

Kender du dit CO₂ aftryk?



Lev

Mål

Find alternativer

Produktrapport CO₂OPERATION

06-10-2020

Udarbejdet af:
Andi, Benjamin, Casper, Jonas



Dato:	06/10-2020
Skole:	ZBC Ringsted
Uddannelse:	Datatekniker /m. speciale i programmering
Klasse:	4008h3dakp

Projektdeltagere:

Andi

Benjamin

Casper

Jonas

Vejledere:

Camilla Mai Ryskjær

Mikkel Andreas Krøll Christiansen

Indhold

1.Produkt beskrivelse	4
2.Kravspecifikationer	5
2.1 Funktionelle krav	5
3.Brugervejledning	6
3.1 Installation	6
3.2 Anvendelse	7
4. Teknisk Produktdokumentation	9
4.1 Domænenemodell	9
4.2 ER-diagram	10
7.1 Klassediagrammer	11
7.2 Sekvensdiagram	14
9. Implementering	15
9.1 Deployment diagram	15

1. Produkt beskrivelse

Co2Opration appen er et produkt som kan holde øjet med brugers co2 aftryk. Ideen er at ved at bruge denne app skal man kunne "betale tilbage" til miljøet og derved værne om det. måden som appen måler på er ved at når man eks. køre til arbejde vil den måle brugers hastighed og afstand og baseret på det og brugers bil som indtaster når man opretter en profil, give et total co2 omkostning og kommer med et alternativ forslag til den route, og hvis man så senere hen vælger at gøre brug af det forslag vil den trække difference fra og ens total co2 aftrykt vil derefter blive mindre.

En anden side af produktet er at man også kan se nogen statistiker over total co2 forbrug, gennemsnitligt forbrug, osv. Alt det data kan derfor også blive brugt til div statistiker formål som ville kunne hjælpe miljøforsker mmv til at lægge en plan baserede på det data til at gøre vores forbrug mindre.

2. Kravspecifikationer

2.1 Funktionelle krav

Beskrivelse: De funktionelle krav indeholder alt hvad vores system skal kunne håndtere, set fra kundens perspektiv. Dette vil generelt ikke være en teknisk betegnelse, men rettere en brugervenlig betegnelse. De tekniske kriterier der eventuelt ville skulle understøtte de funktionelle krav, vil blive dokumenteret i senere kapitler.

ID	Beskrivelse
FUN-01	En registreret brugers CO ₂ aftryk skal måles automatisk, i forbindelse med transport.
FUN-02	Systemet skal komme med forslag til hvordan brugeren mindsker sit CO ₂ aftryk.
FUN-03	Geolokations data skal sendes fra mobilklienten til server med et forudbestemt tidsinterval.
FUN-04	Systemet skal kunne "gætte" på typen af transport, brugeren har benyttet.
FUN-05	Applikationen/Website skal kunne vise statistiske informationer angående brugernes CO ₂ aftryk. Fordelt i totalt, land, og by.
FUN-06	Der skal bruges E-mail og adgangskode til oprettelse af en bruger profil.
FUN-07	En bruger skal authenticeres når de logger ind.
FUN-08	En bruger skal ikke have mulighed for at se personlig information fra andre brugere.
FUN-09	Det skal være muligt at indhente en bil emissions oplysning via nummerplade opslag.
FUN-10	Det skal være muligt at oprette en bruger gennem Applikationen
FUN-11	Det skal være muligt at oprette en bruger gennem Websitet.
FUN-12	Brugere skal kunne se deres personlige CO ₂ aftryk via appen.
FUN-13	Serverside kode skal hostes igennem en supporteret løsning. Server arkitektur skal stilles til rådighed.
FUN-14	Databasen skal beskyttes mod kendte sikkerhedsbrister, som f.eks. SQL-injektions mm.
FUN-15	I tilfælde af at Systemet ikke kan udregne det korrekte køretøj, skal systemet spørge brugeren hvilket køretøj der brugt på ruten.

3.Brugervejledning

3.1 Installation

Start med at installere en windows 2019 server og oprette derefter regler i firewall for at kunne tillade kommunikation med MSSQL server og IIS'en.

Det er henholdsvis port 1432,8012. se figur 1 hvis i tvivl

Installer en MSSQL server og oprette en bruger til at read og wirtte til databasen.

