2018

Alexander, Anton og Mathias W.

TEC Ballerup

03-10-2018

AAM AutoShop



Indholdsfortegnelse

[Indledning 2](#_Toc526328066)

[UML: 3](#_Toc526328067)

[Use Case: 3](#_Toc526328068)

[E/R Diagram: 6](#_Toc526328069)

[Class Diagram: 7](#_Toc526328070)

[Mockups: 8](#_Toc526328071)

[Kravsspecifikation 11](#_Toc526328072)

[SQL Database Source Code: 15](#_Toc526328073)

[Source Code SQL Database 15](#_Toc526328074)

[Tidsplan: 17](#_Toc526328075)

[Konklusion: 18](#_Toc526328076)

[Hvordan er det gået 18](#_Toc526328077)

[Hvis Det skulle havde været bedre 18](#_Toc526328078)

[Hvad vi syntes om projektet som helhed 18](#_Toc526328079)

[Hvad kunne havde været bedre? 18](#_Toc526328080)

[Dagbog: 19](#_Toc526328081)

# Indledning

Lærerene har valgt at vi skal arbejde med et autoværksted, dets kunder og deres biler.

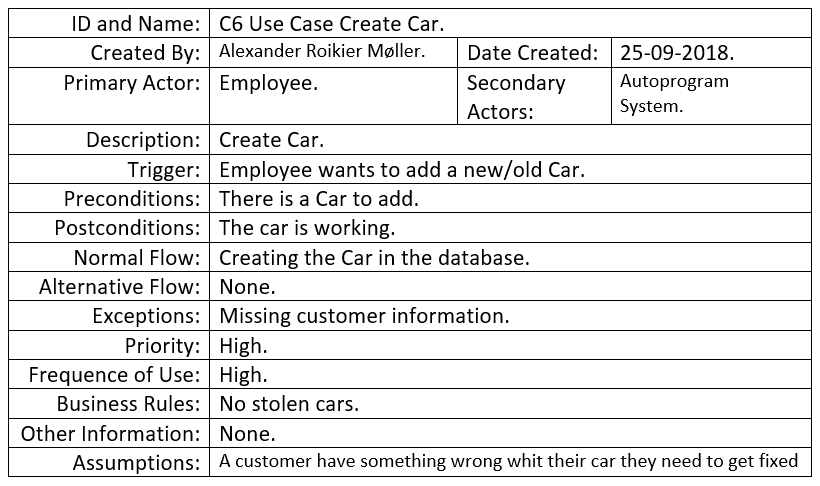
Der er følgende ønsker til systemet (kravsspecifikation):

1. Man skal kunne oprette en kunde, opdatere og slette en kunde
2. Man skal kunne få vist en kundeoversigt (herunder en sorteret form efter efternavn, eller sorteret biler) 3. Man skal kunne få vist en kundes biler.
3. Man skal kunne oprette, opdatere, slette og vise en bil (CRUD).
4. Hver bil skal have reg.nr., mærke, model, årgang, km, brændstoftype find selv på flere.
5. En kunde kan godt have flere biler.
6. En bil kan kun være ejet af én kunde
7. Både kunder og biler skal have en oprettelsesdato.
8. Man skal kunne oprette/rette/slette et værkstedsbesøg for en bil.
9. Det skal være muligt at vise en bils værkstedsophold/datoer.

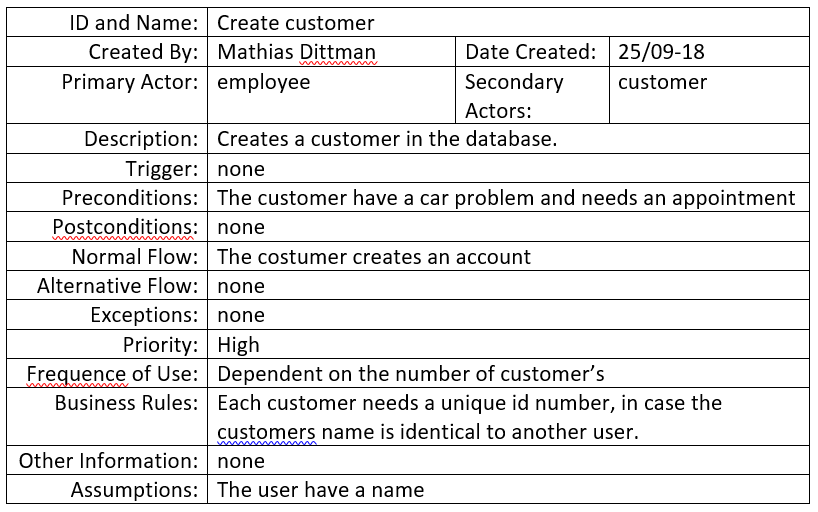
# UML:

