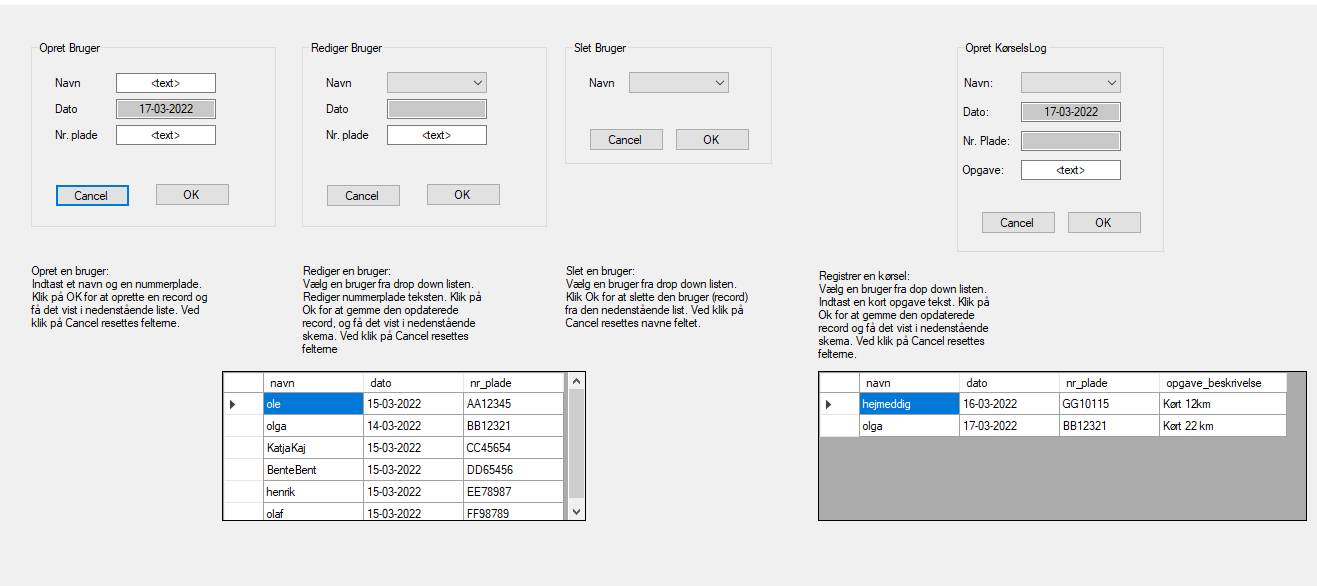
Tur-Retur KørselsLog Rapport

Navn - Jonas Barigo Østergaard

Uddannelse - Datatekniker med speciale i Programmering

Instruktør - Simon Nicolas El Hanafi



Indhold

[Indledning 3](#_Toc100140625)

[Dokumentation 4](#_Toc100140626)

[Konklusion 9](#_Toc100140627)

[Logbog 10](#_Toc100140628)