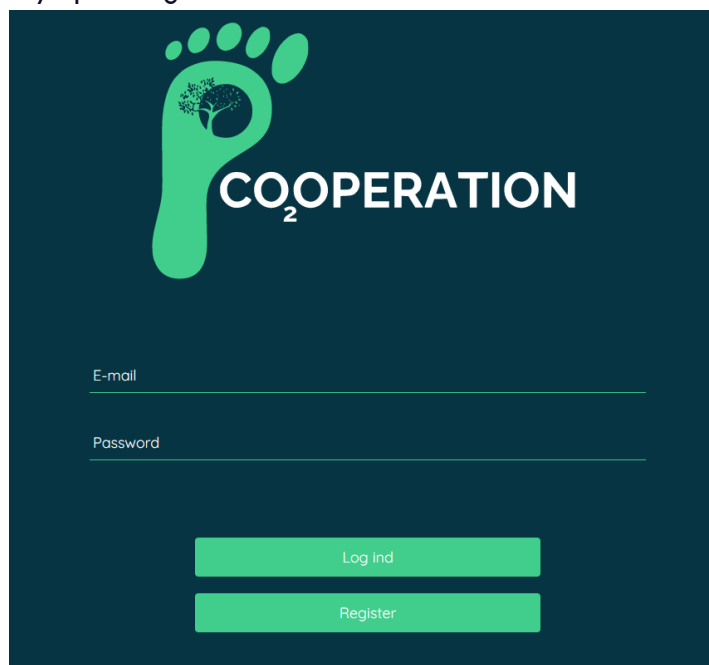
Se figur 2 for et ERD.

3.2 Anvendelse

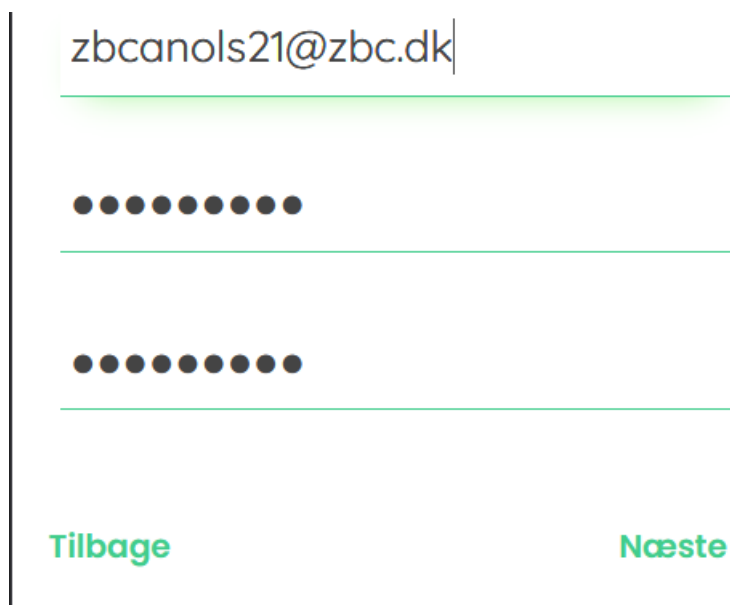
Man kan kun oprette via mobil eller via chrome hvis man trykker "ctrl + shift + m" for at få browser til at vise i mobil tilstand.

Hvis ikke man har en profil skal man gå ind på 172.16.21.44:8012/ og man ville se dette.

Tryk på "Register"

The image shows a login/register interface for 'CO2 COOPERATION'. At the top left is a logo consisting of a green footprint with a small tree inside it. To the right of the logo, the text 'CO2 COOPERATION' is displayed in white, with 'CO2' in a smaller font. Below the logo and text are two input fields: 'E-mail' and 'Password', both with green borders. At the bottom, there are two green buttons: 'Log ind' and 'Register'.

Indtast dine oplysninger og tryk derefter på "Næste".

The image shows a registration form. It has two input fields for email and password. The email field contains the text 'zbcanols21@zbc.dk'. Below the email field is a row of ten dots, with the first four being black and the remaining six being grey. Below the password field is another row of ten dots, with the first four being black and the remaining six being grey. At the bottom left is a green button labeled 'Tilbage' and at the bottom right is a green button labeled 'Næste'.

Der efter svar ja/nej til om du har en bil og du vil blive ført videre

Har du en bil?

Nej

Ja

Hvis der bliver sagt ja til bil, indtast din nummerplade og tryk på næste.

