|  |
| --- |
| [Forfatterens navn]  14-11-2022 |

|  |
| --- |
| [Firmanavn] |
| Produktrapport |
| [Dokumentets undertitel] |



|  |
| --- |
| **Elev:**  Daniel Simonsen |
| **Firma:**  Skolepraktik |
| **Projekt:**  SmartWeight |
| **Uddannelse:**  Datateknikker m. speciale i programmering |
| **Projektperiode:**  19/09/2022 – 23/11/2022 |
| **Afleveringsdato:**  23/11/2022 |
| **Fremlæggelsesdato:**  24/11/2022 |
| **Vejledere:**  Navn på vejledere |
|  |

# Titelblad



Techcollege Aalborg,

Struervej 70,

9220 Aalborg

Indholdsfortegnelse

[Titelblad i](#_Toc98920047)

[(Indeks) ii](#_Toc98920048)

[Læsevejledning ii](#_Toc98920049)

[Forord ii](#_Toc98920050)

[(Indledning) 1](#_Toc98920051)

[Intro til problemet og processen, i modsætning til forord som er intro til rapporten og processen bagved. 1](#_Toc98920052)

[Case beskrivelse 1](#_Toc98920053)

[Problemformulering 1](#_Toc98920054)

[(Afgrænsning) 1](#_Toc98920055)

[Projektplanlægning 1](#_Toc98920056)

[Estimeret tidsplan 1](#_Toc98920057)

[Arbejdsfordeling 1](#_Toc98920058)

[Metode- og teknologivalg 1](#_Toc98920059)

[Væsentlige elementer fra produktrapporten 2](#_Toc98920060)

[Realiseret tidsplan 2](#_Toc98920061)

[Konklusion 2](#_Toc98920062)

[Diskussion 2](#_Toc98920063)

[(Referencer) 3](#_Toc98920064)

[(Bilag) 4](#_Toc98920065)

# Brugervejledning

## Problemformulering

# Noget af det sværeste ved at tabe sig, er at holde sig motiveret til at fortsætte, og at se at det man gør, faktisk gør en forskel for ens krop og helbred. Selvom projektet ikke kan vise dine procenter og fortælle dig, hvordan din levetid har forbedret sig, kan den give dig en oversigt over din vægt, uden at du selv skal holde styr på det…

Så, hvordan kan det gøres nemmere at holde styr på ens vægt, uden at bruge for meget tid på, at skrive det ned i en bog, og hvordan kan produktet hjælpe med motivationsboost?

# Teknisk produktdokumentation

## ASP.NET API

Jeg bruger ASP.NET’s API projekttype. Her har jeg bl.a. adgang til Swagger, som kan hjælpe med tests af endpoints og også give en god visuel præsentation af mine controllers og deres endpoints.

### Connections

Forbindelser/Relationer mellem bruger og vægt. Dette bliver brugt, når vægten sender en værdi op til api’et. For at tilknytte en bruger til vægtmålingen, skal der oprettes forbindelse til vægten.

Hver gang der bliver oprettet forbindelse til en vægt, tjekker den om der er en vægtmåling, der ikke har fået en brugerreference. Hvis der er, bliver brugerreferencen i connectionen tilføjet til vægtmålingen.

A screenshot of a computer

Description automatically generated

**POST**: Brugeren kan tilføje en forbindelse mellem sig selv og vægt.

**GET /all:** Få alle brugerens tidligere forbindelser, så der let kan genoprettes forbindelse til eksisterende vægt.

**GET:** Få brugerens aktive forbindelse

**DELETE**: Afbryd eller slet brugerens nuværende forbindelse. Fuld slet kræver query delete=true

### Measurements

Vægtmåling fra vægten med relation til brugeren, som ejer vægtmålingen.

Background pattern

Description automatically generated

**POST /collection:** Send en liste af målinger, hvis den embeddede enhed har mistet internetforbindelse, og har gemt flere vægtmålinger, der skal sendes.

**GET /overview/{userId}:** Alle målinger relateret til brugeren

**GET /all:** Alle målinger fra alle brugere. Kan filtreres via filter=0,1,2 efter internt enum. Endpointet er *kun* til testing.

**DELETE /all:** Slet alle målinger. Ligesom dens GET, kan målingerne filtreres og er kun ment til tests.

**POST:** Tilføj fuld vægtmåling til databasen. Bruges kun internt i API’et.

**GET:** Få alle vægtmålinger gemt. Her er der ingen filtrering.

**GET /{id}:** Få bestemt vægtmåling efter id.

**PUT /{id}:** Opdatér en vægtmåling efter id.

**DELETE /{id}:** Slet en vægtmåling efter id.

### Partial Measurements

En *Partial Measurement* er en vægtmåling uden brugerrelation. En vægt kan kun vide, hvem den selv er, og hvilket tal den har målt. Derfor sender den en partial measurement til API’et, så API’et selv kan lave den om til en fuld Measurement.

Background pattern

Description automatically generated

**DELETE /{aggregated}:** Slet alle partial measurements via aggregated string, der er sat sammen med alle id-er, separeret med ”,”.

**POST:** Den embeddede vægt, der sender sine oplysninger f.eks. WeightId, Value & Timestamp.

**GET:** Få alle partial measurements. Mest brugt til testing.

**GET /{id}:** Standard få en partial measurements via id.

**PUT:** Standard opdatér en partial measurement via id.

**DELETE:** Standard slet en partial measurement via id.

### Users

Brugere i systemet. En bruger kan bruge vægten, hvor vægten enten kan være privat eller offentlig. For at sikre at en bruger får de rigtige tal gemt til dem selv, skal de forbinde til vægten.

