# Opgave: Uge 3 – Anton’s Auto

Fag: Udviklingsmiljøer

Udarbejdet af: Lars Larsen

### Vi har fået stillet følgende opgave

Antons Auto ønsker en hjemmeside, hvorfra de kan administrere deres bilpark.

Tanken er at applikationen kun kører på deres interne netværk, så du behøver ikke tænke authentication, log in system mv. ind i app'en.

Antons Auto skal bruge applikationen internt, således at de forskellige medarbejdere hurtigt og nemt kan finde oplysninger om hvilke biler, de har i bilparken pt.

Oplysningerne kan for eksempel bruges i samtale med kunde.

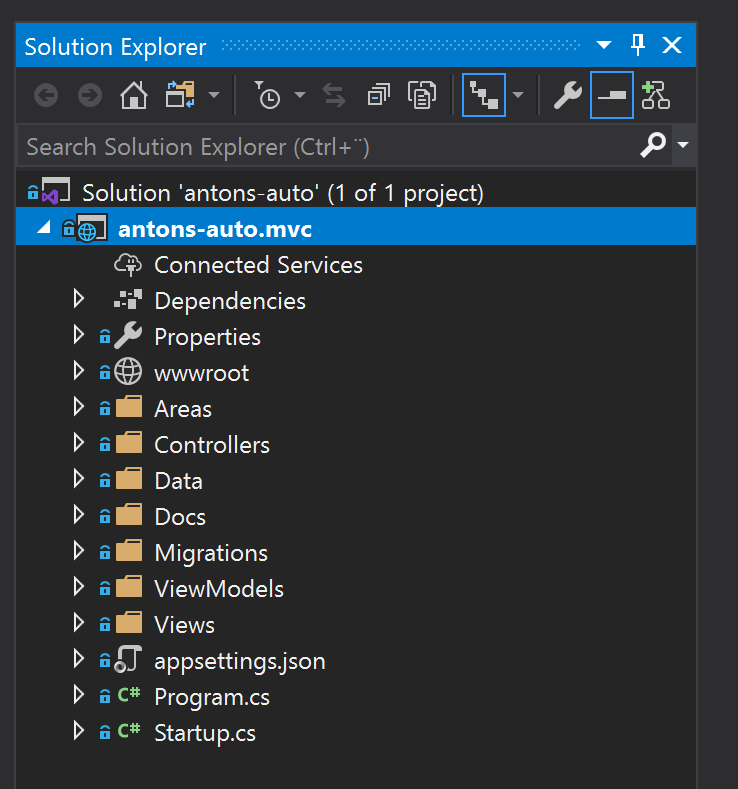
Alle oplysninger, der ikke fremgår af ovenstående, står til fri fortolkning.

#### **Acceptance Criteria**

* Du skal lave det som en EF Core MVC app med tilhørende database
* Fra denne skal man kunne oprette bilmærker (Audi, BMW, Volvo mv)
* Man skal også kunne oprette modeller baseret på det specifikke mærke.
* Man skal kunne se alle bilmærker og alle modeller.
* Man skal kunne se detalje visning for alle bilmærker og modeller.

## Løsning

Anton’s Auto er lavet som en ASP.Net core MVC løsning. I nedenstående er mappe strukturen

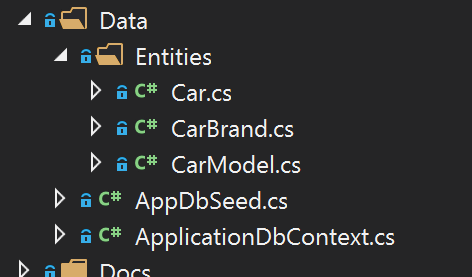


Følgende beskrives de mapper der afviger fra standard skabelonen som Visual Studio generer.

