

## INHOUDSTAFEL

Voorwoord .....	2
<b>1. Analyse &amp; ontwerp:</b>	
1.1 Domeinregels .....	3 - 4
1.2 Klassendiagram (UML) .....	5 - 6
1.3 Entity Relationship Diagram (ERD) .....	7
<b>2. Technologie &amp; opstart:</b>	
2.1 Configuratie .....	8 - 9
2.2 Project opstarten (laten runnen) .....	10
<b>3. Onafhankelijke lagen:</b>	
3.1 Business laag .....	11 - 17
3.2 Data laag .....	18 - 23
3.3 Presentatie laag .....	24 - 46
<b>4. Exception &amp; foutmeldingen:</b>	
4.1 Bespreking .....	47
<b>5. Correctheid -en Robuustheid:</b>	
5.1 Test strategieën .....	48 - 50
<b>6. Documentatie &amp; Bronvermeldingen:</b>	
6.1 Tutorialverwijzingen .....	51
6.2 Informatie .....	51

## Voorwoord

De opdracht werd in groep gemaakt door samenwerking Ali Maes, Ahmet Yimaz en Filip Rigoir. Onze opdracht bestaat uit een wagenparkbeheer from scratch opgebouwd tot een werkend systeem in WPF.

Elke donderdagvoormiddag was er een meeting met onze coach Tom om de stand van zaken te tonen en alles in de juiste context te zien. Daarnaast hebben we op regelmatige tijdstippen afgesproken via teams chat-en video om gedane zaken te bespreken en elkaar bij te sturen.