# Indledning

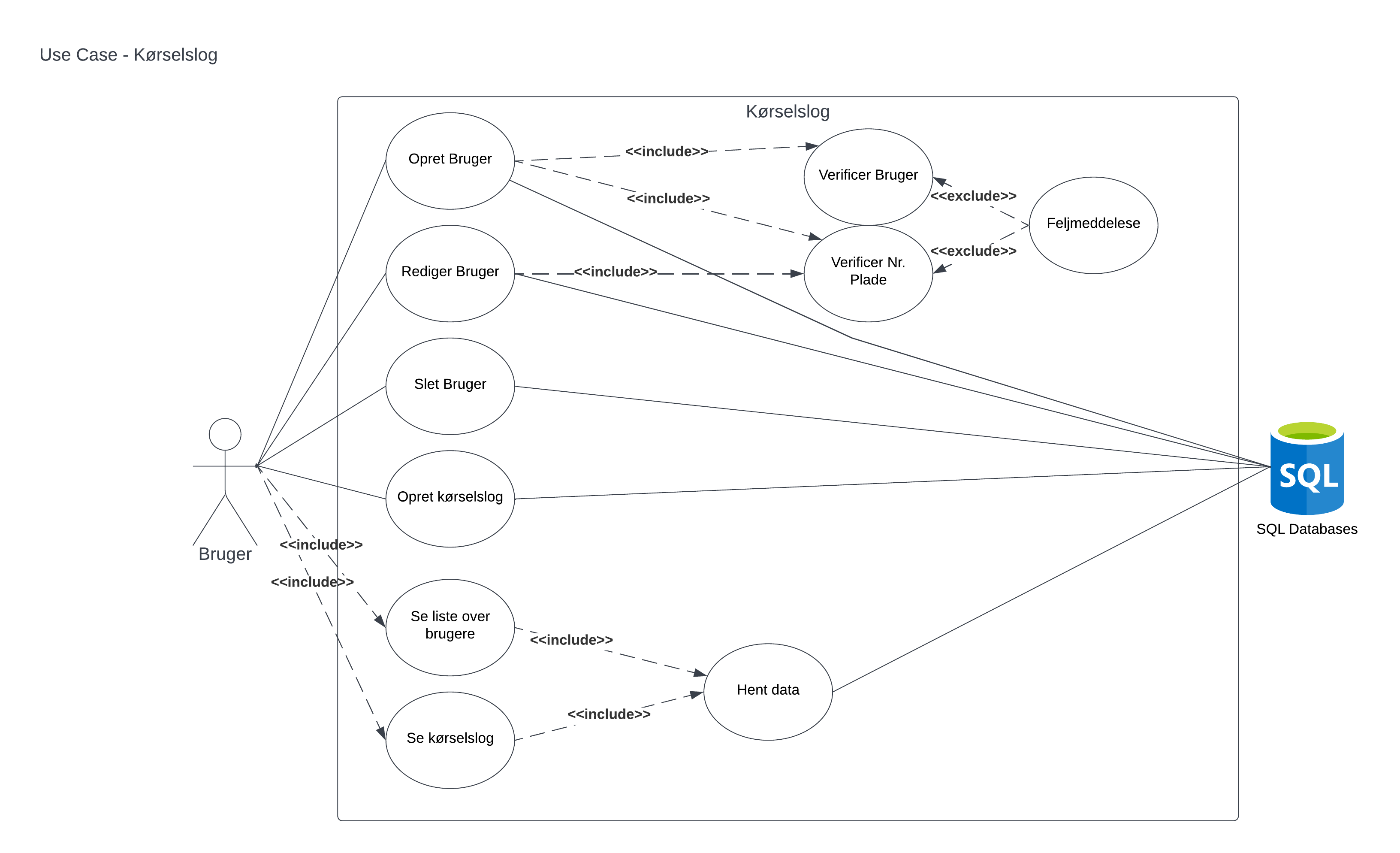
I denne case er jeg blevet bedt om, at lave et windows forms program, der skal kobles til en sql database. Her skal programmet kunne indsætte, redigere, slette og vise data fra databasen. Databasen skal indeholde data der beskriver brugere, dato, nummerplade og beskrivelse af en opgave, som brugeren har udført.

Til opgaven har jeg anvendt OOAD metoder for at analysere og specificere programmet, herunder user case, flowchart, klasse diagram, e/r diagram.

Derudover har jeg anvendt sql scripts, til at oprette min database og tabeller, og objekt orienteret programmering i c#, til at udvikle min brugergrænseflade.

Derudover har jeg også udarbejdet et Work Break Down (WBD) skema for at få afgrænset de enkelte underopgaver til mit program og på denne måde få mere struktur for hvornår og i hvilken rækkefølge jeg skal løse de enkelte opgaver.

