






Pakk

 postnord	DELIVERED Danmark, 0000 - Danmark	Afsender: Techbitshop Service: PostNord MyPack Collect Vægt: 5.760kg	Mere info →
 postnord	DELIVERED Risskov, 8240 - Danmark	Afsender: NIKE Yusen Herentals Service: PostNord MyPack Home Vægt: 1.180kg	Mere info →
 postnord	DELIVERED Risskov, 8240 - Danmark	Afsender: NIKE Yusen Herentals Service: PostNord MyPack Home Vægt: 1.180kg	Mere info →
 postnord	DELIVERED Galten, 8464 - Danmark	Afsender: Batteri Energi ApS Service: PostNord MyPack Collect Vægt: 0.010kg	Mere info →
 postnord	DELIVERED Galten, 8464 - Danmark	Afsender: Batteri Energi ApS Service: PostNord MyPack Collect Vægt: 0.010kg	Mere info →

Eksamensprojekt - Programmering A

Mads Gørup Gjellerod Christiansen

Vejledt af: Gorm

Abstract

Indhold

Abstract	1
1 Indledning	3
2 Fremgangsmåde	4
3 Problemformulering	5
4 Produktprincip	6
4.1 Flowchart over programmet	6
4.2 Produktkrav	6
4.3 Database	7
4.4 Asp.net Identity	7
4.5 Wireframes	7
5 Test af produkt	9
6 Gennemgang af kode	10
6.1 Frontend	10
6.2 Backend	10

1 Indledning

2 Fremgangsmåde

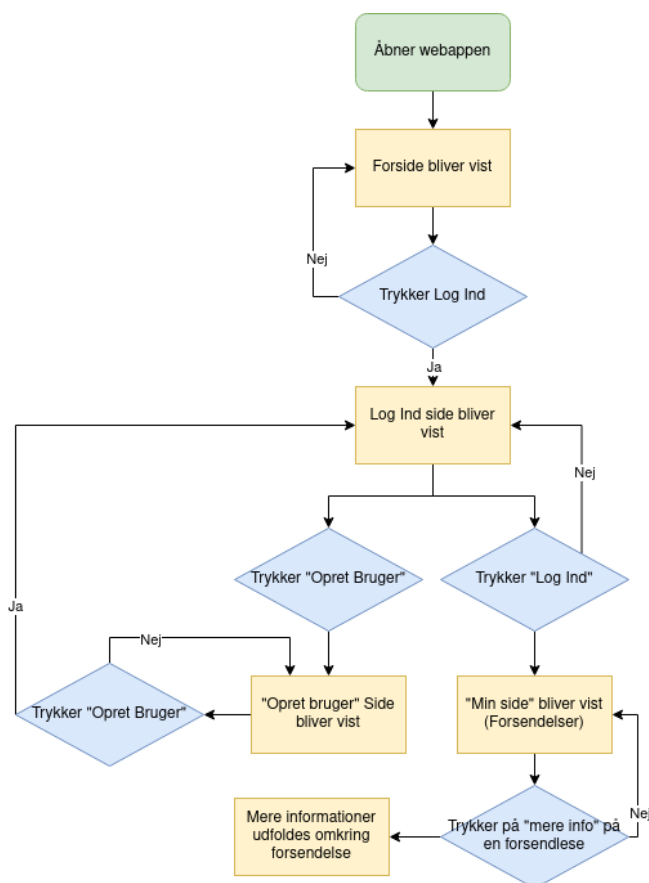
3 Problemformulering

Når man bestiller et produkt eller en vare online, findes der lige nu et væld af forskellige fragtfirmaer som udfører håndteringen og fragten af din varer, hver af disse tilbyder hver deres form for sporing eller tracking af en sådan forsendelse, men hvis man venter flere forsendelser fra forskellige af disse fragtfirmaer kan det hurtigt blive svært at finde rundt i. og der findes dermed heller ikke en løsning som samler alt dette sporingsinformation omkring sin pakke et samlet sted.

4 Produktprincip

Produktets vil være en webapp løsning, som ved brug af Googles bruger API og fragtfirma api'er såsom GLS's og Postnords API'er. skal kunne gennemse den brugerens gmail og derudfra hente trackingnummer på forsendelser fra en bred vifte af fragtfirmaer. denne tracking information skal derefter fremvises til brugeren på webappen, således tracking informationen fra alle brugerens pakker samles.

4.1 Flowchart over programmet



Figur 4.1: General funktion af programmet.

4.2 Produktkrav

Der er her opstillet en række hårde og bløde krav som det udviklede web app skal opnå. de er som følger:

Hårde krav

- Skal kunne tracke pakker fra mindst 2 forskellige fragtfirmaer
- Skal kunne fremvise pakke tracking fra mindst 2 forskellige fragtfirmaer det samme sted Oprette bruger
- siden skal have en lav kompleksitet (tilgå tracking på 2 klik)

Bløde krav

- Skal have et moderne stilrent design
- Oprette bruger med Google
- Skal fungere upåklageligt på mobile enheder (være responsivt)
- Skal selv kunne hente mails fra brugerens google konto

4.3 Database

På figur 4.2 kan man se den overordnede database struktur. Denne database er lavet med Asp.net core Identity som er en API som giver nogle login funktioner til ens program og kan findes som en nuget package.

