|  |
| --- |
| [Forfatterens navn]  19 |

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





# Mini-svendeprøve

|  |
| --- |
| **Elev:**  Omidreza Ahanginashroudkoli |
| **Skolepraktik:**  Teglværket 9400 Nørresundby |
| **Projektnavn:**  Vareskanning |
| **Uddannelse:**  Datatekniker med speciale i programmering |
| **Projektperiode:**  19/Sep/2022 – dd/mm/2022 |
| **Afleveringsdato:**  dd/mm/yyyy |
| **Fremlæggelsesdato:**  dd/mm/yyyy |
| **Vejledere:**  Lærke Brandhøj Kristensen |
|  |

**Elev underskrift Vejleder underskrift**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# (Indeks)

Afgræsning; 1

Diskussion; 2

Indledning; 1

# Læsevejledning

I læsevejledningen beskriver man hvordan rapporterne anbefales at læses og evt. andre skriftlig formalia, som forkortelser og engleske fagord.

*Alle overskrifter i parentes er ikke et krav at have med.  
Nogen overskrifter skal omskrives for at passe ind.*

# Forord

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

Indholds fortegnelse

[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)

# (Indledning)

# I denne rapport vil jeg illustrere hvordan jeg kom frem til mit slutprodukt, som er beskrevet i den relateret produktrapport.

# Jeg vil illustrere projektets case og problemformuleringen og forklare hvordan jeg planlægger at færdiggøre projektet.

# Jeg vil gå i detaljer og illustrere mine valg af teknologier, samt med hvad er alternativet af de teknologier som man kan bruge i stedet.

# Til slut vil jeg gennemgå over hvordan projektet kunne have været bedre hvis jeg havde mere tid at arbejde på det og konkluderer mit projektforløb.

# Projektet udviklet af Omidreza ahanginashroudkoli, i perioden 19. september 2022 til (projektet afslut tid), med vejledning fra Lærke Brandhøj Kristensen.

# Case beskrivelse

Teknologi ændrer sig, og menneskers liv bliver lettere ved at bruge det. Dette projekt hjælper med at skabe en vareskanning system hvor det er let og bruge for alle sammen.

Interessant del med projektet er at du får en liste af de skannet vare på din mobile app også en Graph af dem som viser antal skannet vare og tidspunktet for scanningen, så kan man danne et overblik over sine vare.

Målgruppe er til de virksomheder/folk, der har brug et system, der kan hjælpe dem med, har et overblik over de solgte vare, et system der ikke laver menneskelig fejl og har ikke behov for fysisk dokumentation. vareskanning system der kan organisere og hjælpe dem med se hvor mange vare de har solgt og hvornår.

## Problemformulering

Hvordan kan jeg skanne vare og visualisere dem på en app?

# 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

Jeg har valgt at planlægge mit projekt ud fra min rutine som er at start fra backend og gå fremad til frontend. Min estimerede tidsplan kan ses i bilag 1.

Derfor vil jeg starte med min Library model så kan jeg bruge den i hele projektet, ved hjælp af min Library Model og Entity Framework, kan jeg sætte min database op og køre, derefter kan jeg Begynd med på min API som mine forskellige funktionalitet skal laves det igennem. Næste skridt bliver at sætte min mikrocontroller op og gøre den klar til at sende det information til min database.

Til sidst vil jeg begynde med programmere min Cross-platform mobil app, gennem hvilken