# Dokumentation

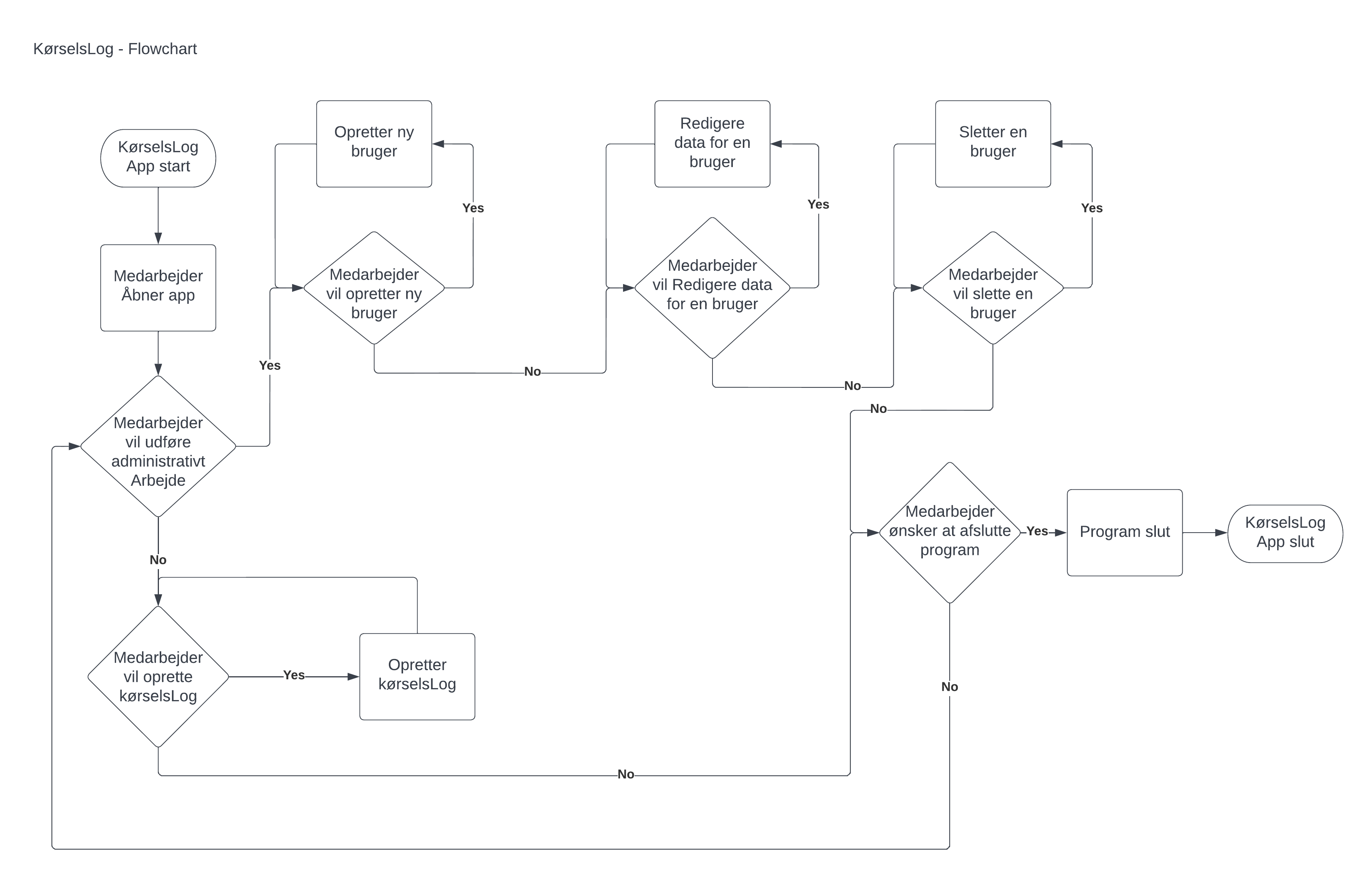


(Use case)

I denne use case er der en bruger. Denne bruger har valget mellem at udføre administrative opgaver eller oprette en kørsels log. For de administrative opgaver har brugeren valget mellem at oprette en bruger, redigere en bruger eller slette en bruger. Desuden kan brugeren også se en liste med oprettede brugere. Hvis brugeren ikke vil udføre administrative opgaver, er der kun et valg tilbage, nemlig at oprette en kørselslog.

Når data bliver oprettet eller opdateret, bliver de desuden verificeret af systemet automatisk, altså skal brugeren ikke foretage sig noget. Her bliver data for brugeren verificeret. Hvis verifikationen ikke går igennem kommer der en fejlmeddelelse, dette er dog kun hvis data ikke er korrekt, altså vil der ikke altid automatiske komme en fejlmeddelelse.