## Use Case:

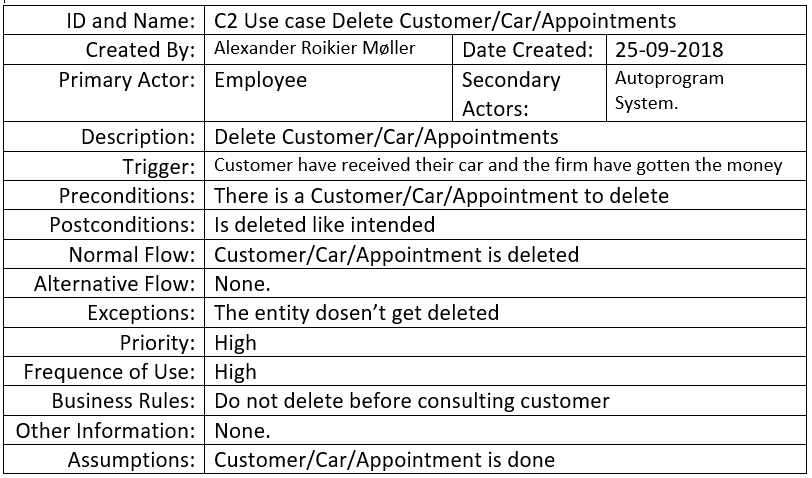
Use case: Create Car



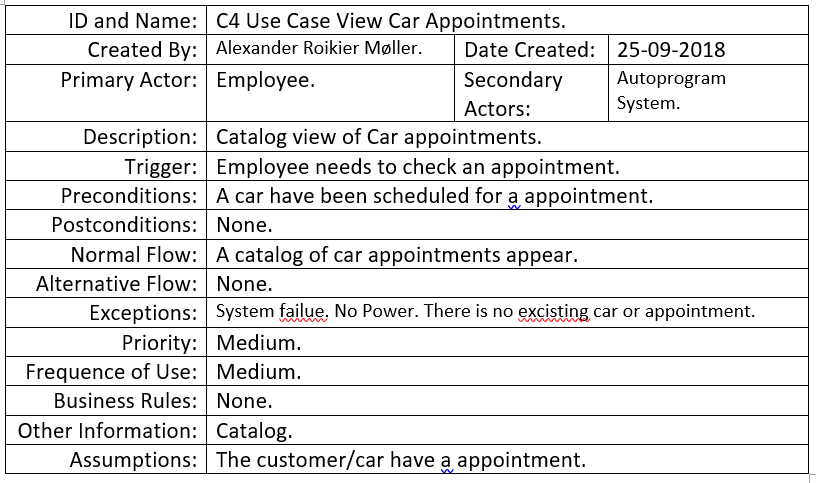
Use case: Create Customer



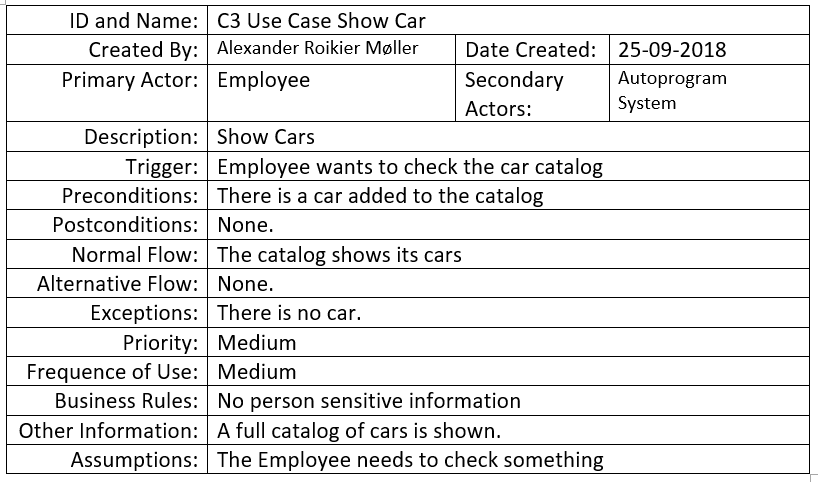
Use case: Delete Customer/Car/Appointments