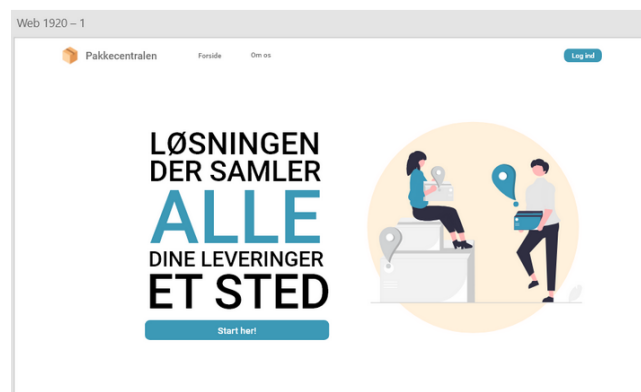
4.4 Asp.net Identity

Asp.net core er det som bliver brugt til håndtere autentificering og autorisation i webapplikationen. Denne måde at gøre det på giver en standardiseret måde at håndtere dette på. Identity giver også indbygget funktioner til at håndtere brugerregistrering, login, håndtering af adgangskoder og meget mere.

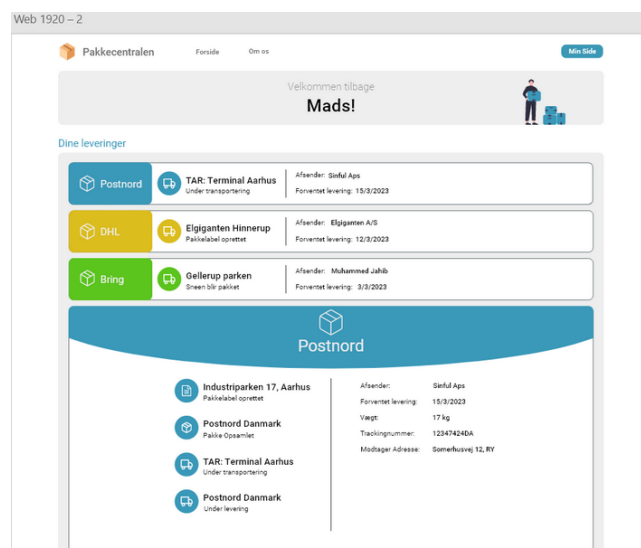
4.4.1 Google OAuth

Google OAuth bruger OAuth som er en åben protokol som alle kan bruge. Denne protokol bygger på HTTP protokolen. Ved hjælp af OAuth kan man som tredjepartsapplikation få adgang til brugers data. Det er det vi skal bruge den data vi godt vil have fundet i fra brugeren er at læse brugers emails.

4.5 Wireframes



Figur 4.3: G



Figur 4.4: D

5 Test af produkt

For at teste vores webapp efter færdiggørelse af 1. iteration vil vi teste om de hårde / bløde krav er opnået, dette vil blive gjort f.eks. ved at forsøge at foretage os at tracke 2 pakker fra 2 forskellige fragtfirmaer. I tilfældet af visuelle krav "Skal have et moderne stilrent design" vil dette kunne testes på en række testpersoner hvor deres holdning til blive taget og dermed vil der evalueres på dette krav.

6 Gennemgang af kode

6.1 Frontend

6.2 Backend

6.2.1 Oprettelse af Konto

```
1      // denne funktion sender en Email med sendgrid ved at bruge sendgrids smtp server.
2      public async Task SendEmailAsync(string toEmail, string subject, string message)
3      {
4          var apiKey = "-----"; // sendgrid api key
5          await Execute(apiKey, subject, message, toEmail);
6      }
7
8      public async Task Execute(string apiKey, string subject, string message, string toEmail)
9      {
10         _logger.LogInformation(apiKey);
11         var client = new SendGridClient(apiKey);
12         var msg = new SendGridMessage()
13         {
14             From = new EmailAddress("mads.gjellerod@gmail.com", "Password Recovery"),
15             Subject = subject,
16             PlainTextContent = message,
17             HtmlContent = message
18         };
19         msg.AddTo(new EmailAddress(toEmail));
20         // fjerner klik tracking fra koden
21         msg.SetClickTracking(false, false);
22         var response = await client.SendEmailAsync(msg);
23         // sender en besked til konsollen til at debug evtuelle fejl
24         _logger.LogInformation(response.IsSuccessStatusCode
25                                ? $"Email to {toEmail} queued successfully!"
26                                : $"Failure Email to {toEmail}");
27     }
```

Figur 6.1: EmailSender.cs

Dette er koden der sender en email når man eksempelvis registrere sig på siden og skal konfirmitere ens email. når man vil sende en email i programmet kalder man metoden `SendEmailAsync`. Denne metode har signature af at være `public` hvilket betyder at andre metoder kan kalde den. Så har den også signaturene `"async Task"` som gør at metoden gøre parrelet med andre opgaver så når man venter på en Api response så stopper programmet ikke. Denne metode tager tre parametere som er strege af tekst. Det første der sker i metoden er at vi laver en variabel der indeholder vores Sendgrid api key. herefter laves der et kaldt til metoden `Execute` som bliver kaldt asynkront ved hjælp af `"await"` operatøren.

I `Execute` metoden bliver der først oprettet et instans af `SendGridClient` som indeholder api nøglen.