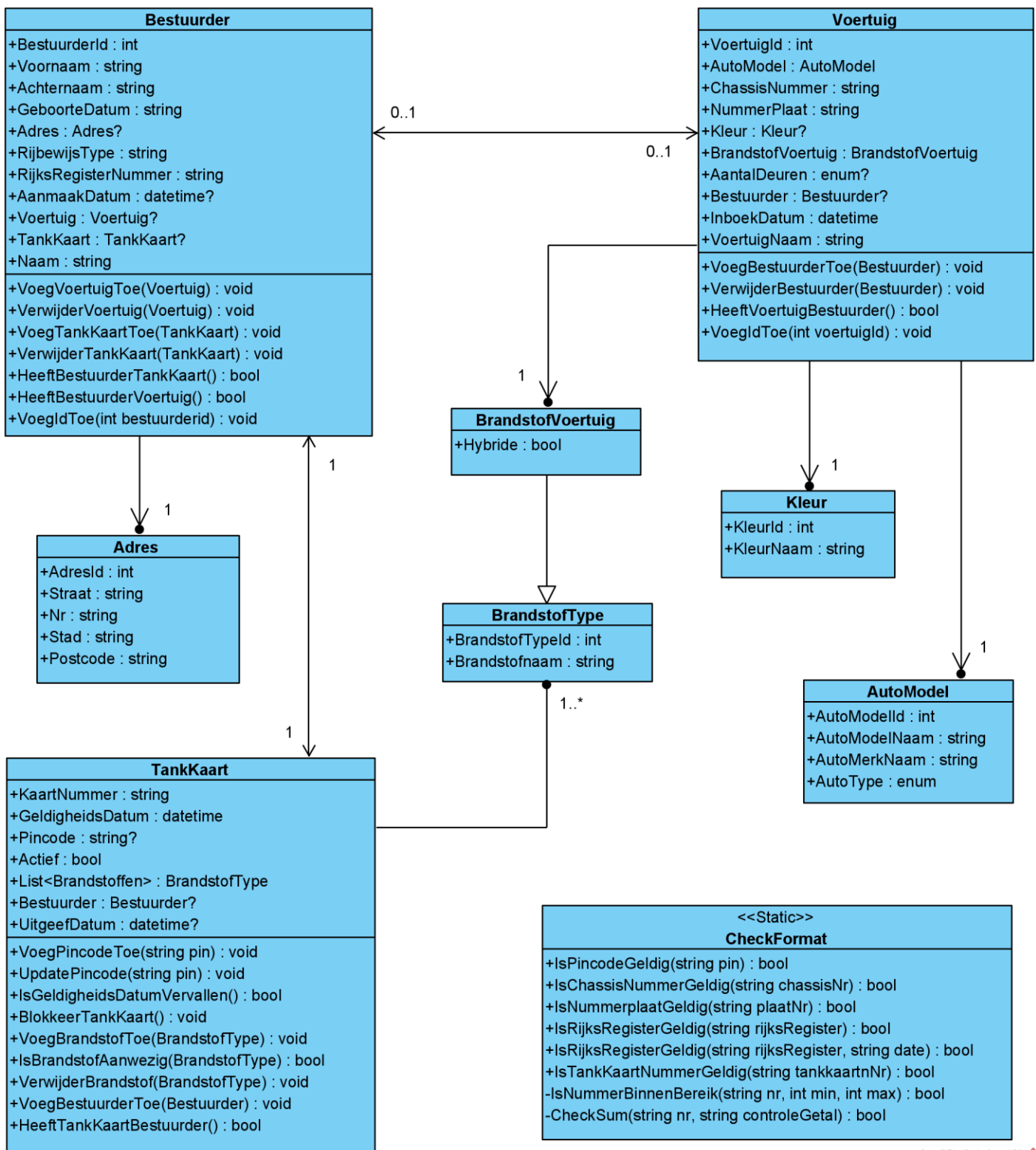
# 1. Analyse & ontwerp

## 1.1 Domeinregels

1. Een bestuurder bevat de volgende verplichte velden:
  - Voornaam, Achternaam
  - Geboortedatum
  - Rijksregisternummer
  - Type rijbewijs
2. Een bestuurder kan optioneel ook de volgende velden bevatten:
  - Straat, nummer, postcode en gemeente
  - Voertuig
  - Tankkaart
  - Bij het genereren kan een aanmaakdatum worden bewaard
3. Een tankkaart heeft de volgende verplichte velden:
  - Tankkaartnummer
  - Geldigheidsdatum
4. Een tankkaart kan optioneel de volgende velden invoegen:
  - Uitgeefdatum
  - Pincode
  - Bestuurder
5. Voertuig bevat de volgende verplichte velden:
  - Automodel : Merk, Naam en Autotype
  - Chassisnummer
  - Nummerplaat
  - Brandstoftype: Brandstofnaam, hybride
6. Voertuig kan optioneel volgende velden invoegen:
  - Kleur
  - Aantal deuren
  - Inboekdatum
  - Bestuurder
7. Kleur en brandstoftype is een bestaande lijst en kan mits uitbreiding beheerd worden om toe te voegen, te updaten en te verwijderen
8. Automodel is een lijst dat volledig beheerd kan worden via UI. Een lijst met automodellen worden getoond via het Window scherm
9. Autotype is een lijst via file dat at-runtime ingeladen wordt. Uitbreiding is mogelijk mits het stopzetten van de service. Dan aan te passen en terug op te starten. De volgende mogelijkheden zijn momenteel beschikbaar:
  - Cabriolet,
  - Coupé,
  - GT,
  - Terreinwagen,
  - Sedan,
  - Stationwagen,
  - SUV,
  - Sportwagen

10. Lijst aantal deuren is een hard gecodeerde lijst in de kern van het systeem. De kans voor uitbreiding is bijzonder klein. De volgende keuzes zijn mogelijk:
  - Twee,
  - Drie,
  - Vier,
  - Vijf
  - Zes
11. Tankkaartnummer wordt gegenereerd door een derde partij en kan worden ingegeven in het systeem via formulier. Een tankkaartnummer is uniek en kan maar eenmaal in het systeem voorkomen. Bij zowel het aanmaken als het updaten wordt de tankkaartnummer altijd gecontroleerd
12. Een tankkaart kan geblokkeerd worden maar niet uit het systeem verwijderd worden. Tankkaartnummer kan weliswaar gewijzigd worden indien een fout werd gemaakt
13. Rijksregisternummer wordt bij het aanmaken en het updaten gecontroleerd (samen met de geboortedatum)
14. Geboortedatum van bestuurder is altijd synchroon ten opzicht van het rijksregisternummer
15. Chassisnummer & nummerplaat wordt bij het aanmaken gecontroleerd zodat deze uniek zijn en blijven
16. Na het opstarten van de applicatie geeft de startpagina weer welk actie kan worden ondernomen:
  - Nieuw toevoegen
  - Zoeken -en/of updaten
  - Sluiten
17. Bij het aanklikken van button “Nieuw toevoegen” wordt een scherm geopend met 4 tabs:
  - Nieuw voertuig
  - Bestuurder aanmaken
  - Tankkaart ingeven
  - Automodel invoegen
18. Wanneer de keuze met button “Zoeken -en updaten” wordt geopend zijn dat eveneens 4 tabs:
  - Voertuigen
  - Bestuurders
  - Tankkaarten
  - Automodellen

## 1.2 Klassendiagram



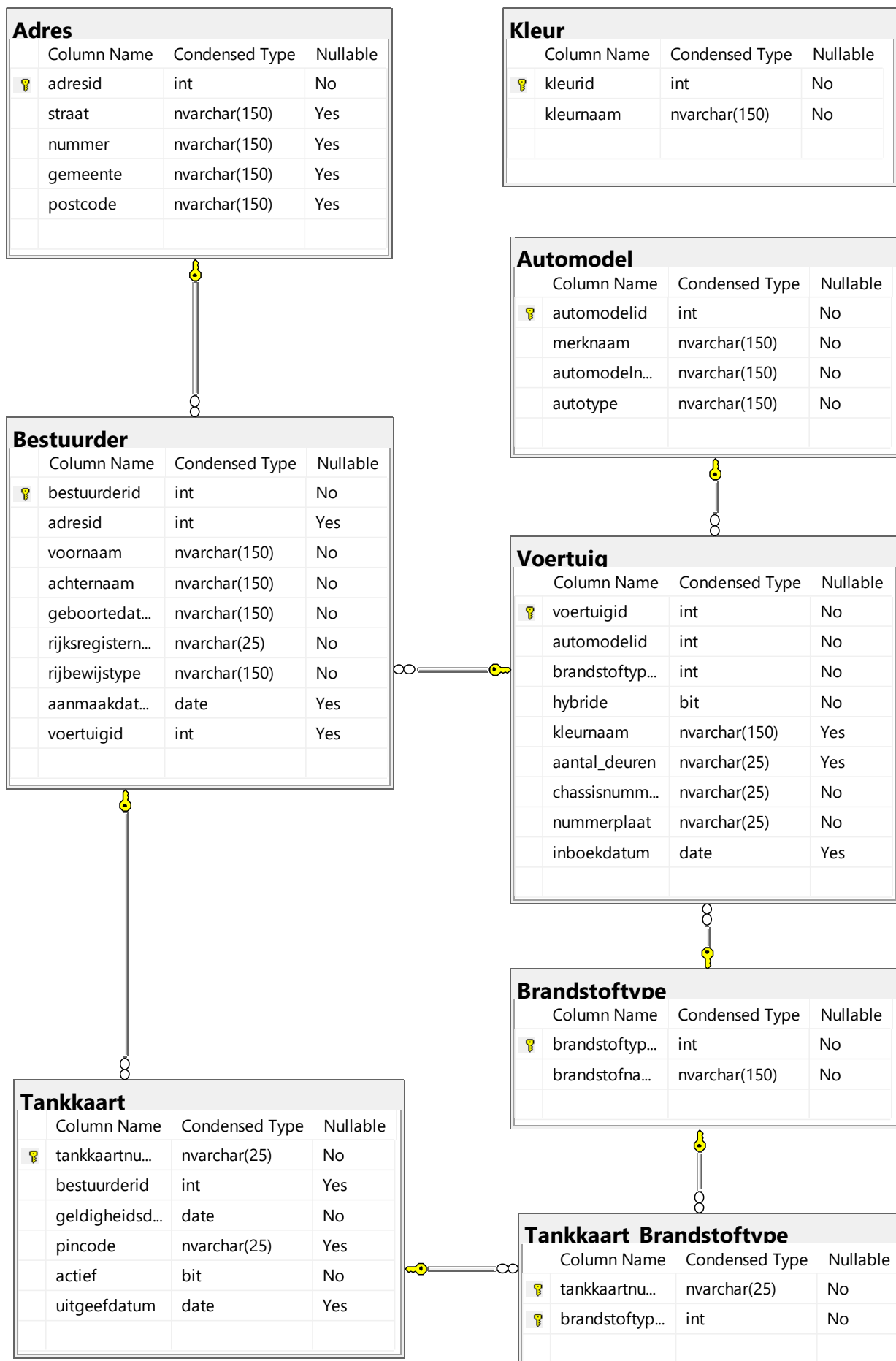
## UML aanvulling:

Aangezien het overzicht van klassendiagram niet overzichtelijk is met lange parameters in constructors, kan het hieronder geraadpleegd worden.

Entiteit	Constructor verplichte velden	Constructor overflow
Voertuig()	string voornaam, string achternaam, string geboorteDatum, string typeRijbewijs, string rijksRegisterNummer	int bestuurderId
AutoModel()	string merk, string autoModelNaam, AutoType autoType	int autoModelId
AutoType()	string autoTypeNaam	int autoTypeId
BrandstofType()	string brandstofNaam	int brandstofTypeId
BrandstofVoertuig()	string brandstofNaam, bool isHybride	int brandstofTypeId
Kleur()	string kleurNaam	int kleurId
AantalDeuren	Enum: id, string	
Bestuurder()	string voornaam, string achternaam, string geboorteDatum, string typeRijbewijs, string rijBewijsNummer, string rijksRegisterNummer	int bestuurderId
Adres()	string straat, string nr, string postcode, string gemeente	int adresId
TankKaart()	string kaartnummer, bool actief, DateTime geldigheidsDatum, string pincode = null	Bool Actief staat default op true

Propertes met een vraagteken in de UML tonen aan dat deze eventueel een NULL-waarde mag bevatten. Deze properties zijn optioneel in te vullen en dus niet verplicht.

## 1.3 Entity Relationship Diagram



## 2. Technologie & opstart

### 2.1 Configuratie

De applicatie is geprogrammeert in C# met verschillende onafhankelijke lagen. Er wordt voor elk nieuw project binnen de solution gebruik gemaakt van .NET Core 5.0. We gebruiken als debugger Microsoft Visual Studio.

De opstart van het project kent een minimale configuratie die we kort bespreken hoe het project tot stand is gekomen. Onderstaande acties werden systematisch uitgevoerd wanneer de app daar klaar voor was om de volgende stap in te gaan:

Projectnaam	Beschrijving
Opstart & configuratie Github	<p>Een bestaande account van Ali werd hiervoor gebruikt. Ahmet &amp; Filip kregen toegang voor de private repo.</p> <p>De volgende zaken werden in acht genomen om GIT succesvol te integreren in ons project via Visual Studio:</p> <ol style="list-style-type: none"> <li>1). Selecteer in Visual studio Team Explorer</li> <li>2). Selecteer Manage Connections in toolbar Icon</li> <li>3). Klik op connect in github section</li> <li>4). Selecteer de aangemaakte repo en klik op connecteer</li> <li>5). Fetch het project, maak een comment en push als test</li> <li>6). Clone project voor de andere</li> </ol>
FleetManagement	<p>Om de business laag op te starten is een <i>class Library .net core 5.0</i> gekozen en naam FleetManagement toegekend. Er werden geen Nuget packages geïnstalleerd.</p>
FleetManagement.TEST	<p>Om de business laag van een goede Unit test te voorzien werd een class Library .net core 5.0 gekozen als project.</p> <p>De volgende acties werden daarna ondernomen:</p> <ul style="list-style-type: none"> <li>• Via NuGet Package =&gt; xUnit Test Project geïnstalleerd.</li> <li>• Via Add =&gt; project reference =&gt; FleetManagement aangeduid</li> </ul>
FleetManagement.ADO	<p>Eveneens om een data laag te integreren werd er gekozen voor class Library .net 5.0.</p> <p>Daarna werd de volgende configuratie voltooid:</p> <ul style="list-style-type: none"> <li>• Via NuGet Package =&gt; Microsoft.Data.SqlClient geïnstalleerd</li> <li>• Via Add =&gt; Project reference =&gt; FleetManagement aangeduid</li> </ul>
ConsoleApplication	<p>Een eenvoudige console app .net core 5.0 werd vervolgens aangemaakt om een simpele tests te maken dat aantoonst, dat volgende zaken slagen waarvoor ze dienen:</p> <ul style="list-style-type: none"> <li>- Aanmaken,</li> <li>- Updaten</li> <li>- Verwijderen</li> </ul> <p>Volgende Entiteiten worden getest:</p> <ul style="list-style-type: none"> <li>- Bestuurder</li> <li>- Tankkaart</li> <li>- Voertuig</li> </ul>