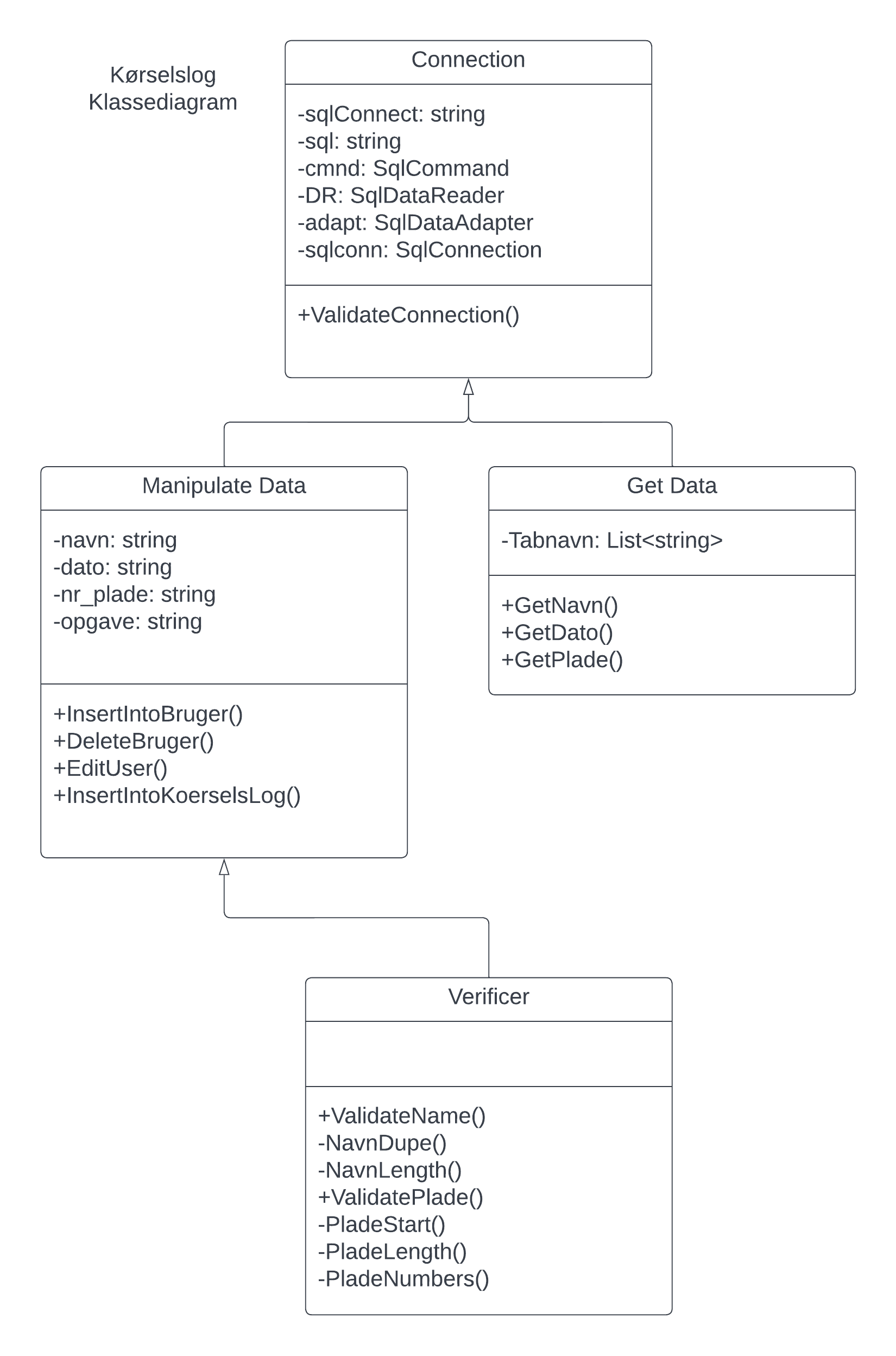
Brugeren har desuden ikke et valg om at få vist data, dette er noget der sker automatisk uden input fra brugeren. Data bliver desuden også automatisk hentet fra serveren.



