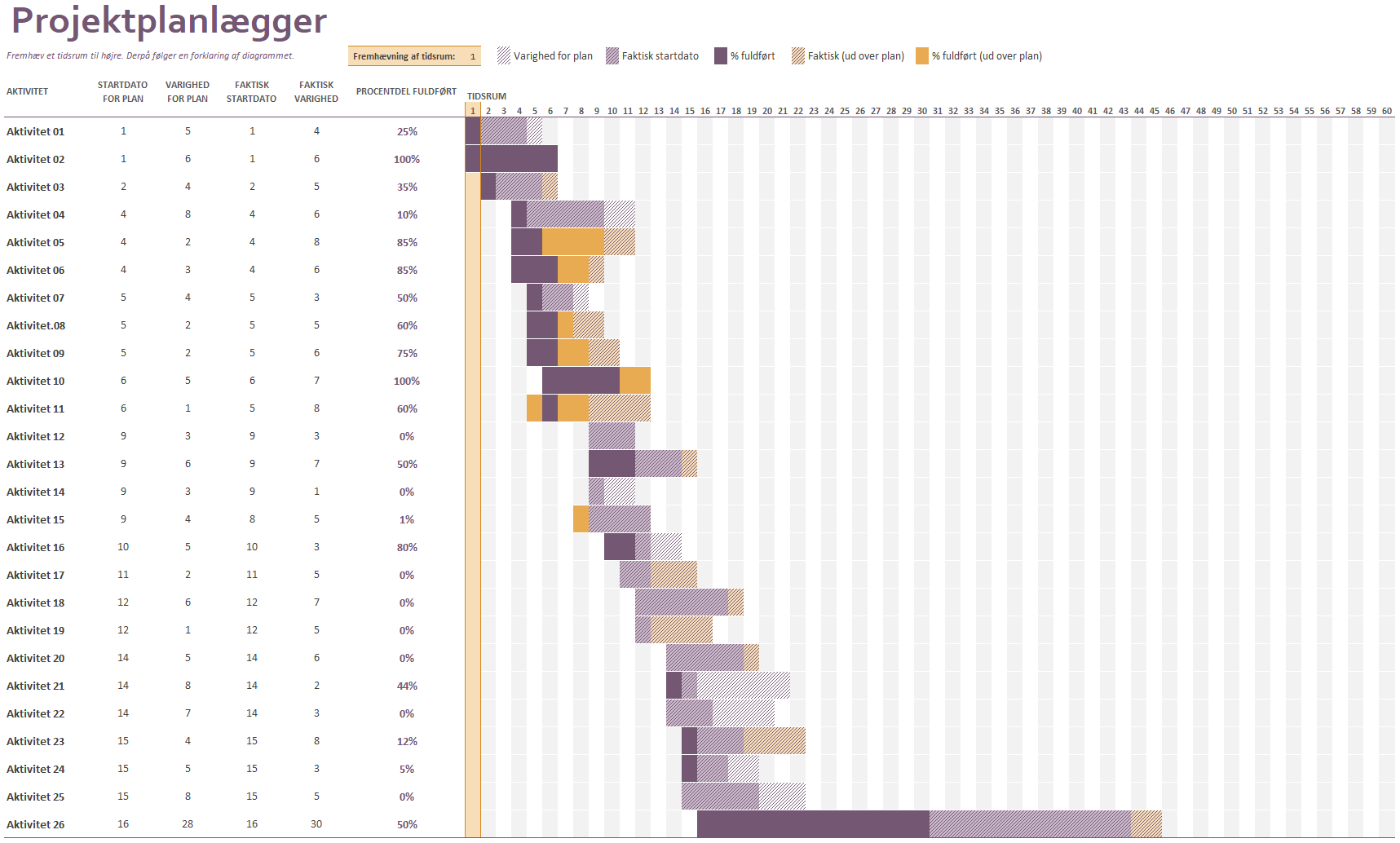
# (Bilag)

[Figur 1 PLaceholder billede 1](#_Toc80946617)

[Figur 2 Biag A: Estimeret Tidsplan 5](file:///C:\Users\lkri\Documents\Undervisningsmateriale\H6\Rapport%20materiale\Procesraport%20template.docx#_Toc80946618)

#### Estimeret Tidsplan

Figur Biag A: Estimeret Tidsplan



#### Dagbog/Logbog

Den skal føres fra dag 1 i forløbet.