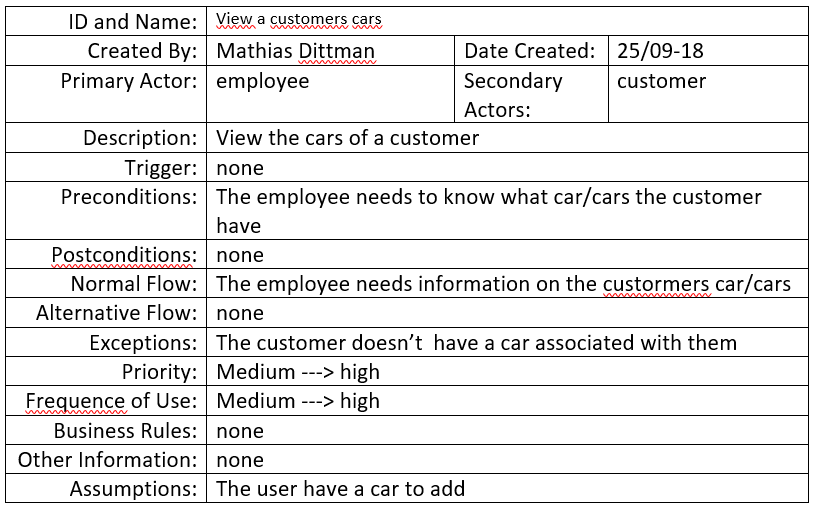
Use case: Show Car Appointments



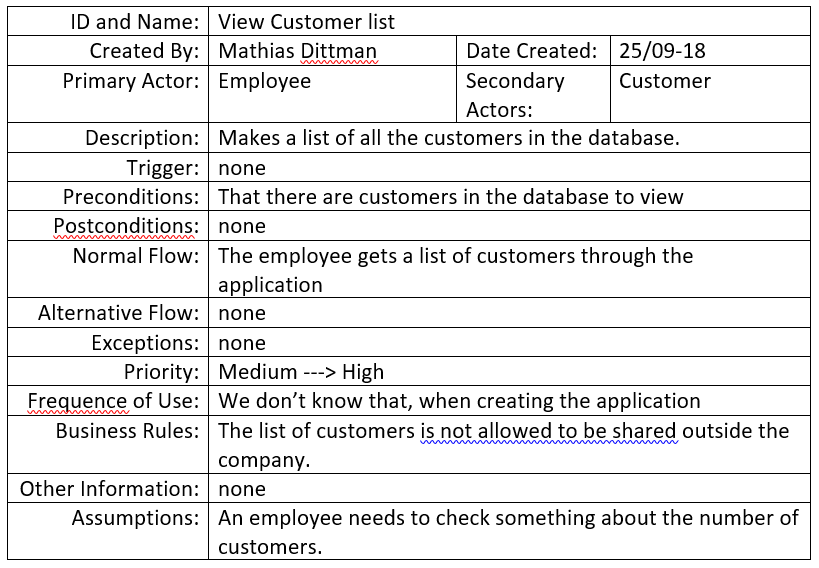
Use case: Show Car



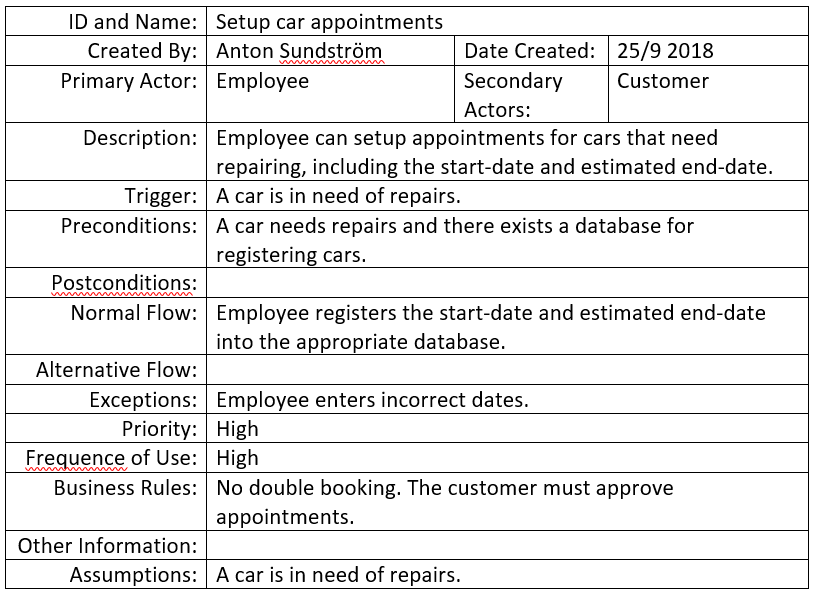
Use case: View A Customers Cars



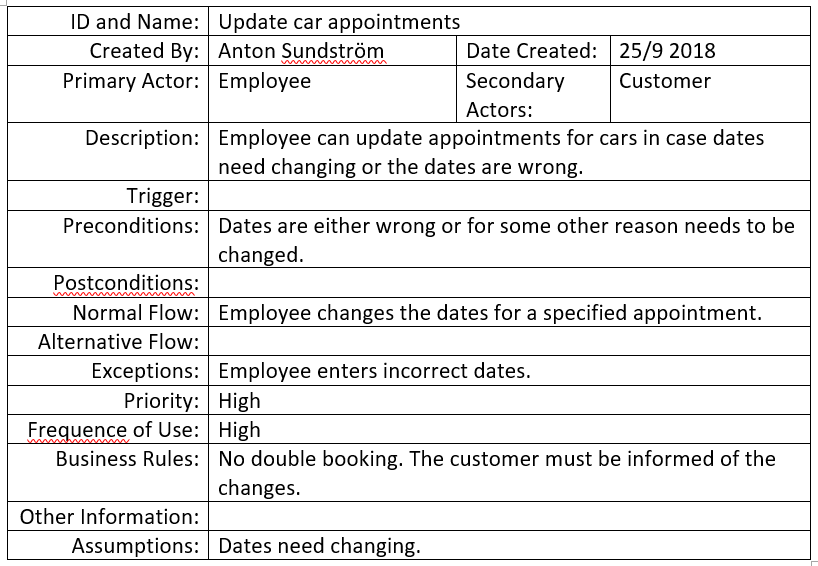
Use case: View Customer List



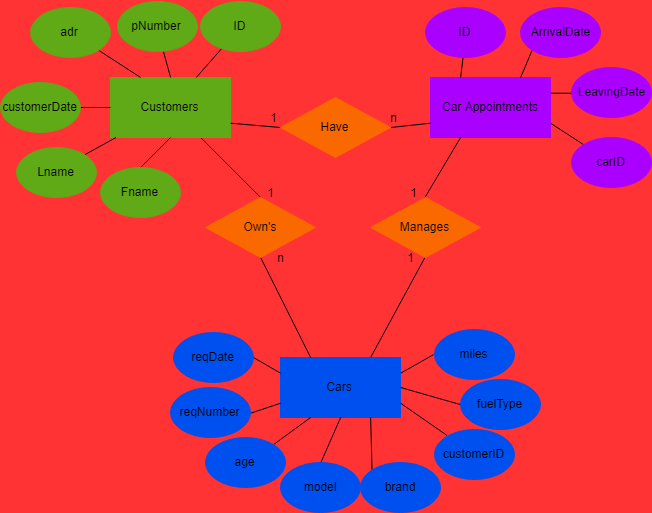
Use case: Setup Car Appointments



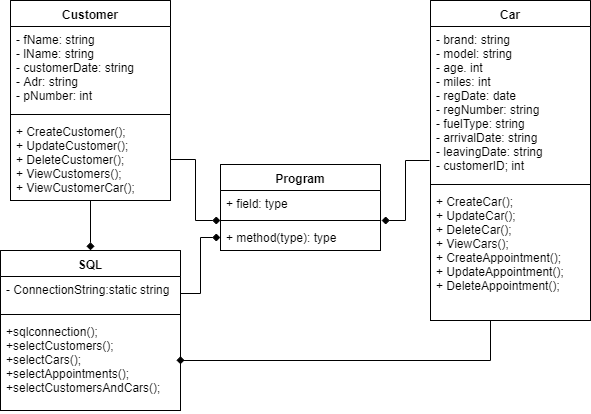
Use case: Update Car Appointments



## E/R Diagram:

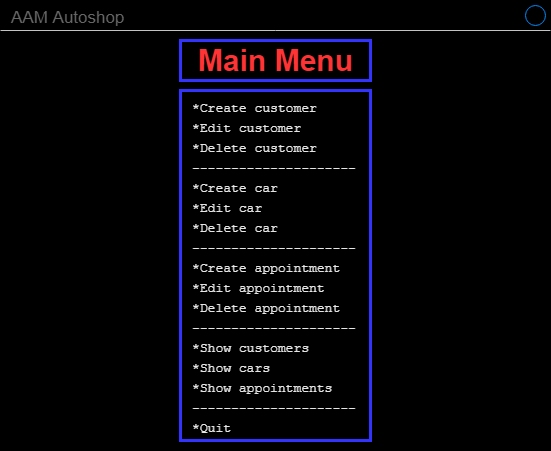


## Class Diagram:

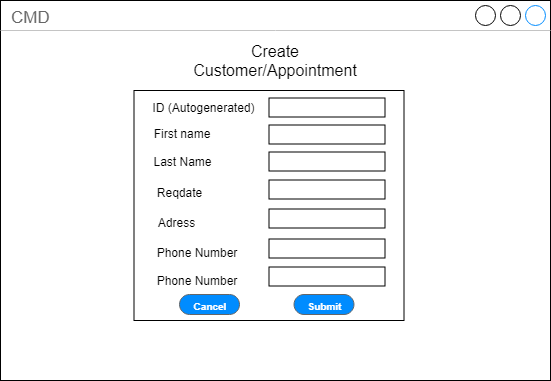


## Mockups:

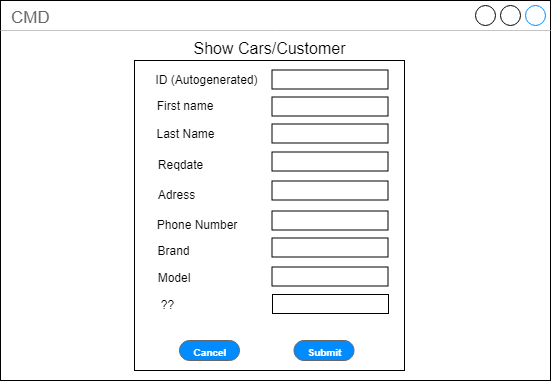