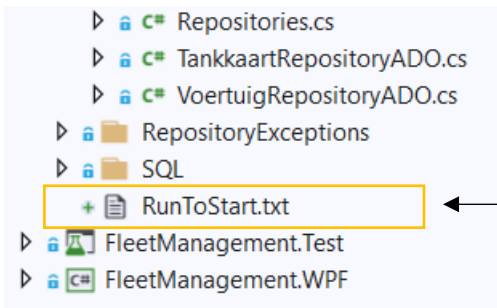
FleetManagemet.WPF	<p>Om een Visueel aspect te creëren is gekozen voor WPF. Deze installatie ging als volgt:</p> <ul style="list-style-type: none"> <li>- Start Project en kies voor .net core 5.0 en WPF Application</li> <li>- Via NuGet package =&gt; Microsoft.Extensions.Configuration &amp; Microsoft.Extensions.Configuration.Json geïnstalleerd</li> <li>- Via Add =&gt; Project reference =&gt; FleetManagement &amp; FleetManagent.ADO aangeduid</li> <li>- Via Add =&gt; New Item =&gt; app.json toegevoegd (zie verder om de configuratie te doorlopen)</li> </ul>
--------------------	---

## App.Json config-File

Benaming	Beschrijving
connectionStrings: FleetManagementConnectionString	De connectie string wordt ingeladen via dit pad met een daartoe voorziene property GetConnectionString
autotypes	Alle autotypes worden in deze file gezet om later eenmalig in te laden als WPF opstart. De property GetSection wordt hiervoor gebruikt en gecast naar IEnumerable.
Extra configuratie vereisten	Klik op rechtermuisklik van app.json en zet "Copy to Output Directory" op "Copy if Newer" om succesvol te kunnen inladen.

### 3.1 Project opstarten (en laten runnen)

Om een snelle start te nemen met ons project kan de volgende informatie zeer nuttig zijn:

Stappen	Actie
Repository GitHub:	<a href="https://github.com/alimaes/FleetManagement.git">https://github.com/alimaes/FleetManagement.git</a>
Open Visual Studio:	Kies “Clone a Repository” en plak de link bovenaan in de wizard. Volg de wizard en het project wordt gecloned.
Database	Maak een database onder .\SQLEXPRESS via MSSMS, genaamd; <b>fleetManagement</b>
Create Tables commands	In onze solution staat bij project FleetManagement.ADO een file RunToStart.txt. Deze loodst je door een reeks instellingen voor database. In deze file staat een script om alle tabellen in de juiste volgorde te creëren en minimale lijst aan te leggen die nodig zijn om als FK te dienen.
Insert commands	 <p>Kopieer het script en plak deze reeks query's in MSSMS.</p>
ConsoleApplicatie	Je kan bij voorkeur eerst de zopas aangemaakte database en tabellen testen. Kies dan ConsoleApplicatie als start up en klik op debug.
FleetManagement.WPF	Daarna of direct (wat je zelf verkiest) kan je WPF als startup kiezen en de applicatie met de UI runnen. Alle bestaande lijsten zijn nu aanwezig om alles perfect te laten verlopen.

### 3. Onafhankelijke lagen

#### 3.1 Business laag

Het businessmodel bestaat uit 10 classes met elk hun eigen doel. Elke class heeft een exception om problemen en berichten op te gooien zodat dit met een try & catch kan opgevangen worden. Om zaken te controleren is een aparte static class aangemaakt met daarin de nodige checks. We geven een globaal beknopte beschrijvingen met daarin de belangrijkste eigenschappen.

#### Class Bestuurder:

De controles in constructor	
Het rijksregisternummer wordt via een static class gecontroleerd op fouten. De geboortedatum en rijksregister is technisch met elkaar verbonden waardoor deze twee samen gecontroleerd worden.	
Alle velden worden gecontroleerd op nullable en lege string	
Via overflow kan ook de ID nummer worden meegegeven. Alle andere velden worden via this() naar de bovenliggende Constructor verwezen; zodat maar één keer moet gecheckt worden. Het ID nummer wordt in de huidige constructor gecontroleerd op zijn geldigheid.	
Property/Method	Beschrijving
GeboorteDatum	Datatype bij dit property is een string zodat dag en maand ook 00 kan zijn
HeeftBestuurderVoertuig	Een property met als doel te controleren dat de bestuurder een voertuig heeft. Voertuig is een containment class dat via een method geïnjecteerd wordt. Dit kan een object ofwel null zijn.
HeeftBestuurderTankKaart	Een property dat controleert dat een bestuurder in het bezit is van een tankkaart. Tankkaart is een containment class en kan object ofwel null zijn.
Naam	Een property dat twee properties met elkaar verbindt: Achternaam + Voornaam. Op deze wijze kan de volledige naam met één property worden voorgesteld.
VoegIdToe()	Wanneer bestuurder wordt aangemaakt bevat dit geen ID nummer. Dit wordt automatisch toegekend. Zo'n ID nummer kan meegegeven worden via constructor maar eveneens geïnjecteerd worden via method. Deze method controleert eerst het ID nummer.
VoegVoertuigToe()	Deze method bestaat uit twee belangrijke aspecten.  1). Om containment te plaatsen en zijn relatie tot stand te brengen 2). Om de relatie op te vangen en te stoppen wanneer de relatie is voltooid  Na het succesvol toevoegen kunnen beide elkaar valideren.
VoegTankKaartToe()	
VerwijderVoertuig()	Op dezelfde wijze kan een containment van elkaar losgemaakt worden. Nadat deze is verwijderd kunnen beide confirmeren dat ze niet meer samen horen.
VerwijderTankKaart()	
Equals()	Om dezelfde instantie te kunnen vergelijken wordt deze method overschreven met de nodige property instellingen.
GetHashCode()	Als Equals() wordt overschreven wordt deze automatisch ook geconfigureerd.