BJ80789

Bil model

TOYOTA YARIS - 2015

Tilbage

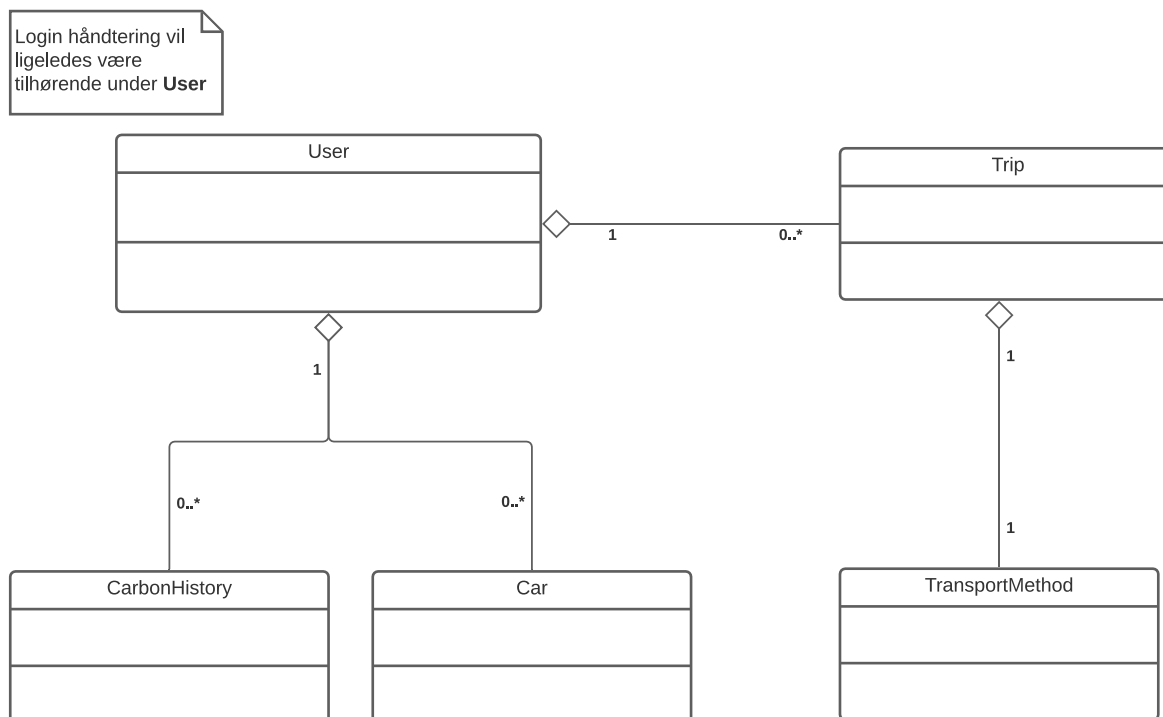
Næste

Og nu skulle du være oprettet i systemet, og kan nu gøre brug af både mobil version og web version.
Gør verden en bedre sted.