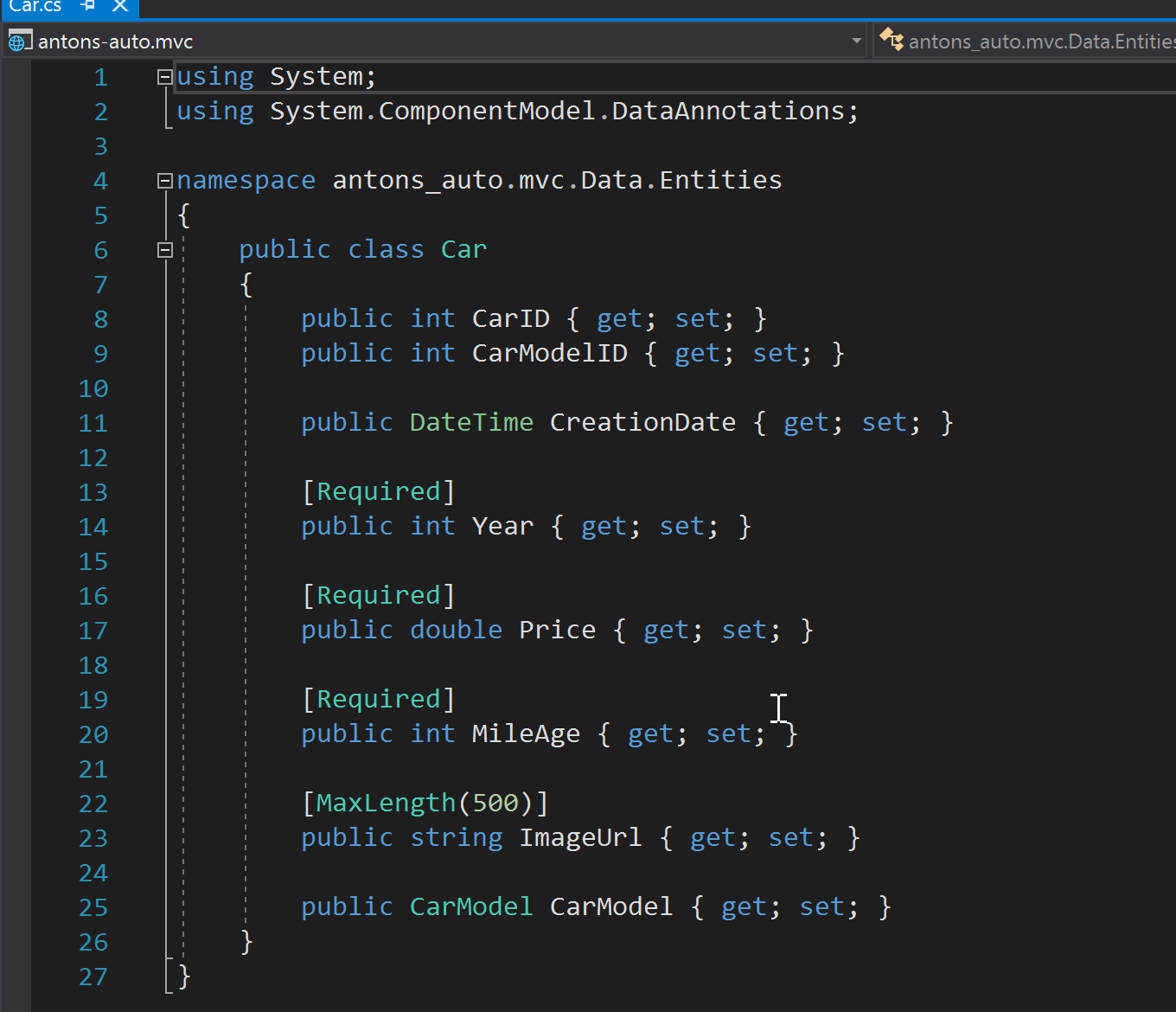
* Data/Entities => Indeholder klasser der beskriver database modellen
* ViewModels => Indeholder klasser der benyttes til views