## Class Adres:

De controles in constructor	
Alle velden worden gecontroleerd op nullable maar lege string is toegestaan.	
Property/Method	Beschrijving
VoegIdToe()	Om geen nieuw instantie van een bestaande instantie te moeten aanmaken, wordt een method gebruikt dat de ID nummer kan plaatsen. Het wordt gecontroleerd op zijn geldigheid met extra voorwaarden dat Adres nog geen ID nummer heeft.
Override ToString()	Deze method wordt overschreven om een output te maken met properties en uitzicht die wij wensen.

## Class Kleur:

De controles in constructor
Deze class bestaat uit twee properties. ID nummer en kleurnaam. Kleurnaam wordt gecheckt want null en leeg is niet toegestaan.
ID nummer kan worden ingegeven via een overflow constructor en kleurnaam wordt via this verwezen naar de top van de hiërarchie zodat maar één check moet gedaan worden.
ID nummer wordt gecontroleerd zodat er geen ongeldige ID 's geplaatst worden.
Deze class heeft als doel een lijst te zijn en is onderdeel van class Voertuig.

## Class AantalDeuren:

Deze class is van het datatype Enum. Het is niet noodzakelijk om aantal deuren voortdurend aan te passen. Dit verandert bijna nooit, tenzij een uitbreiding van het project gevraagd wordt.
---

## Class AutoType:

Deze lijst is met opzet in de config-file gezet. Het leek ons een uitgelezen kans om zo'n integratie en uitdaging aan te gaan. Dit wordt éénmalig ingeladen bij de manager om als lijst beschikbaar te zijn.
--

## Class AutoModel:

Dit is een datatype dat fungeert als onderdeel van class Voertuig. In de constructor zijn alle velden verplicht en mogen zij niet null of leeg zijn.
Via een tweede constructor kan ook de ID meegegeven worden die eveneens aan de juiste voorwaarden voldoet.

## Class Voertuig:

### De controles in constructor

In de constructor worden cruciale checks uitgevoerd:

- Nullable of leeg
- Chassisnummer formaat moet juist zijn
- Nummerplaat moet aan alle eisen voldoen

In de overload method kan ook een ID worden meegegeven. Alle andere argumenten worden doorverwezen naar de bovenliggende constructor via this. Dit zorgt ervoor dat er maar eenmalig moet gecontroleerd worden.

Property/Method	Beschrijving
HeeftVoertuigBestuurder	Voertuig kan nagaan of een entiteit Bestuurder is geplaatst.
VoertuigNaam	Een property dat twee properties met elkaar verbindt: Merknaam + automodelnaam. Op deze wijze kan de volledige naam met één property worden voorgesteld.
VoegIdToe()	Kan een ID plaatsen die achteraf wordt toegekend. Voorwaarde is dat een ID nog niet is geplaatst.
VoegBestuurderToe()	Deze method bestaat uit twee belangrijke rollen.  1). Om containment te plaatsen en zijn relatie tot stand te brengen 2). Om de relatie op te vangen en te stoppen wanneer de relatie is voltooid  Na het succesvol toevoegen kunnen beide elkaar valideren.
VerwijderBestuurder()	Op dezelfde wijze kan een containment van elkaar losgemaakt worden. Nadat deze is verwijderd kunnen beide confirmeren dat ze niet meer samen horen.
UpdateNummerplaat()	Kan een nummerplaat overschrijven met de nodige check van het format.
Equals()	Om dezelfde instantie te kunnen vergelijken wordt deze method overschreven met de nodige property instellingen.
GetHashCode()	Als Equals() wordt overschreven wordt deze automatisch ook geconfigureerd.

## Class Tankkaart:

### De controles in constructor

In de constructor worden cruciale checks uitgevoerd:

- Nullable of lege string
- Tankkaartnummer
- Is geldigheidsdatum vervallen
- Wanneer een pincode niet leeg is wordt deze gecheckt

In database kan het voorvallen dat een tankkaart nog geldig is maar toch is de datum vervallen. Om ervoor te zorgen dat tankkaartobject geen anomalieën bevatten, wordt dit gecheckt bij elke nieuwe instantie.

Anders bestaat het risico dat de ene method de tankkaart ongeldig verklaard en property status op zijn beurt zegt: het is geldig. In ons model is dat onmogelijk; het wordt gecorrigeerd.

Onder geen enkel beding wordt de status automatisch van true naar false gezet. Geblokkeerd is geblokkeerd.

Property/Method	Beschrijving
HeeftTankKaartBestuurder	Deze property checkt of een tankkaart gelinkt is aan een Bestuurder.
IsGeldigheidsDatumVervallen	Bekijkt dat de datum niet is vervallen. Let op het feit dat we een geldigheidsdatum hebben en geen vervaldatum. Dit betekent concreet dat een datum is vervallen zodra de datum voorbij is. Dus op dezelfde dag is de tankkaart nog steeds geldig.
BlokkeerTankKaart()	Deze zet per direct de tankkaart status op ongeldig (false)
VoegPincodeToe()	Een pincode kan achteraf toegevoegd worden maar op voorwaarden dat er nog geen pincode is geplaatst.
UpdatePincode()	Kan een pincode wijzigen als er een pincode aanwezig is. Wanneer een pincode niet aanwezig is dan levert dat een exception op.
IsBrandstofAanwezig()	Controleert dat een brandstoftype aan de lijst is toegevoegd.
VoegBrandstofToe()	Voegt een brandstof toe nadat deze niet in de lijst staat.
VoegBestuurderToe()	Deze method bestaat uit twee belangrijke rollen.  1). Om containment te plaatsen en zijn relatie tot stand te brengen 2). Om de relatie op te vangen en te stoppen wanneer de relatie is voltooid  Na het succesvol toevoegen kunnen beide elkaar valideren.
VerwijderBestuurder()	Op dezelfde wijze kan een containment van elkaar losgemaakt worden. Nadat deze is verwijderd kunnen beide confirmeren dat ze niet meer samen horen.
UpdateTankkaartNummer()	Een nummer kan verkeerd worden ingegeven dus kan deze gewijzigd worden en de huidige overschrijven.
Equals()	Om dezelfde instantie te kunnen vergelijken wordt deze method overschreven
GetHashCode()	Als Equals() wordt overschreven wordt deze automatisch ook geconfigureerd.