(flowchart)

Dette flowchart tager udgangspunkt i start af Kørselslog appen, efter medarbejderen har åbnet appen, har medarbejderen et valg mellem at udføre administrativt arbejde eller om brugeren vil oprette en kørselslog. Hvis medarbejderen kun vil oprette en kørselslog, kan medarbejderen gøre dette. Medarbejderen kan desuden blive ved med at oprette kørselslogs indtil medarbejderen er tilfreds.

Hvis medarbejderen derimod vil udføre administrativt arbejde. Har medarbejderen valget mellem at oprette en ny bruger, redigere data for en bruger eller slette en bruger. Medarbejderen kan desuden gøre dette indtil medarbejderen er tilfreds. Når medarbejderen er færdig med de administrative opgaver, kan medarbejderen vælge at lukke programmet, eller starte forfra ved valget om at udføre administrativ arbejde, eller oprette en kørselslog.



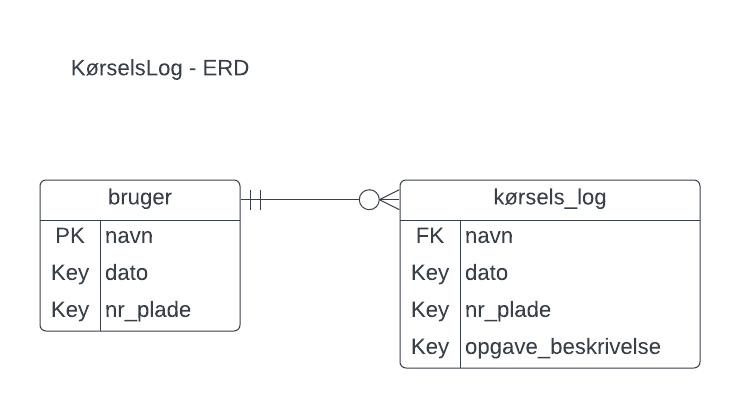
(Klassediagram)

I dette klassediagram kan man se klasserne Connection, Manipulate Data, Get Data ogVerificer. Connection er en klasse der tager sig af forbindelsen mellem programmet og SQL databasen, samt at initialisere commandoer der bruges i sammenhæng mellem SQL databasen og C# programmet.

Manipulate Data klassen tager sig af at indsætte, opdatere og slette data i SQL databasen. Her arver den egenskaber fra connection klassen og benytter forbindelsen idet klassens metoder bliver anvendt.

Get data klassen arver også fra connection klassen. Get data klassens formål er at hente data fra SQL databasen og vise dem i C# programmet.

Til sidst er der Verificer Klassen. Verificer klassen arver fra Manipulate data. Verificer klassen har til formål at verificere om de data der bliver oprettet og opdateret lever op til de krav jeg har sat. Inden de bliver indsat I SQL databasen.

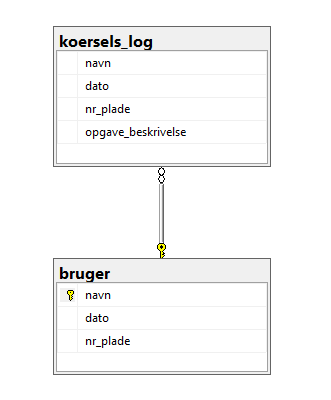


(ERD)

I dette Entity Relationship Diagram, kan det ses at jeg har to tabeller i min database. Den første tabel ’bruger’ har en primary key (PK) som har navnet ’navn’. Denne value skal være unik da det er en primary key. Denne priarmy key er en unik identifikation og kan derfor ikke have værdier der er den samme. Dette betyder at hvis der bliver oprettet en bruger ved navn Jonas I min tabel, kan der ikke oprettes flere brugere med navnet Jonas. Desuden bruges primary key’en som et reference punkt af andre tabeller.

I den anden tabel ’kørsels\_log’ har vi næsten de samme values som i bruger. Her er navn dog en Foreign key, og der er en ny value kaldet ’opgave\_beskrivelse’. Navn har en reference til tabellen bruger og valuen navn.

Relationen mellem de to tabeller er som følgende. For hver brugere kan der eksistere 0 eller flere kørsels\_log, men for hver kørsels\_log kan der kun eksistere 1 og kun 1 bruger.