Main Menu:



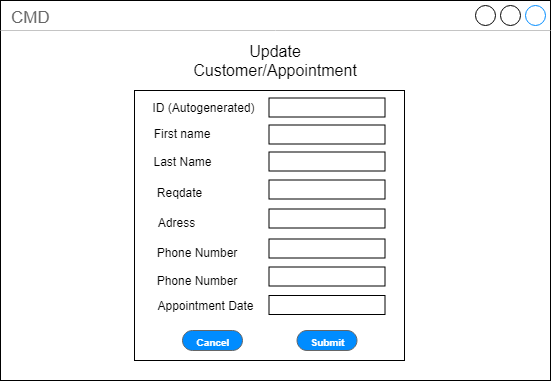
Create Customer/Appointment:



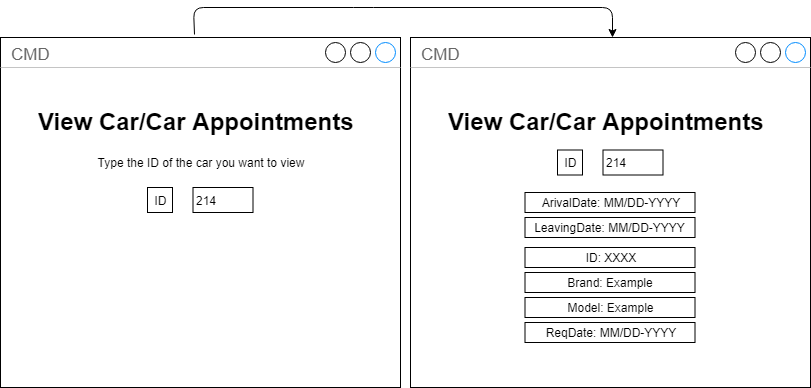
Show Cars/Customer:



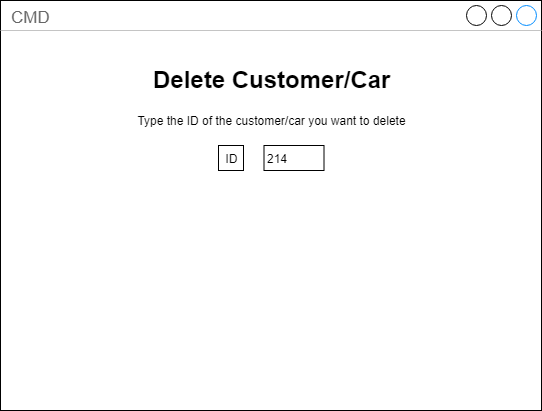
Update Customer/Appointment:



View Car/Car Appointment:



Delete Customer Car:



## Kravsspecifikation

##### 1. Indledning

1.1 Formål

* Hvad skal udvikles.
  + En system app til et værksted
* Hvad skal det bruges til.
  + Værkstedet skal bruge det til indregistrering af kunder, biler mm.
* Hvem er udvikleren.
  + AAM
* Hvem er kunden.
  + AutoShop

1.2 Referencer

* Kundens oplæg.
  + Man skal kunne oprette en kunde, opdatere og slette en kunde
  + Man skal kunne få vist en kundeoversigt (herunder en sorteret form efter efternavn, eller sorteret biler)
  + Man skal kunne få vist en kundes biler.
  + Man skal kunne oprette, opdatere, slette og vise en bil (CRUD).
  + Hver bil skal have reg.nr., mærke, model, årgang, km, brændstoftype find selv på flere.
  + En kunde kan godt have flere biler.
  + En bil kan kun være ejet af én kunde
  + Både kunder og biler skal have en oprettelsesdato.
  + Man skal kunne oprette/rette/slette et værkstedsbesøg for en bil.
  + Det skal være muligt at vise en bils værkstedsophold/datoer.