## Class BrandstofType:

Deze class speelt een belangrijke rol voor twee andere classes; namelijk tankkaart en voertuig.

Deze class heeft enkel een ID en naam.

Alle velden zijn verplicht in te vullen die als parameters in de constructor worden gevraagd. De eerste is zonder ID en de tweede met ID. Alles wordt gecontroleerd op null en lege string.

## Class BrandstofVoertuig:

### De controles in constructor

Deze class is een overerving van BrandstofType met uitbreiding van een eigen property Hybride.

Aangezien de ID nummer in BrandstofType in de tweede constructor zit, is het technisch moeilijk dit mee te geven in de this en base. Hierdoor wordt de ID overschreven (override).

## Class Filter:

### De controles in constructor

Dit is een class Filter voor Voertuig, zodat op brandstofType, kleur en autotype kan gefilterd worden.

De map Filters met daarin class Filter staat in de rout van de business laag omdat volgende lagen daar gebruik van maken:

- de business laag,
- de data laag
- de presentatie

Deze filter filtert nog eens verder op merk en modelnaam.

## Static Class CheckFormat:

Deze class met bijhorende methods is volledig static. Dit zorgt ervoor dat een externe bron de validatie op zich kan nemen. Dit zorgt ervoor dat een eigen class zichzelf niet valideert. Alle checks maken gebruik van reguliere expressies

CheckFormat maakt ook gebruik van extra hulpclasses om code te onderscheiden en te onderhouden. De volgende classes leveren nuttige properties aan checkFormat die een string volledige ontlede:

1. **SplitGeboorteDatum:** Jaartal, ControleDatum
2. **SplitRijksregister:** Maand, Dag, ControleDatum, Geslacht, CheckGetal, ControleSom

Static method	Beschrijving
IsPincodeGeldig()	De pincode wordt gecontroleerd en moet bestaan uit 4 of 5 cijfers
IsChassisNummerGeldig()	De chassisnummer wordt gecheckt zoals aangegeven op wikipedia
IsNummerplaatGeldig()	Nummerplaat wordt gecontroleerd zoals is aangegeven op de website van de overheid. Namelijk het eerste gedeelte bestaat uit cijfer of letter. Een letter kan verwijzen naar een A-plaat voor garagist enz.
IsRijksRegisterGeldig()	Rijksregisternummer kan met en zonder geboortedatum gecheckt worden. De reglementen hieromtrent zijn opgevolgd van wikipedia en de overheid die nog extra regels opgeeft t.o.v. wikipedia.
IsTankKaartNummerGeldig()	Tankkaart wordt gecontroleerd dat het voldoet tussen de 16 en 20 digits. Bij research is gebleken dat niet alle tankstations hetzelfde formaat aanbieden. Om deze redenen is aantal digits flexibel ingesteld. Het kan eenvoudig veranderd worden.
IsNummerBinnenBereik()	Deze private method wordt meermaals aangesproken om een range tussen min en max te checken. Vb.: het getal moet tussen 0 tot en met 31 zijn.
CheckSum()	Private method dat de berekening op zich neemt om modulo uit te rekenen van het rijksregisternummer.



## Interfaces:

Interface	Beschrijving
IAutoModelRepository	De gevraagde implementatie van de business laag voor data te ontvangen, ongeacht wie deze opdracht uitvoert.
IBestuurderRepository	
IBrandstofRepository	
ITankkaartRepository	
IVoertuigKleurRepository	
IVoertuigRepository	
IRepositORIES	Extra gevraagde implementatie voor gezamenlijke interfaces

## Class Managers:

Method	Beschrijving
AutoModelManager	Beheert en controleert automodellen en filtert op autotype.
BestuurderManager	Beheert en controleert bestuurders en adressen.
BrandstofManager	Geeft de brandstoffenlijst.
KleurManager	Vraagt de kleurenlijst op
TankkaartManager	Beheert en controleert tankkaarten en brandstoffen voor tankkaarten
VoertuigManager	Beheert en controleert voertuigen en brandstof van voertuigen
Managers	Een class dat alle managers representeert. In de constructor wordt de IRepositORIES verwacht met de gezamenlijke implementaties

### 3.2 Data laag ADO.Net

Deze laag heeft als doel data op te slaan, te retrieven, te updaten en te verwijderen naargelang de gevraagde implementatie van businesslaag. Via dependency injection wordt de connectie string aangeleverd. Elke method heeft een try-catch-finally model integratie.

#### Abstract Class RepoConnection:

Fungeert als basistentiteit om een SqlConnection tot stand te brengen. Elke andere class dat communiceert met de database erf over van deze class.	
Via dit abstract datatype is de onderhoud en flexibiliteit verzekerd. Een uitbreiding zoals bijvoorbeeld “dispose” is gemakkelijk te integreren.	
Dit basistype is een abstract datatype zodat geen instantie kan gemaakt worden zonder over te erven. De variabelen binnen dit basistype is private en verwacht van de afgeleide datatypes een :base(connectionString).	
Property	Beschrijving
Connection	De modifier staat ingesteld op protected zodat in een overgeërfde omgeving de property kan worden aangesproken. De property kijkt eerst dat er een instantie is aangemaakt en alleen wanneer er geen instantie is, zal deze een instantie opleveren. De value kan via setter worden overschreven om een transactie tot stand te brengen.