4. Teknisk Produktdokumentation

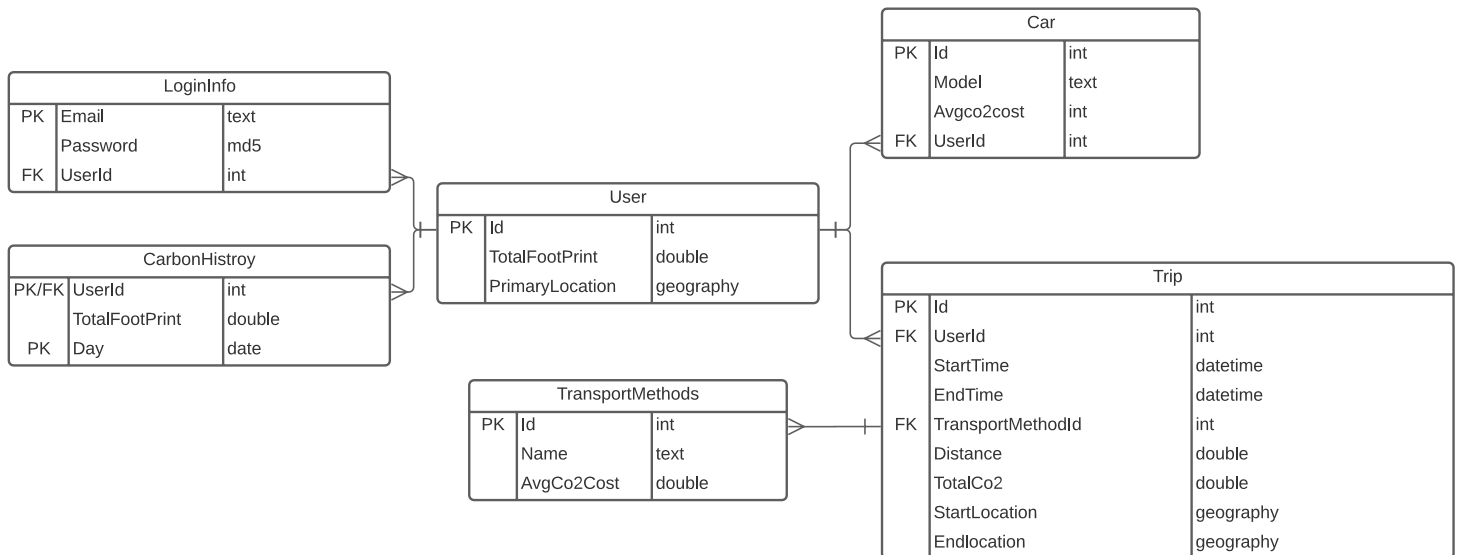
4.1 Domænemodel

Domænemodellen er udviklet ud fra vores første iterationsplanlægning. Den har ikke været nogle markante ændringer til modellen igennem udviklingsprocessen, og domæne-modellen samt kardinaliteter har holdt hele vejen igennem. De specifikke klassediagrammer bliver defineret nærmere senere i rapporten.



4.2 ER-diagram

Vores ER-diagram er udviklet efter vores domæne model. Vi har valgt at gøre brug af Entity framework, dog med Database-first fremgangsmåden. Dette betyder at vi har kunne designe vores database efter flere normaliseringsregler, end hvis vi havde gået med en fuld Entity Framework code-first model.