## Data

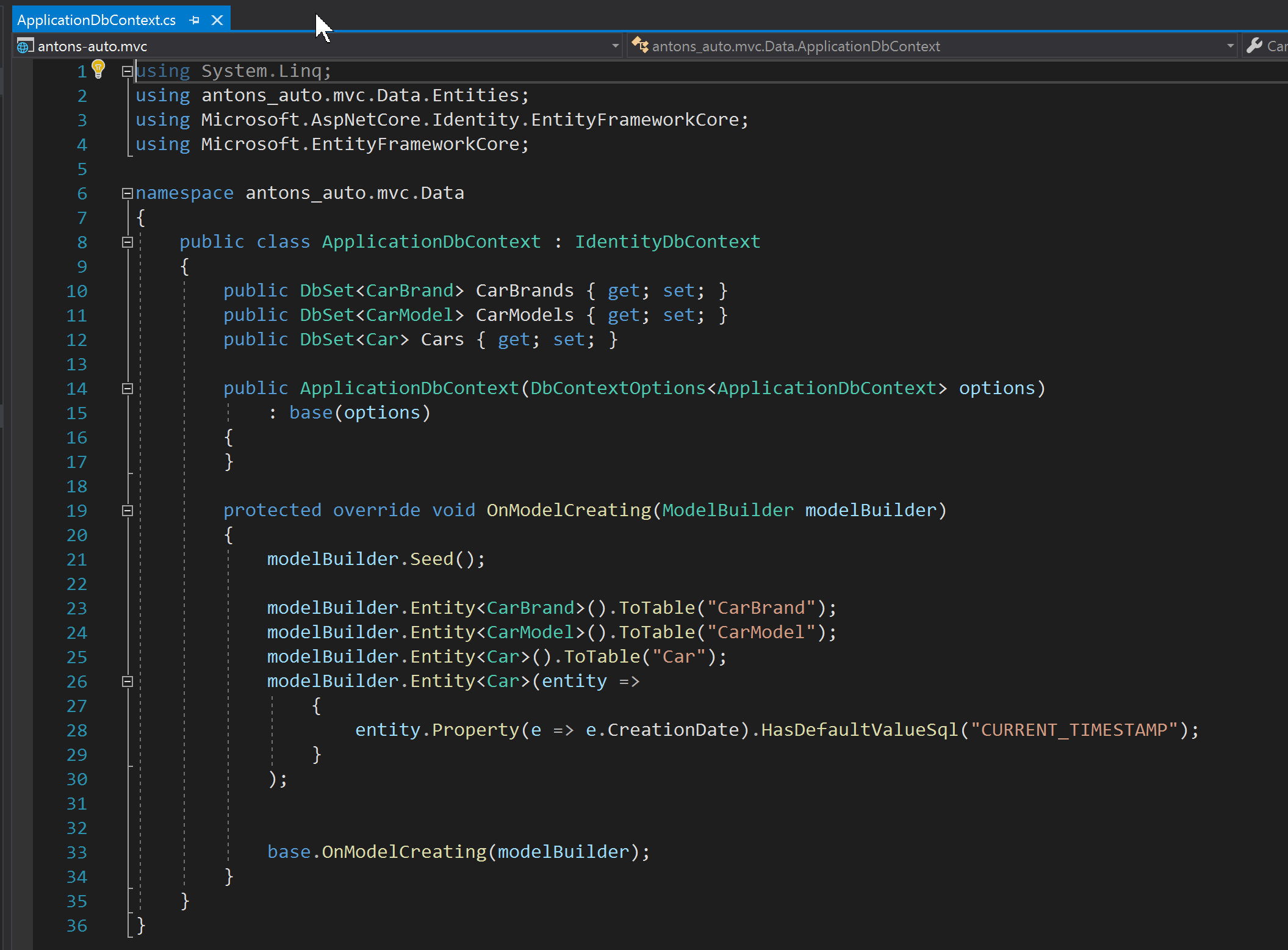
Under data/entities er klasser som definerer databasen. Databasen er oprettet i Azure som en SQL Server database.



* CarBrand indeholder bilmærker
* CarModel indeholder bilmodeller
* Car indeholder bil