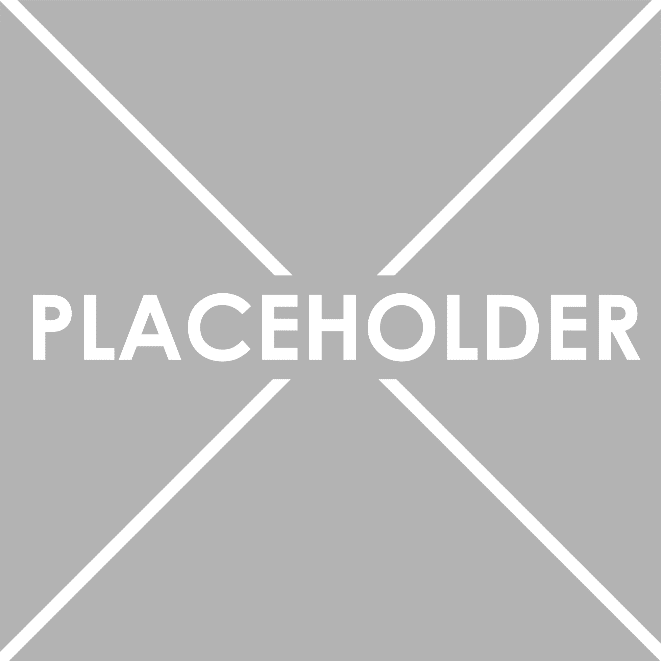
brugeren vil have adgang til deres oplysninger og nogle andre funktioner.

Rapporterne vil jeg arbejde på løbende gennem hele forløbet med forbedre det.

# Metode- og teknologivalg

I dette afsnit vil jeg diskutere at hvorfor jeg har valgt de teknologier og om der er alternativer af dem, som kan gøre det sammen eller bedre.

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



Figur 1 PLaceholder billede

# 

Database

Jeg har valgt at bruge MSSQL database i mit projekt. Det har jeg valgt fordi den giver mest mening efter min mening og er noget jeg har mest erfaring og kendskab til. MSSQL også har en applikation som kaldet Microsoft SQL Server Management Studio (IDE) da kan hjælpe med har et overblik over sine data og ændre i den.

Relationel vs. Dokument

Der er også anden slags database man kunne have brugt som kaldet dokumentdatabase. I en dokumentdatabase, ligger dataene som enkelte entries uden relation til hinanden, i stedet for rækker i tabeller. Dokumentdatabaser er typisk hurtigere end relationelle databaser, men har ikke mulighed for at have relationer mellem data og det kan være en ulempe med det.

I stedet for en relationel database

Entity Framework

Jeg har sat min database op via code-first princippet og Entity Framework. Code-first er smart, fordi i stedet for at bruge en masse tid på at lave en database og skrive kode der kan kommunikere med den, kan du lave din modeller og skrive kode for dem og generere din database ud fra det. Det smart del er at du minimerer menneskelige fejl og sparer en masse tid.

Web API

API ‘er er nødvendige for at bringe applikationer sammen for at udføre en designet funktion bygget op omkring deling af data og udførelse af foruddefinerede processer. API fungerer som mellemmand giver os udviklere mulighed for at bygge applikationer hvor de kan interagere med hinanden. Fordel ved at bruge API ‘er er det hjælper med har bedre data kvalitet gennemgang og oprydning. API kan understøtte hurtigere og lettere datamigrering og det giver større fleksibilitet.

Jeg har valgt at bruge API i ASP.NET Core fordi det er jeg har kendskab til. Jeg har valgt .NET Core frem for .NET Framework, fordi det er nyere og understøtter flere platforme.

ASP.NET vs. Python vs. Node.js

WEB API er et bedre valg til enklere, lette services. WEB API kan bruge ethvert tekstformat inklusive XML og er hurtigere end WCF.

Der er forskellige måder at oprette en API i Python, den mest brugte er FastAPI og Flask. Selvom begge disse Python-frameworks er enkle og nemme at bruge, har FastAPI fordelen, da den kompenserer for Flasks begrænsninger. FastAPI har en høj ydeevne, samtidighed kan nemt understøttes, og det tilbyder et enkelt og letanvendeligt afhængighedsinjektionssystem.

Node.js er et open-source framework som man kan bruge til at eksekvere JavaScript på en server. Siden Node.js bruger JavaScript, den er nem at bruge den foretrækkes af folk. Node.js tilbyder nem skalerbarhed på grund af sin asynkrone og begivenhedsdrevne natur.

Jeg har brugt ASP.NET, fordi det er jeg har mest erfaring og kendskab til og den kan det jeg har bruge for til den opgave.

Docker //Måske

Mikrocontroller

Mikrocontroller er komprimeret mikrocomputer fremstillet til kontrolfunktioner i Embedded system, for eksempel robotter, hjemmeapplikationer og mange andre.