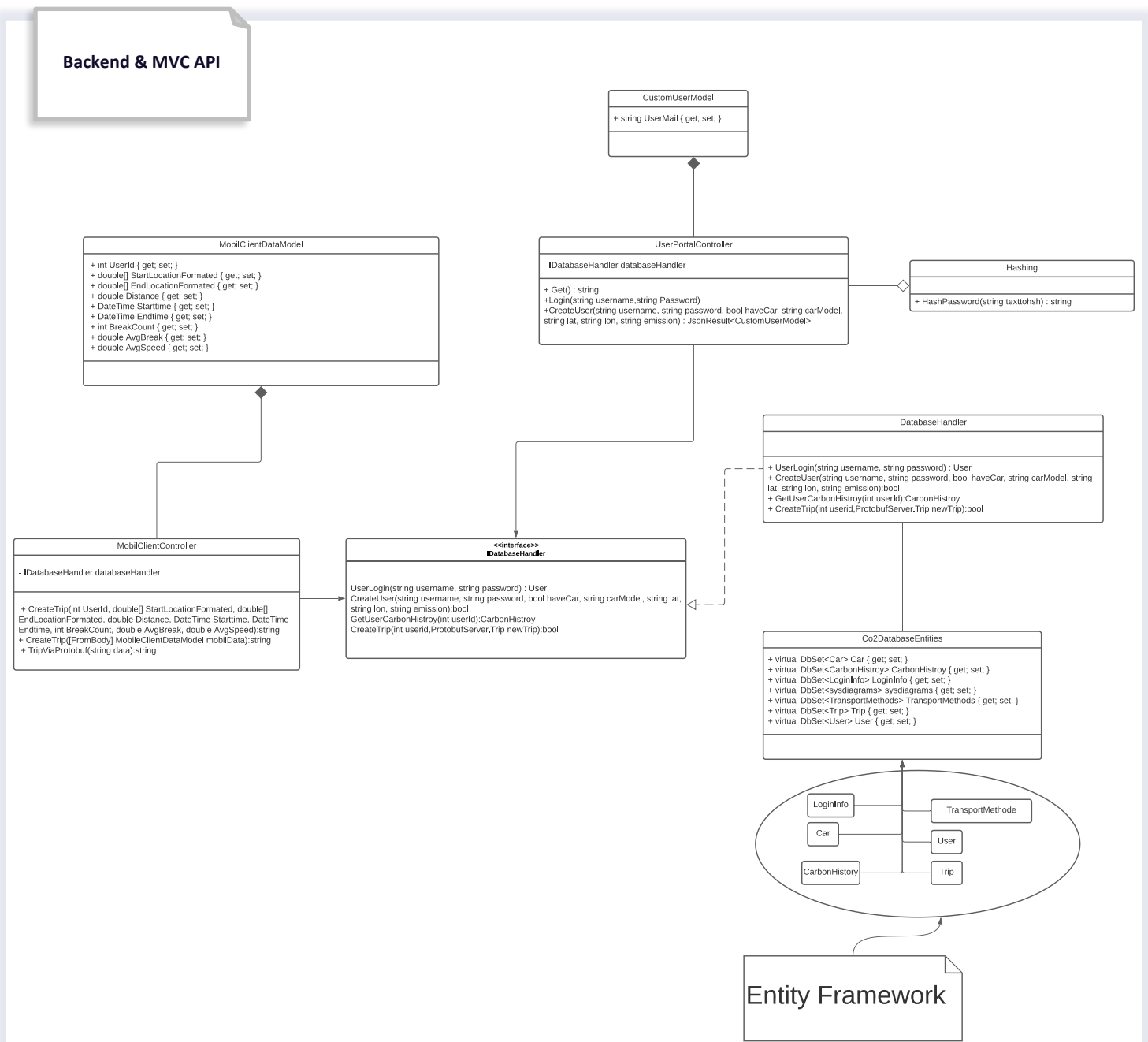


7.1 Klassediagrammer

Klassediagrammer er designet for hvert enkelt subsystem. Af note skal det siges, at specielt MVC Backenden er underlagt integreringen af Entity Framework, hvilket selvfølgelig har en effekt på klasse opbygning.