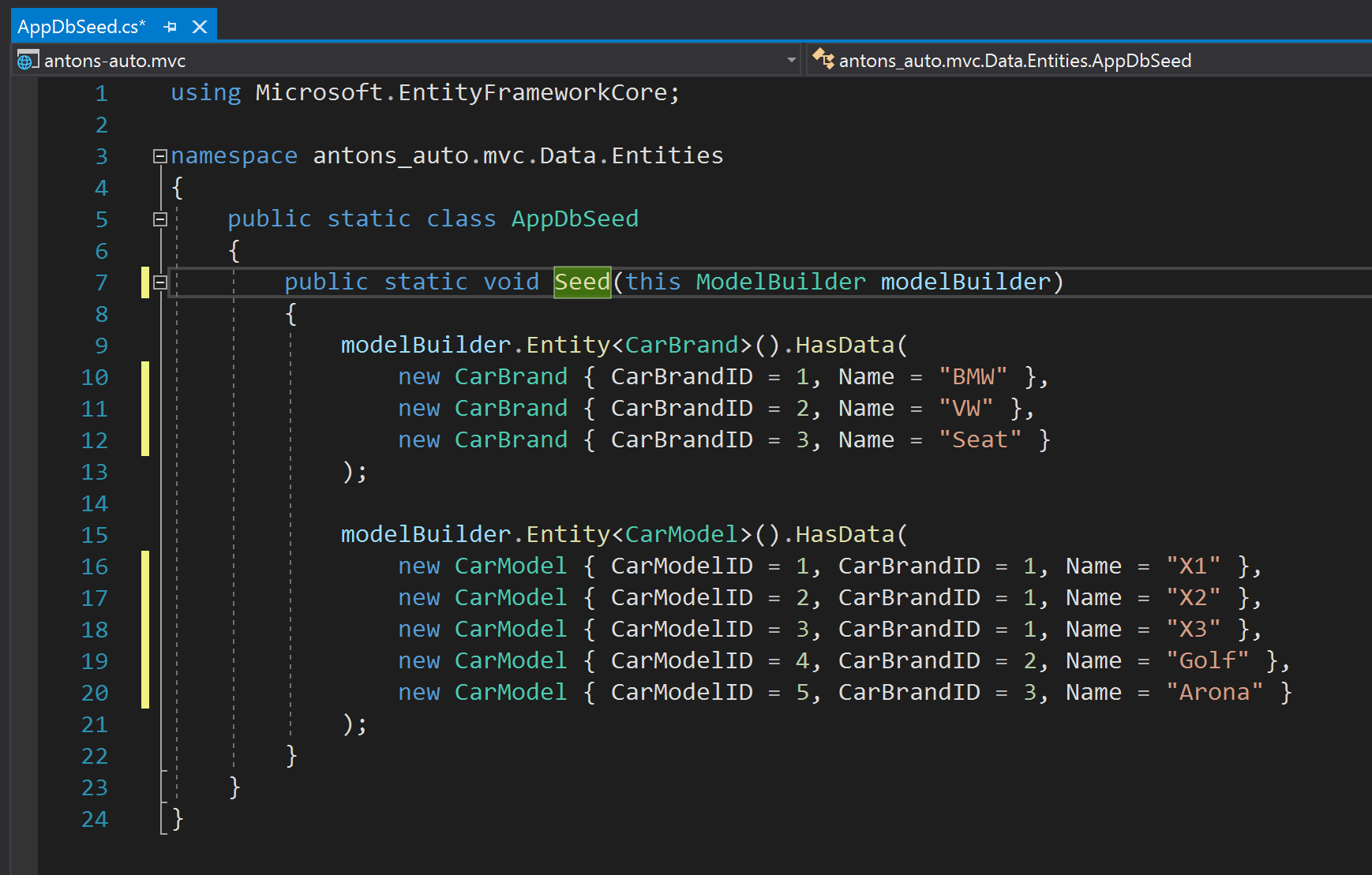


Jeg har på de enkelte felter defineret hvilke egenskaber de skal have, ved brug af attributter. På ovenstående vises Car klassen. Her er Year, Price og MileAge obligatoriske felter og dermed not null i databasen. Endvidere er MaxLength på ImageUrl sat til maksimum 500 karakterer.