Background pattern

Description automatically generated

**POST /login:** Få en bruger logget ind på appen.

**DELETE /login/{userId}:** Få en bruger logget ud af appen. Her afbrydes alle vægt forbindelser, der måtte være aktive.

**POST:** Opret bruger I systemet.

**GET:** Få alle brugere i systemet.

**GET /{userId}:** Få bestemt bruger efter id.

**PUT /{userId}:** Opdatér bestemt bruger efter id.

**DELETE /{userId}:** Slet bruger fra systemet.

### Weight

Objektreferencen til en embedded vægt. Den har et id, der kan bruges til at spore tilbage til, hvilken vægt der tog en bestemt måling. Vægten kan også gøres mere personlig via eget givet navn.

Background pattern

Description automatically generated

**POST:** Opret ny vægt i systemet

**GET:** Få alle vægte i systemet.

**GET /{weightId}:** Få bestemt vægt efter id.

**PUT /{weightId}:** Opdatér bestemt bruger efter id.

**DELETE /{weightId}:** Slet vægt fra systemet.

## SQL Database

Til database valg, valgte jeg at gå med en klassisk Microsoft SQL Database. EntityFramework var en stor hjælp til databasehåndteringen. Da jeg bruger code-first princippet, skulle jeg bare lave mine modeller og lave en smule opsætningskode for at få databasen til at spille sammen med mine modeller og hermed mit API og app.

## MAUI Cross-platform App

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?

## ESP Embedded Enhed

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

Skriv beskrivelser til dine billeder og referer til dem i teksten eksemple se Figur 1

# 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

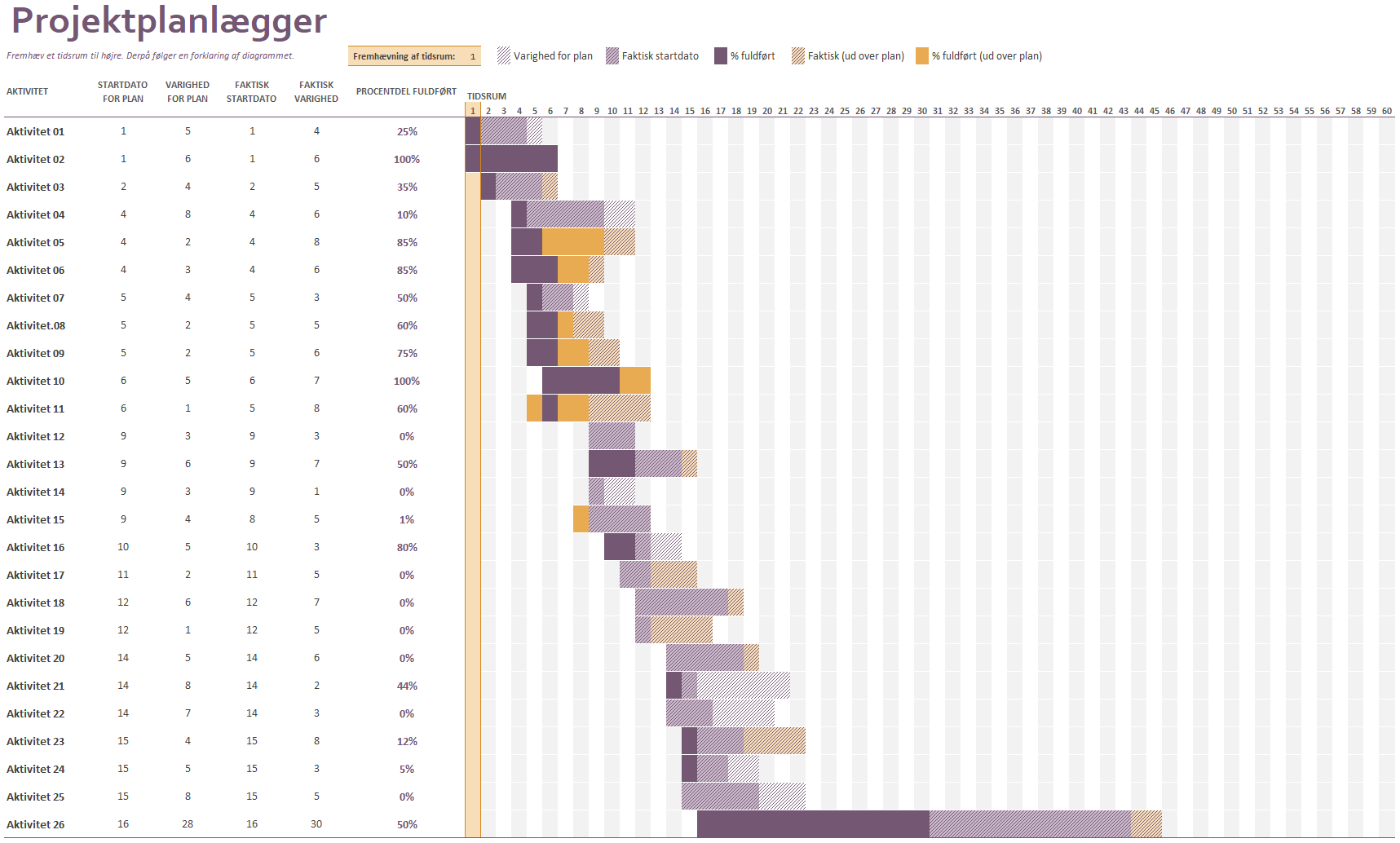
# (Bilag)

[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.