I dette projekt vil min mikrocontroller kunne skanne varerne med stregkode(barcode) med hjælp af en skanner (barcode reader) og viser den på en Oled Display, på samme tid vil sende den til vores API med hjælp af NodeMCU (ESP8266) og bliver den gemt på databasen.

NodeMCU (ESP8266) vs. ESP32

ESP8266 er billig Wi-Fi mikrochip med indbygget TCP/IP-netværk software og mikrocontroller kapacitet.

Et alternativ kunne være ESP32, som er bedre og hurtigere version af den. Siden min embedded del af mit projekt er ikke så kompliceret, så er der ikke noget grund til at bruge ESP32 fordi det ville være overkill.

Raspberry Pi

MAUI

.NET Multi-platform App UI (.NET MAUI) er et open-source cross-platform framework til at skabe native mobile og desktop apps med C# og XAML. Med .NET MAUI man kan udvikle apps der kan køre på Android, IOS, MacOS og Windows fra en enkelt delt kodebase. Jeg har valgt at bruge.NET MAUI på grund af nogle fordele som, mit kendskab til selve.NET MAUI og de sprog som den opererer på som er C# og XAML.

Maui vs. Xamarin vs. Flutter

I stedet for .NET MAUI kunne man have brugt et udvalg af andre fronthend Frameworks. Grund til jeg har valgt .NET MAUI er den understøtter det seneste .NET 6 og det er samme som jeg burger til min Model Library og min API. .NET MAUI er beregnet til at forene og erstatte teknologier som WPF. UWP eller Xamarin.

Man kunne også have brugt Xamarin Froms hvilket også er et open-source. Men Xamarin Forms er ikke lige så godt og opdateret som .NET MAUI siden den kører på .NET Framework, hvis jeg skulle have brugt Xamarin Froms, så er jeg nødt til at ændre det hele til .NET Framework.

Man kunne også have brugt Flutter til at lave sin Cross-platform mobile app. Flutter er også et open-source. Udvikling i Flutter er meget hurtigt og effektivt siden den giver dig mulighed for at foretage øjeblikkelige ændringer på din app og finde bugs og rette dem. Ulemper med Flutter selvom den er meget populært, det har ikke eksisteret længe nok til at have en enorm ressourcebase og de apps som blev lavet, kan være vægtige.

Flutter bruger Dart sprog og den er ikke så meget populært og jeg har heller ikke erfaring i det.

Sprog

C#

Projektets primære sprog er C#. Min erfaring spiller en stor rolle i det valg også de teknologier jeg har valgt til min projektet, kører på C#.

Arduino (C)

Xaml

Min Cross-platform Application Maui burger Extensible Application Markup Language (XAML) til sin frontend.

IDE

Integrated Development Environment (IDE) er software applikation der hjælper programmer med at udvikle softwarekode effektivt. Her er en liste over de IDE’er jeg har brugt at udvikle mit projekt.

Visual Studio vs. Visual Studio Code

Jeg har brugt Visual Studio til sat op mit projekt. Jeg har sat op min API, min mobile app, min mikrocontroller og kodning af min database. Visual Studio tilbyder en masse funktioner og teknologier uanset hvilket sprog man bruger, der kan hjælpe programmer med spare en masse tid med at skrive ren kode og test og fejlfinde dem.

At have alle de funktioner kommer med en Ulempe, Visual Studio er en tung software at installere og kompiler kode, det er derfor nogle gange vil det gå ned og være frustrerende at arbejde med.

Alternativt af Visual Studio kunne være Visual Studio Code, men den er mere begrænsede i funktionerne, fordi den er letvægt version af Visual Studio.

Jeg har valgt Visual Studio, fordi jeg har mere kendskab til det.

Microsoft SQL Server Management Studio

Der er mange fordele ved brug af Microsoft SQL Server Management Studio, nogle af dem er nemt at installere, Enhanced Performance med indbygget gennemsigtig datakomprimering og krypteringsfunktioner tilbyder SQL-serveren forbedret ydeevne og den er meget sikker.