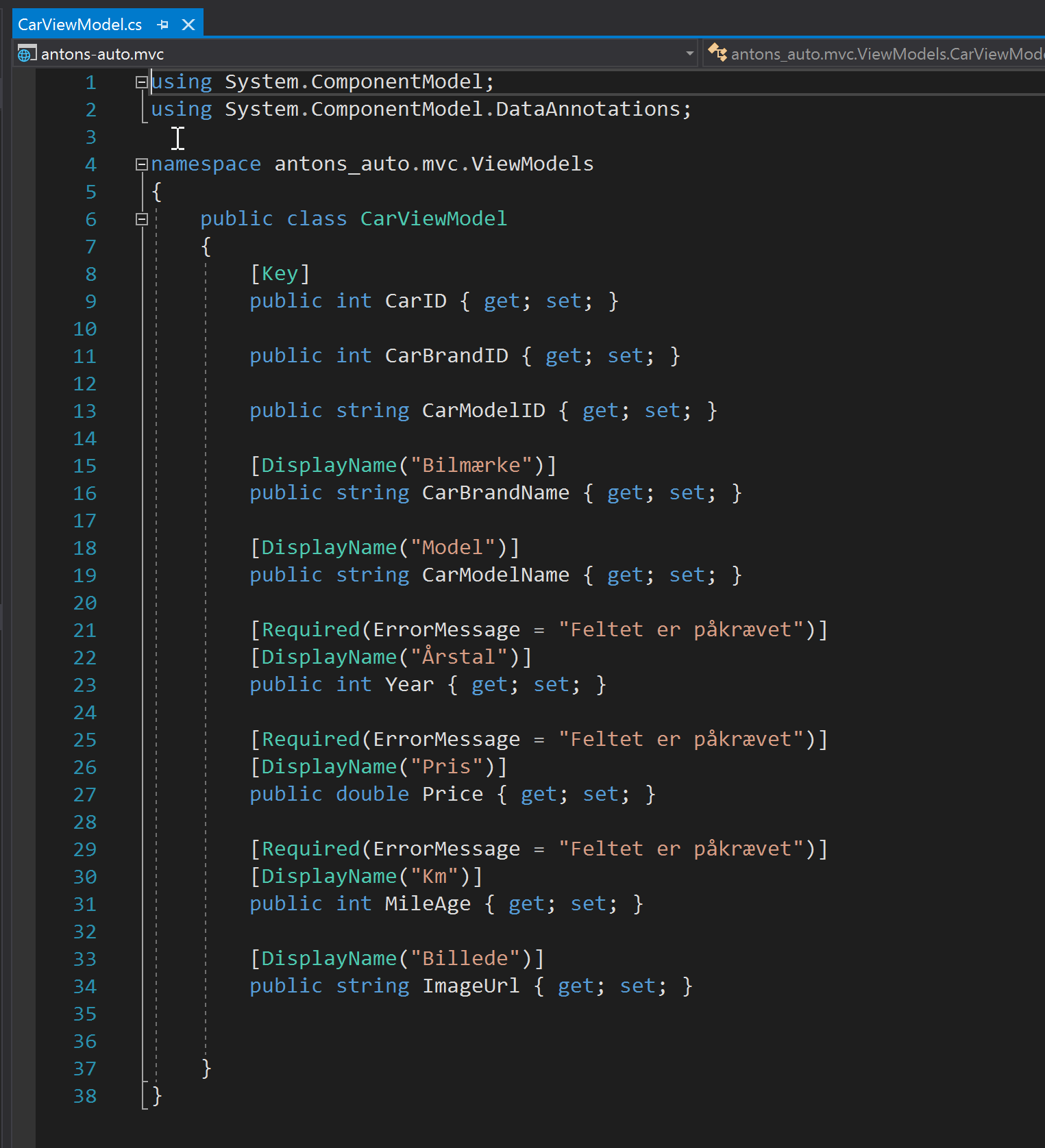
I ApplicationDbContext konfigureres EF. Her angives hvad tabel navnene skal være. Som udgangspunkt vil de være i flertal. På Car tabellen angives en default værdi på CreationDate. Den udfyldes med dato og tid når en ny bil oprettes.

I linie 21 kaldes en funktion som opretter nogle predefinerede bilmærker og bilmodeller.



## ViewModels

For at afkoble database klasserne fra View’s benytter jeg mig af et MVVM pattern. På denne måde kan jeg ændre i database konfigurationen, uden at det har indflydelse på mine views.