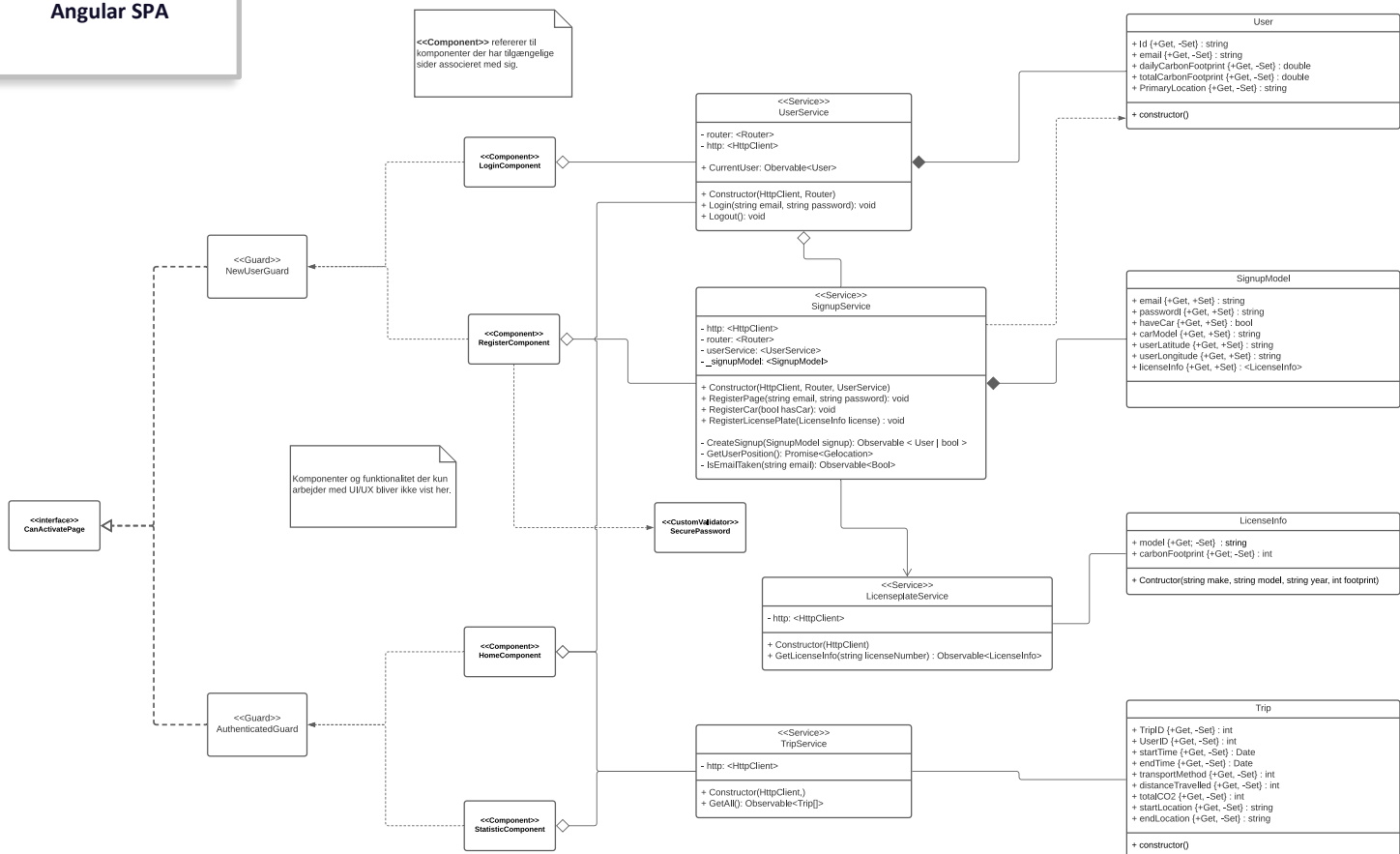
Desuden er der her blevet tænkt både SOLID principper, samt diverse designmønstre ind under udviklingen. Dette vil kunne ses, bl.a. med MVC mønstret i Backend API'en, samt Dependency Injection mønstret i Angular applikationen.