#### Class : AutoModelRepositoryADO

Method	Beschrijving
BestaatAutoModelNaam()	Controleert dat het automodel nog niet bestaat. Aangezien elk automodel als FK kan optreden is het onnodig nog eens hetzelfde automodel toe te voegen. Het verandert enkel wanneer een van deze drie verschillend zijn: <i>merknaam</i> , <i>automodelnaam</i> en <i>autotypenaam</i> .
BestaatAutoModel()	Dit controleert het ID automodel zodat op een goede manier een updaten of verwijdering kan plaatsvinden.
FilterOpAutoModelNaam()	Deze zoekt in twee tabellen met een LIKE functie. Er is een concat voorzien van <i>merknaam + automodelnaam</i> . De resultaten zijn beperkt tot 50.
ZoekOpAutoType()	De standaard CRUD operaties die wordt gehanteerd.
UpdateAutoModel()	
VerwijderAutoModel()	
VoegAutoModelToe()	
IsAutoModelInGebruik()	Wordt aangesproken om te ondervragen of een automodel is gekoppeld aan een voertuig.

## Class VoertuigRepositoryADO:

Method	Beschrijving
BestaatVoertuig()	Een voertuig zoeken op ID nummer om te achterhalen of een meegeleverd voertuig object ook daadwerkelijk in database staat.
BestaatChassisnummer()	Checkt de chassisnummer om te achterhalen of deze uniek is.
BestaatNummerplaat()	Checkt de nummerplaat om te achterhalen of deze uniek is.
ZoekOpNummerplaatOfChassisNummer()	<p>De query kan tegelijkertijd zoeken in twee tabellen:</p> <ol style="list-style-type: none"> <li>1. Nummerplaat</li> <li>2. Chassisnummer</li> </ol> <p>Er is ofwel 0 of ofwel 1 resultaat mogelijk. Wanneer geen resultaten worden gevonden keert een NULL terug. In het andere geval een object van Voertuig met alle mogelijk containments.</p>
GeefAlleVoertuigenFilter()	<p>Vertegenwoordigt een lijst met alle voertuigen beperkt tot de eerste 50 resultaten. Er is een filternaam mogelijk met LIKE merk + modelnaam. Like werkt met concat om in twee tabellen tegelijkertijd te zoeken.</p> <p>De extra filter bevat 3 properties met een List met gevraagde items dat via IN(...) wordt geïntegreerd in de query. Dit gaat in drie stappen:</p> <ol style="list-style-type: none"> <li>1). De variabelen worden samengesteld met een vaste key + 1 omdat de waarde van buitenaf komt. Zo heeft de query nooit een dubbele variabele:</li> </ol> <pre> string queryKleur = ""; if(filter.Kleuren.Count &gt; 0) {     string kleur = "";     for (int i = 1; i &lt;= filter.Kleuren.Count; i++)     {         kleur += "@kleur" + i + ", "; // "@kleur1, @kleur2, etc.."     }      queryKleur = \$"AND v.kleurnaam IN({kleur[0..^2]}) "; } </pre> <ol style="list-style-type: none"> <li>2). Query wordt vervolgens dynamisch ingevuld. Als de lijst leeg is dan wordt het niet ingevoegd:</li> </ol> <pre> string query = "SELECT * FROM Voertuig v " + "JOIN AutoModel a ON v.automodelid = a.automodelid " + "JOIN Brandstofstypetype br ON v.brandstofstypetypeid = br.brandstofstypetypeid " + "LEFT JOIN Bestuurder b ON v.voertuigid = b.voertuigid " + "LEFT JOIN adres ad ON b.adresId = ad.adresId " + "WHERE concat(a.merknaam, ' ', a.automodelnaam) LIKE @autonaam + '% ' " + "{queryHybride}" + "{queryKleur}" + "{queryAutoType}" + "{queryBrandstof}" + "ORDER BY a.automodelnaam ASC, a.merknaam ASC " + "OFFSET 0 ROWS FETCH NEXT 50 ROWS ONLY"; </pre>

	<p>3). De parameters en values worden aan de hand van dezelfde tabel samengesteld:</p> <pre> if(filter.Kleuren.Count &gt; 0) {     int i = 1;     filter.Kleuren.ForEach(kleur =&gt;     {         _ = command.Parameters.AddWithValue("@kleur" + i, kleur);         i++;     }); } </pre> <p>Het resultaat is een zeer flexibele filter om te zoeken die wordt meegeleverd via een tweede argument.</p>
UpdateVoertuig()	De standaard operaties die worden gehanteerd.
VerwijderVoertuig()	
SelecteerZonderBestuurderFilter()	Presenteert een lijst van voertuigen die nog geen bestuurder hebben. Eveneens kan er gezocht worden op merk + automodel. Deze gegevens worden beperkt tot een 50-tal.

### Class KleurRepositoryADO:

Mits uitbreiding kan ook kleuren worden gemanipuleerd. Momenteel bestaat deze tabel enkel als lijstweergave.	
Method	Beschrijving
GeefAlleVoertuigKleuren()	Een lijst dat alle kleuren weergeeft om een Voertuig samen te stellen. De Kleur ID fungeert momenteel niet als FK. Zie class Voertuig hoe dat wordt aangegeven.

### Class BrandstofRepositoryADO:

Mits uitbreiding kan ook Brandstoffen worden toegevoegd en verwijderd worden. Momenteel bestaat deze tabel enkel als lijstweergave.	
Method	Beschrijving
GeeAlleBrandstoffen()	Een lijst dat alle brandstoffen kan geven die in de database staan zodat deze kan gebruikt worden als FK.