Jeg har valgt bruge Microsoft SQL Server Management Studio til at interagere med min database, fordi det jeg har brugt i mange år og har kendskab til.

# 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

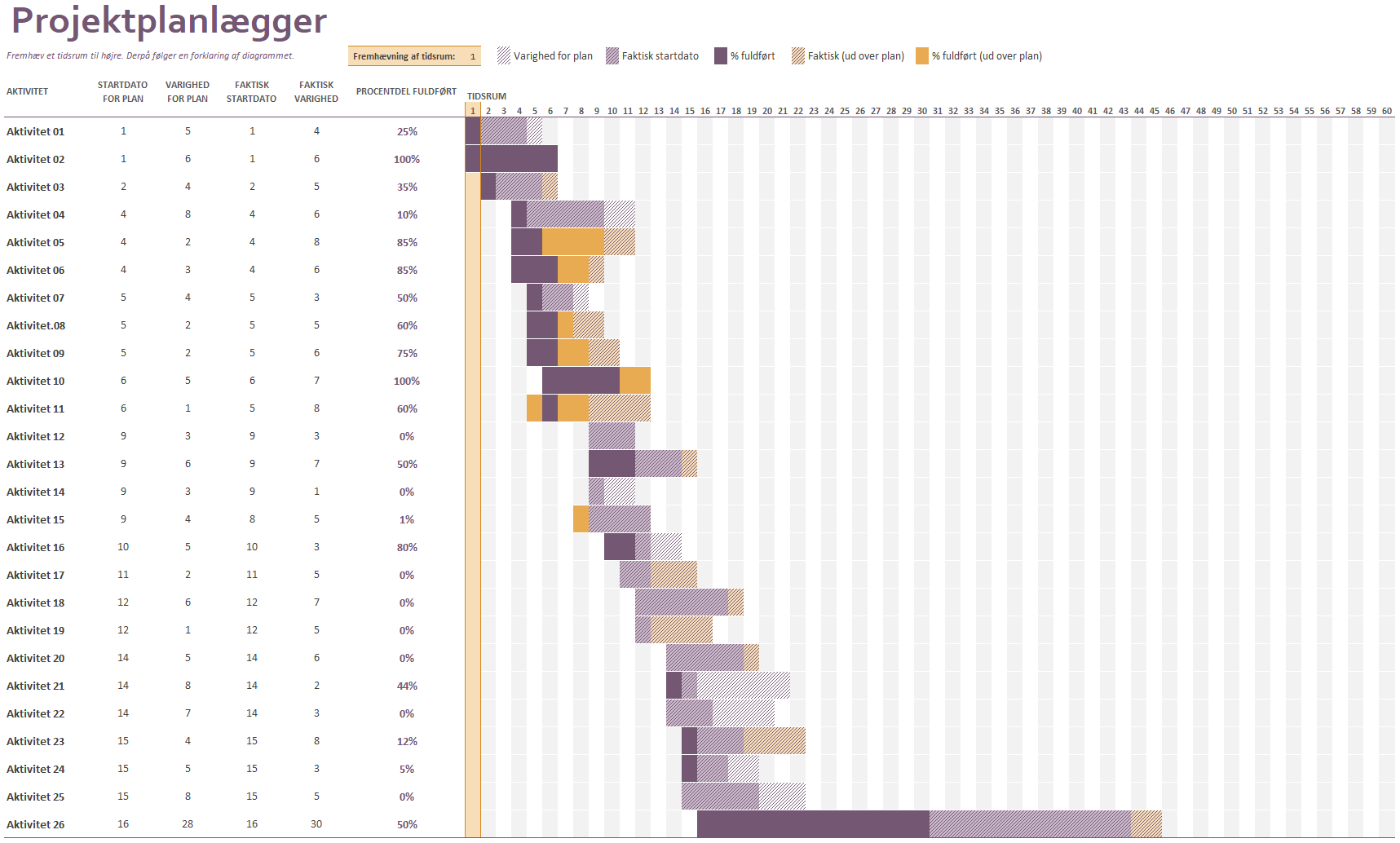
# (Bilag)

[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.