#### 2. Generel beskrivelse

2.1 Systembeskrivelse.

* Verdensbillede, en tegning af det totale system.
* Kort beskrivelse af hardwaren.
  + Der skal bruges en PC
* Kort beskrivelse af softwaren.
  + Visual Studio, SQL og DrawIo

2.2 Programmellets funktion

* En lidt grundigere beskrivelse af programmet, uden at berøre specifikke krav til detaljer.
  + Programmet består af en Menu, med funktioner til oprettelse, opdatere og fjernelse af kunder, biler og værkstedsbesøg

2.4 Programmellets fremtid

* Programmets forventede levetid.
  + Indtil udvidelse af firmaet, med flere firma adresser.
* Udvidelsesmuligheder i senere versioner. Er der konkrete ting, der skal tages højde for nu.
  + Nej

2.5 Brugerprofil

* Hvem skal bruge systemet.
  + De ansatte i firmaet
* Stilles der krav til erfaring med edb-udstyr.
  + Ja, basale it-færdigheder

2.6 Krav til udviklingsforløbet

* Krav fra såvel udvikleren som kunden.
  + Præcis Kommunikation
* Anvendelse af vejledninger, designmetoder, standarder.
  + En simpel manual til brug af programmet.
* Programmeringssprog.
  + C#
* Hvilken dokumentation, skal der udarbejdes.
  + Kravspecifikation
  + Mockup af program
  + Use case
  + Manual til brug af programmet
* Hvordan skal ændringer i kravspecifikationen håndteres.
  + De skal oplyses til kunden (AutoShop)

2.7 Omfang af kundeleverence

* Hvor meget af det samlede system skal leveres til kunden.
  + Det kompileret program
* Hvor meget af dokumentationen skal leveres til kunden.
  + Alt, samlet i et stort dokument

2.8 Forudsætninger

* Udstyr som kunden skal stille tilrådighed under udviklingen.
  + Intet
* Personer som kunden skal stille til rådighed.
  + Test personer, til test af brugervenlighed

#### 3. De specifikke krav

3.1 Definitioner

* Formatet på væsentlige data, som kunden ønsker fastlagt fra starten.
  + Tidsfomat skal være Gmt+1

3.2 Funktionelle krav

* Beskrivelse af hver af de funktionaliteter, som programmet består af. Det være sig funktionaliteter, som brugeren oplever, men også funktionaliteter, der er væsentlige for programmets funktion.
  + Man skal kunne oprette en kunde, opdatere og slette en kunde
  + Man skal kunne få vist en kundeoversigt (herunder en sorteret form efter efternavn, eller sorteret biler)
  + Man skal kunne få vist en kundes biler.
  + Man skal kunne oprette, opdatere, slette og vise en bil (CRUD).
  + Hver bil skal have reg.nr., mærke, model, årgang, km, brændstoftype find selv på flere.
  + En kunde kan godt have flere biler.
  + En bil kan kun være ejet af én kunde
  + Både kunder og biler skal have en oprettelsesdato.
  + Man skal kunne oprette/rette/slette et værkstedsbesøg for en bil.
  + Det skal være muligt at vise en bils værkstedsophold/datoer.

#### 4. Eksterne grænseflade-krav

4.1 Bruger-grænseflade

* Krav til måden programmet betjenes på: Menuer/ mus/ tastatur.
  + Mus, tastatur, skærm og PC.
* Forskellige brugeres rettigheder til brug af forskellige funktioner.
  + Nej, alle ansatte har samme rettigheder

4.2 Hardware-grænseflade

* På hvilken elektrisk form optræder informationerne.
  + I tekstform

4.4 Software-grænseflade

* Operativsystemet som programmellet skal køre under.
  + Windows xp, 7 og 10

#### 5. Krav til programmellets ydelse

* Specifikke tidskrav til udførelse af bestemte funktioner.
  + Det hele er lavet i CMD, så alt er med minimal ventetid
* Krav til det eksekverbare programs størrelse.
  + Nej

6.1 Pålidelighed

* Fejl i produktet.
  + Helst ikke
* Nøjagtighed.
  + Meget
* Håndtering af fejlbetjening.
  + Hvis fejl skulle opstå, håndteres de af Udviklings Teamet (AAM)

6.2 Vedligeholdelsesvenlighed

* Hvor lang tid tager det at finde en fejl.
  + Det tager ligeså lang tid, som at teste alle funktioner i programmet
* Hvor nemt er det at lave en mindre tilpasning til et ændret behov.
  + Kan være svært, angående tilpasningen

6.3 Udvidelsesvenlighed.

* Hvor nemt er det at lave en egentlig udvidelse af produktet.
  + Rimelig nemt, man kan altid lave flere metoder og klasser efter behov.