(DB diagram)

Her ses et database diagram. Tabellen bruger har 3 values. Navn, dato og nr\_plade. Navn er en Primary Key og har derfor et nøgle ikon ved siden af sig. Tabellen koersels\_log har 4 values. Navn, dato, nr\_plade og opgave\_beskrivelse. Navn er her en foreign key or har en reference til primary key’en navn i tabellen bruger.

# Konklusion

I denne opgaver har jeg arbejdet med OOAD, SQL og OOP. Jeg har brugt disse til at oprette diagrammer over den case jeg har fået, oprette en database udfra de UML diagrammer jeg har lavet og til sidst programmeret en windows forms application, til at varetager en kørselslog. I mit program har jeg derfor skulle forbinde windows forms application’en til min SQL database, hvilket jeg fandt lidt udfordrende til at starte med, men endte op med at finde det spændende. Jeg blev nødt til at ændre i portene og rettighederne for min database på min windows server i den virtuelle maskine jeg anvendte.

For at forbinde C# og SQL Databasen brugte jeg System.Data.SqlClient, da jeg her kunne bruge kommandoer egnet til at forbinde til sql databasen, samt manipulere databasen via inputs fra C#. Desuden har jeg anvendt CRUD princippet. Dette kommer til udtryk i min grændsflade, hvor der er muligheder for at oprette data, læse data, opdatere data og slette data fra min sql database.

# Logbog

08-03-2022

I dag startede jeg med at læse og forstå opgaven, samt start af de første diagrammer. Først lavede jeg use case og derefter flowchart. Derefter oprettede jeg et E/R diagram, dette nåede jeg ikke at blive færdig med i dag.

09-03-2022

I dag startede jeg med at lave resten af E/R diagrammet, derefter lavede et database diagram og til sidst lavede jeg databasen via et script. I databasen oprettede jeg desuden to tabeller vi det same script, som jeg oprettede databasen med.

10-03-2022

I dag startede jeg med at lave et klasse diagram over databasen, hvorefter jeg startede på mit windows forms program. Da jeg ikke har beskæftiget mig med windows forms før, satte jeg mig til at finde oplysninger på nettet om hvordan man gør. Derudover satte jeg windows forms op således at det lignede mockup’en vi fik udleveret i opgaven.

11-03-2022

I dag fortsatte jeg med at opsøge viden omkring windows forms og hvordan man forbinder til en database i C#. Derudover begyndte jeg for alvor at skrive programmet og teste den viden jeg havde fundet af. Her fik jeg C# til at hente mine tabeller’s data og jeg lavede funktioner der tjekkede mine data for at se om de levede op til nogle krav jeg selv fandt på.

14-03-2022

I dag fik jeg lavet en metode der oprettede en bruger i sql databasens tabel. Her indtaster man et navn og en nummerplade trykker på OK knappen og dataen bliver så indsat i databasen.

15-03-2022

I dag fik jeg lavet en metoder til at redigere en eksisterende bruger, i databasen. Derudover fik jeg også lavet en metode til at slette en bruger. Til sidst anvendte jeg også et datagridview for at få vist de tabeller jeg har oprettet og deres indtastede data.

16-03-2022

I dag fik jeg oprettet en metode til at indsætte data for kørselsloggen. Derudover fik jeg også vist den i et datagridview. Derudover lavede jeg også metoder til at verificere om der er forbindelse til serveren, samt om nr. pladen er det rigtige format og om brugerens navn allerede eksistere i databasen.

17-03-2022

I dag itererede jeg på nogle af de metoder jeg havde lavet og andre dele af programmet. For at se om jeg kunne gøre dem bedre eller lave dem på en smartere måde.

18-03-2022

I dag fortsatte jeg med iterationen af koden. Her ændrede jeg min opdatering af navn i drop down menuen. Førhen slettede jeg en liste med alle navnene i og skrev dem alle sammen ind i listen igen via et loop. Dette ændrede jeg til at det navn som jeg skriver ind i tabellen bliver tilføjet til min liste uden at hente det fra databasen.

Derefter gik jeg i gang med at lave rapporten.