Yderligere noter, vil kunne ses indlejret på de enkelte diagrammer.



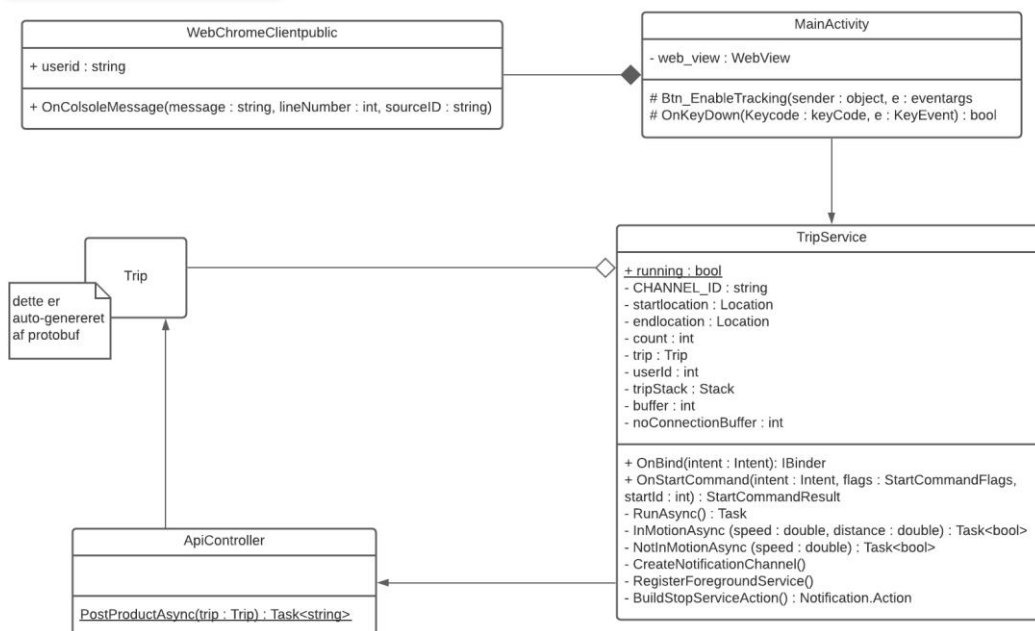
MVC API klassediagram

Angular SPA



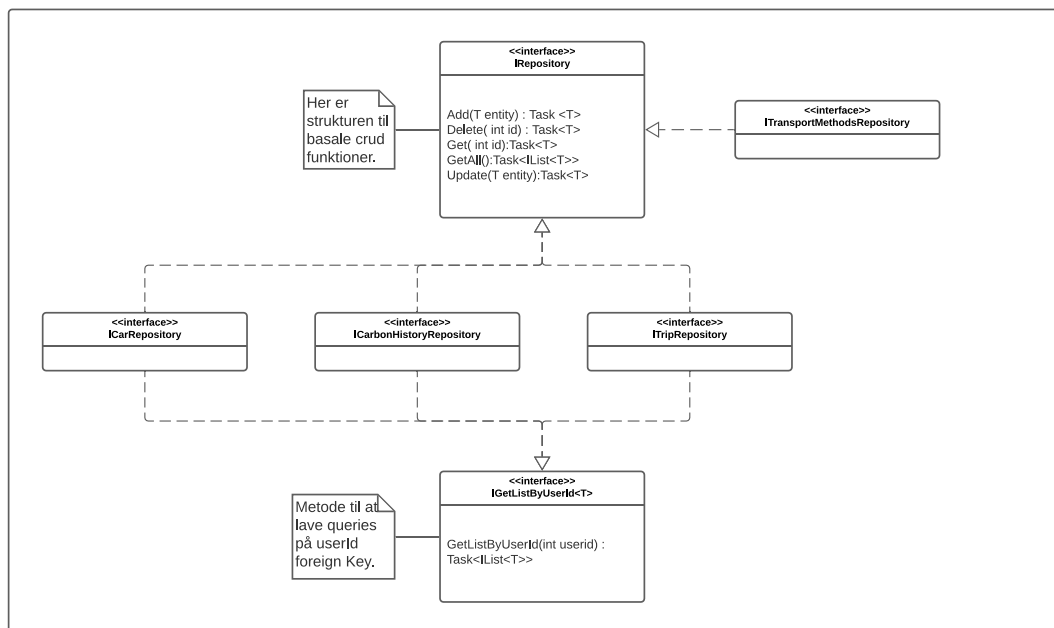
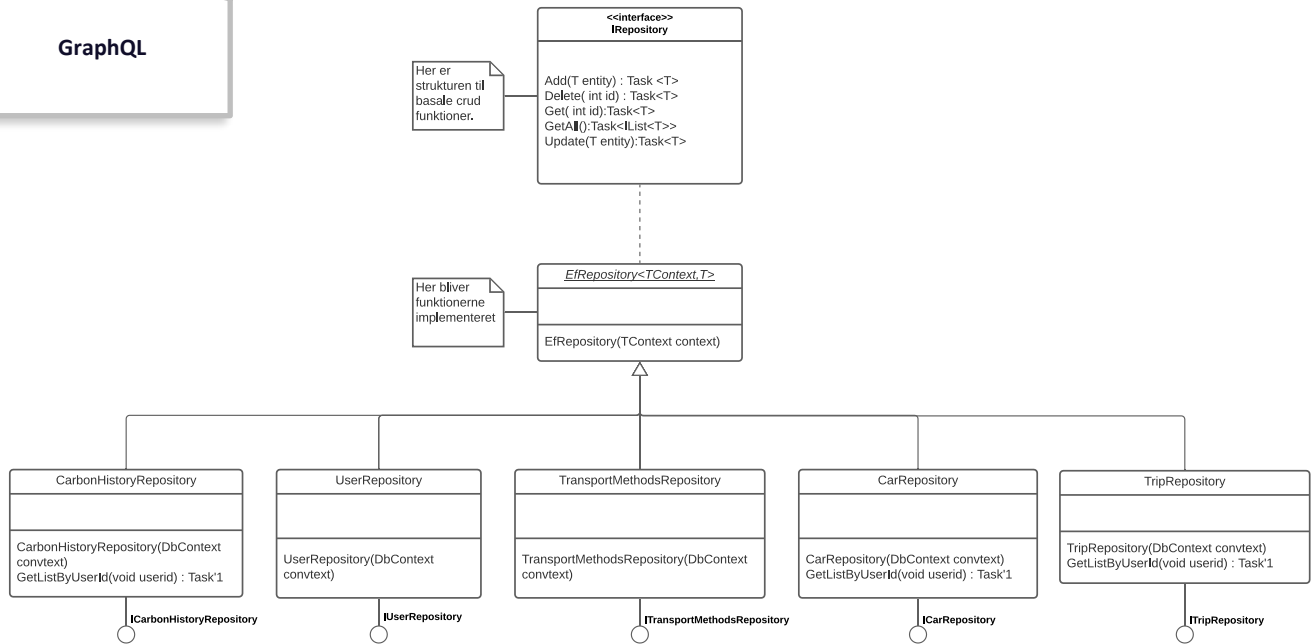
Angular SPA klassediagram