6.4 Brugervenlighed

* Hvor lang tid tager det for en ny bruger at lære at betjene produktet, mm.
  + Ikke så lang tid, programmet er meget overskueligt.

6.5 Genbrugbarhed

* Skal dele af programmet laves med henblik på at kunne bruges andetsteds.
  + Nej

6.7 Effektivitet

* Hvilke dele af produktet skal prioriteres høj effektivitet.
  + Alt prioriteres lige højt i programmet

# SQL Database Source Code:

## Source Code SQL Database

# use master

# if exists (select \* from sysdatabases where name = 'Autoshop')

# drop database Autoshop

# create database Autoshop

# go

# use Autoshop

# create table Customer (

# id int identity,

# fName nvarchar(50),

# lName nvarchar(50),

# customerDate nvarchar(50),

# adr nvarchar(50),

# pNumber int NOT NULL,

# constraint CustomerPK Primary Key(pNumber)

# )

# go

# create table Cars (

# id int identity,

# brand nvarchar(50),

# model nvarchar(50),

# age int,

# regNumber nvarchar(50) primary key NOT NULL,

# carDate nvarchar(50),

# miles int,

# fuelType nvarchar(50),

# customerID int foreign key references Customer(pNumber)

# on delete cascade

# on update cascade

# )

# go

# create table carAppointments (

# id int identity primary key NOT NULL,

# arrivalDate nvarchar(50),

# leavingDate nvarchar(50),

# carID nvarchar(50) foreign key references Cars(regNumber)

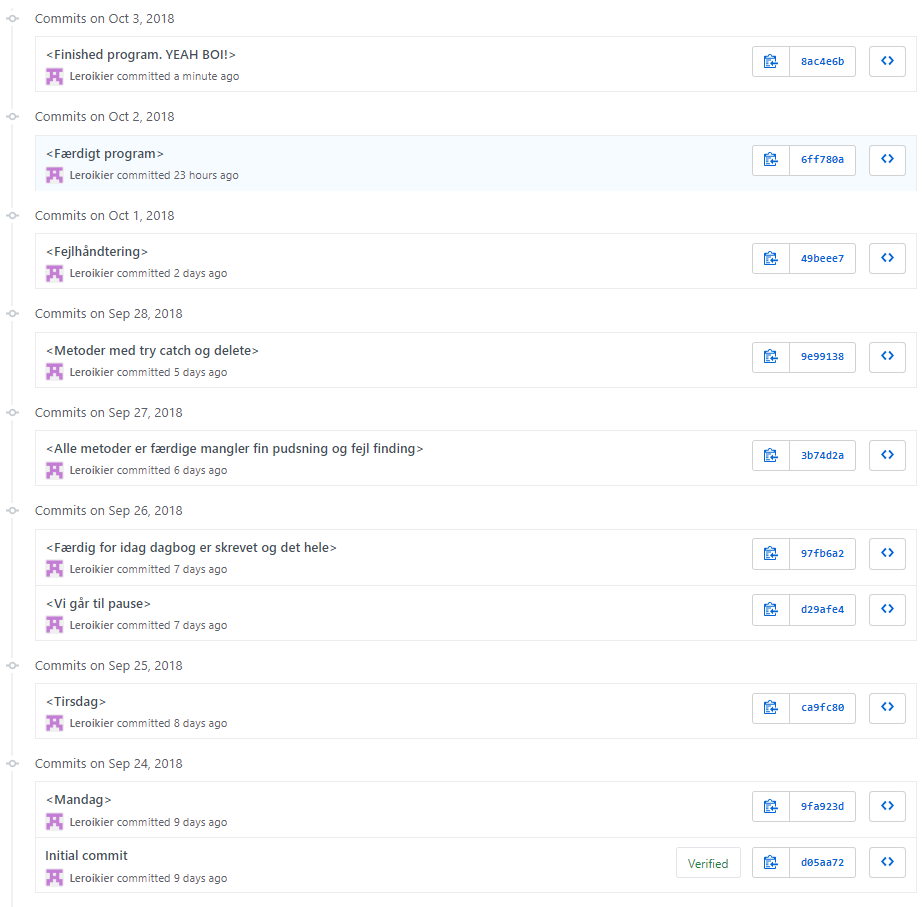
# on delete cascade

# on update cascade

# )

# Go

# Version styring (GitHub):



# Tidsplan:

Mandag = Teste SQL sammen kobling til C#, Skrive dagbog, Lave tidsplan

Tirsdag = Klasse diagram, ER diagram, Use cases.