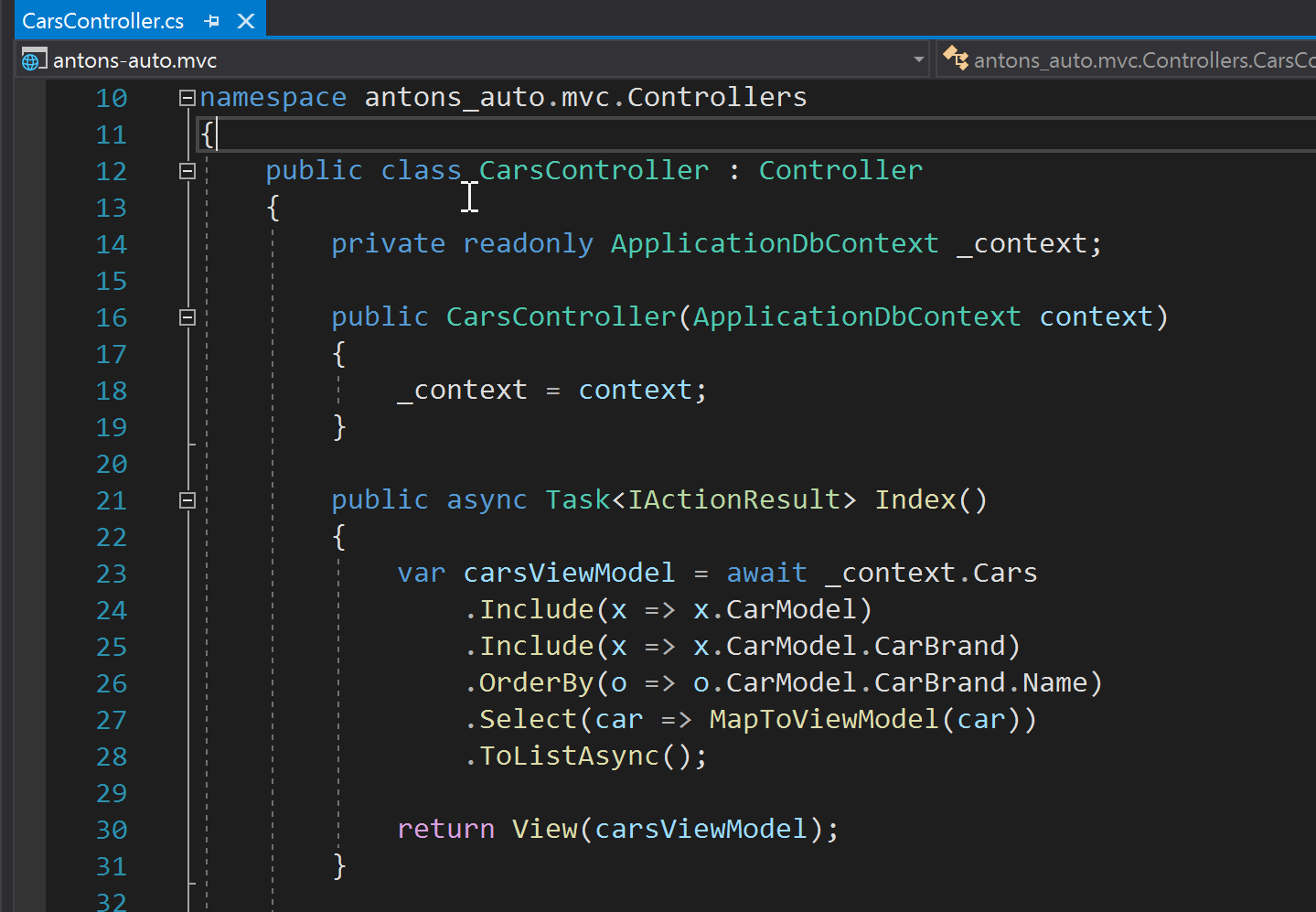
En anden fordel ved viewmodeller, er brugen af attributter til at konfigurere UI’en. Her er displayname konfigureret, så hvis det ændres slår det igennem på alle views.

Ved brug af required, får man klient validering out-of-the-box.

## Controllers

I Car controlleren har jeg taget udgangspunkt i den kode som VS genererer. Det er første gang jeg rigtig har prøvet det.

For at hente bilmærker og modeller, benytter jeg mig af Include. På denne måde er de hentet i samme forespørgsel.



## Anton’s Auto web

Følgende er skærmbilleder fra web løsningen