Xamarin Applikation



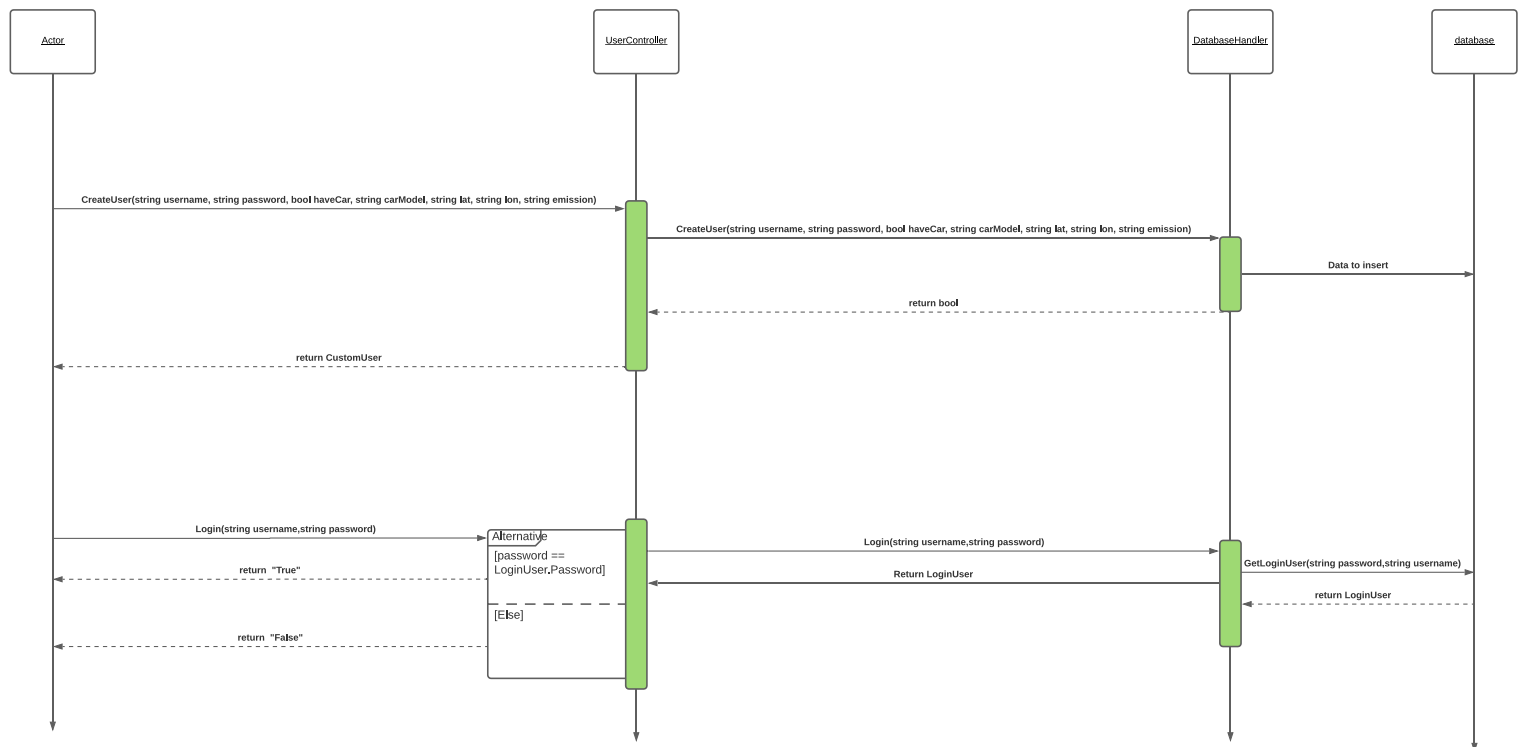
Xamarin applikation klassediagram

GraphQL



7.2 Sekvensdiagram

Sequence Diagram
MVC/Backend



9. Implementering

9.1 Deployment diagram

Vi har udarbejdet et deployment diagram, som skal visualisere hvordan vores deployment miljø skal opsættes. Vi har, som vist, valgt at køre en ESXi Hypervisor, hvorpå vi lægger en Windows server 2019 OS. Vi vælger at køre både vores MSSQL 2019 database, samt hoste vores 2 websider, på samme server. Dette ville ikke være den korrekte løsning i et "virkeligt" scenarie, men i vores tilfælde valgte vi at arbejdstiden vi ville spare ved at køre det samlet, var vigtigere.