Og hvis vi har tid til overs laver vi flere metoder til og hente/oprette personer/biler osv.

Onsdag = laver videre på metoderne i vores c# program

Thorsdag = fortsætter med kodning af programmet, eventuelt test/debugging af programmet.

Fredag = status dag, opfølgning hvad mangler vi og hvad har vi nået.

Mandag d. 1/10 = samler projektet, hvis vi er tæt på færdig med programmet, kigger vi på ekstra forbedringer såsom try/catch og fejlfinding.

Tirsdag = alt skal helst være færdigt inden fyraften, hvis vi ikke er, så bliver vi det!

Onsdag = aflevering!

# Konklusion:

## Hvordan er det gået

Vi konkludere at projektet som helhed er gået utrolig godt, vi har været gode til at fordele arbejdet, og samarbejde der hvor det var nødvendigt.

Vores planlægning har passet til punkt og prikke, og vi er meget tilfredse med det endelige resultat af vores program.

## Hvis Det skulle havde været bedre

Hvis vi skulle udvide vores program med mere, så snakkede vi om at oprette en ny tabel med de informationer som bliver slettet, således at man har en historik over kunder, biler og aftaler som har forgået på værkstedet.

Så snakkede vi om en form for søgefunktion, hvor man kunne søge efter et navn, og så få vist alle kunder med det navn, og på den måde få en god søgefunktion, hvor man hurtigt kan finde en kundes id og lign.

## Hvad vi syntes om projektet som helhed

Vi syntes projektet som helhed var et rigtigt godt projekt, som tager fadt i rigtig mange vigtige dele af arbejdsforløbet, når man skal lave et nyt program til en kunde. Og projektets krav var godt formuleret og derved ikke til at tage fejl af, og kravene der blev stillet til os var rimelige og fair.

## Hvad kunne havde været bedre?

Tja, måske at man selv havde kunne vælge tema eller lign, så alle projekterne ikke var så ens, og man blev lidt mere udfordret i at man selv skulle være lidt opfindsom.

# Dagbog:

Mandag 24-09

Vi startede så småt med at diskutere hvilke variabler vores SQL database skulle indeholde.

Efterfølgende oprettede vi vores sql Querys, og fik inkorporeret det i vores C# Visual studios

Vi fik også oprettet vores metoder (Create, Select, Customers) uden yderligere problemer.

Og til sidst men ikke mindst lavede vi en overordnet tidsplan.

Tirsdag 25-09

Vi startede med at lave use case’s, user stories, E/R Diagram , Class Diagram og mockup af vores Console program.

Derefter lavede vi vores kravspecifikation også arbejdede vi videre på vores program i Visual Studio.

Onsdag 26-09

i dag har vi lavet mange af de metoder som vores program skal indeholde, det fleste fik uden problemer, der var få små ting som vi tjekkede op på, på nettet løbende.

i dag har vi lavet

* Create customer
* Update customer
* View customer list
* View a customers cars
* Create car

Torsdag 27-09

I dag lavede vi videre på vores metoder i visual studio, vi blev færdig med alle metoderne til vores program.

* Delete customer
* Update car
* Delete car
* Update car appointments
* Delete car appointments
* Create car appointments
* View a cars appointments

Fredag 28-09

Vi har brugt fredagen på at diskuterer hvad vi mangler og hvad der er vigtigst at lave af ekstra funktioner i vores program.

Ellers har vi lavet try catch på de fleste af vores metoder, så vi forbedre brugervenligheden af vores program.

Efterfølgende har vi findpudset opsætningen af programmet.

Vores tidsplan har holdt indtil nu, nøjagtigt som planlagt.

Mandag 01-10

Det meste af mandagen gik ved fejlfinding, vi brugte lidt tid på at lave tjek af indtastning, på telefon nr. mm.

Tirsdag 02-10

Mere fejlfinding i programmet for finpudsning, så arbejdede vi på metoderne til at tjekke indtastninger i programmet, f.eks. så man ikke kan indtaste bokstaver i tlf. nummeret.

Så lavede vi dokumentationen i Word, som vi så kan aflevere onsdag.

Onsdag 03-10

i dag har vi gennemgået programmet for fejl og mangler, og færdiggjort programmet, så det er klar til aflevering inden klokken 15:00.

derudover har vi lavet dokumentationen færdig og oprettet test brugere til programmet.