## Class TankkaartRepositoryADO:

Method	Beschrijving
VoegTankKaartToe()	<p>Voegt een tankkaart toe met transactie. Zodra een tankkaart is aangemaakt zal de brandstoffen in de koppeltabel ingevoegd worden indien van toepassing. Heeft de kaart geen brandstof wordt dit gedeelte niet uitgevoerd.</p> <p>Deze method heeft een overload met:</p> <ol style="list-style-type: none"> <li>1). Public Method</li> <li>2). Private Method als parameters connectie en parameters)</li> </ol>
VoegBrandstoffenToe()	Om de brandstoffen in te voegen afkomstig van de transactie. Deze method is private en kan niet van buitenaf geraadpleegd worden.
BestaatTankkaartBrandstof()	Om te achterhalen dat een type brandstof in de database staat.
VerwijderBrandstoffen()	Verwijdert alle brandstoffen van een tankkaart in één query
BestaatTankkaart()	Zoekt of een tanknummer uniek is.
GeefAlleTankkaarten()	Een lijst met de eerste 50 tankkaarten.
BrandstoffenVoorTankaart()	Alle brandstoffen worden opgehaald die gelinkt zijn aan de tankkaart.
ZoekTankKaart()	Zoekt op tankkaartnummer en geeft de Bestuurder mee indien deze tankkaart een bestuurder heeft. De brandstoffen worden niet mee ingeladen.
UpdateTankKaart()	<p>Deze method heeft een overload.</p> <ol style="list-style-type: none"> <li>1). Updaten met Object Tankkaart</li> <li>2). Updaten met object Tankkaart en ander tankkaartnummer</li> </ol>
TankaartenZonderBestuurder()	Levert een lijst van tankkaarten dat geen bestuurder heeft en op actief staat.
ZoekTankKaarten()	Geeft een lijstweergave van de eerste 50 tankkaarten dat actief ofwel inactief zijn. Een bool (geldig, niet geldig) wordt verwacht.

## Class BestuurderRepositoryADO:

Method	Beschrijving
BestaatBestuurder()	Een method dat op bestuurder ID checkt om te achterhalen of deze in database aanwezig is.
BestaatRijksRegisterNummer()	Check de rijksregisternummer in de volledige unieke tabel en kijkt of deze al dan niet in de database al voorkomt.
UpdateBestuurder()	Wordt met transactie uitgevoerd. Deze method heeft een paar overloads:  1). Public Method met als parameter Bestuurder 2). Public Method met als parameter Bestuurder en Nieuw rijksregisternummer 2). Private Method als parameters connectie en parameters
VoegBestuurderToe()	
VoegBestuurderAdresToe()	Wordt enkele gebruikt met een transactie en is ingesteld op private.
UpdateBestuurderAdres()	
HeeftBestuurderAdres()	Aangezien een bestuurder wordt aangemaakt en geüpdatet via transactie in Bestuurder dient er ook een mogelijkheid te zijn deze te checken.
VerwijderBestuurderAdres()	Adres wordt enkel gepost in database wanneer het object een adres containment heeft. Een leeg Adres tijdens aanmaken levert geen rij op in database.
VerwijderBestuurder()	Om een bestuurder te verwijderen uit de database
ZoekBestuurder()	Zoekt op Rijksregisternummer en geeft een object bestuurder mee indien het geen NULL is. Alle containments worden via JOIN & LEFT JOIN meegeleverd indien er gegevens aanwezig zijn.
SelecteerBestuurdersZonderVoertuig()	Een weergavelijst van bestuurders met de eerste 50 resultaten die geen voertuig hebben. Er kan gefilterd worden op achternaam + voornaam. Tabel achternaam en voornaam worden aan elkaar geconcat.
SelecteerBestuurdersZondertankkaart()	Een lijstweergave van bestuurder met de eerste 50 resultaten die nog geen tankkaart hebben. Eveneens heeft dit een filternaam als parameter zodat deze als argument kan worden meegegeven.
FilterOpBestuurdersNaam()	Filtert op achternaam + voornaam en geeft alle resultaten weer indien een lee string is gegeven. Weliswaar om de snelheid niet in gedrang te brengen worden deze resultaten ook beperkt voor de eerste 50 resultaten. Verder filteren kunnen de resultaten nog meer verfijnen.

## Class Repositories:

Deze implementatie is eveneens gedefinieerd in de interface van de business laag. De properties worden gezamenlijk gevraagd met eigen interface, zodat in één instantie alle repos staan.

Properties	Interface
BestuurderRepo()	IBestuurderRepository
VoertuigRepo()	IVoertuigRepository
TankkaartRepo()	ITankkaartRepository
AutoModelRepo()	IAutoModelRepository
KleurRepo()	IVoertuigKleurRepository
BrandstofRepo()	IBrandstofRepository

### 3.3 Presentatie laag WPF

Om het project een visueel gezicht te geven is WPF gebruikt. We geven kort en bondig uitleg over elk visueel aspect met daarbij de belangrijkste kensmerken en werking ervan.

We beginnen alvast met een kort schets van de configuratie opstartscherm. Wat gebeurt er achter de schermen vooral het visuele gedeelte tot stand wordt gebracht:

```
public partial class App : Application
{
    2 references | Filip Rigoir, 3 days ago | 1 author, 2 changes
    public Managers Managers { get; private set; }

    1 reference | Filip Rigoir, 14 days ago | 1 author, 9 changes
    public App()
    {
        //ConnectionString inladen
        IConfiguration config = new ConfigurationBuilder().AddJsonFile("app.json").Build();
        string connectionString = config.GetConnectionString("FleetManagementConnectionString");

        //ADO instantie aanmaken
        Interfaces.IRepositories repositories = new ADO.Repositories.Repositories(connectionString);

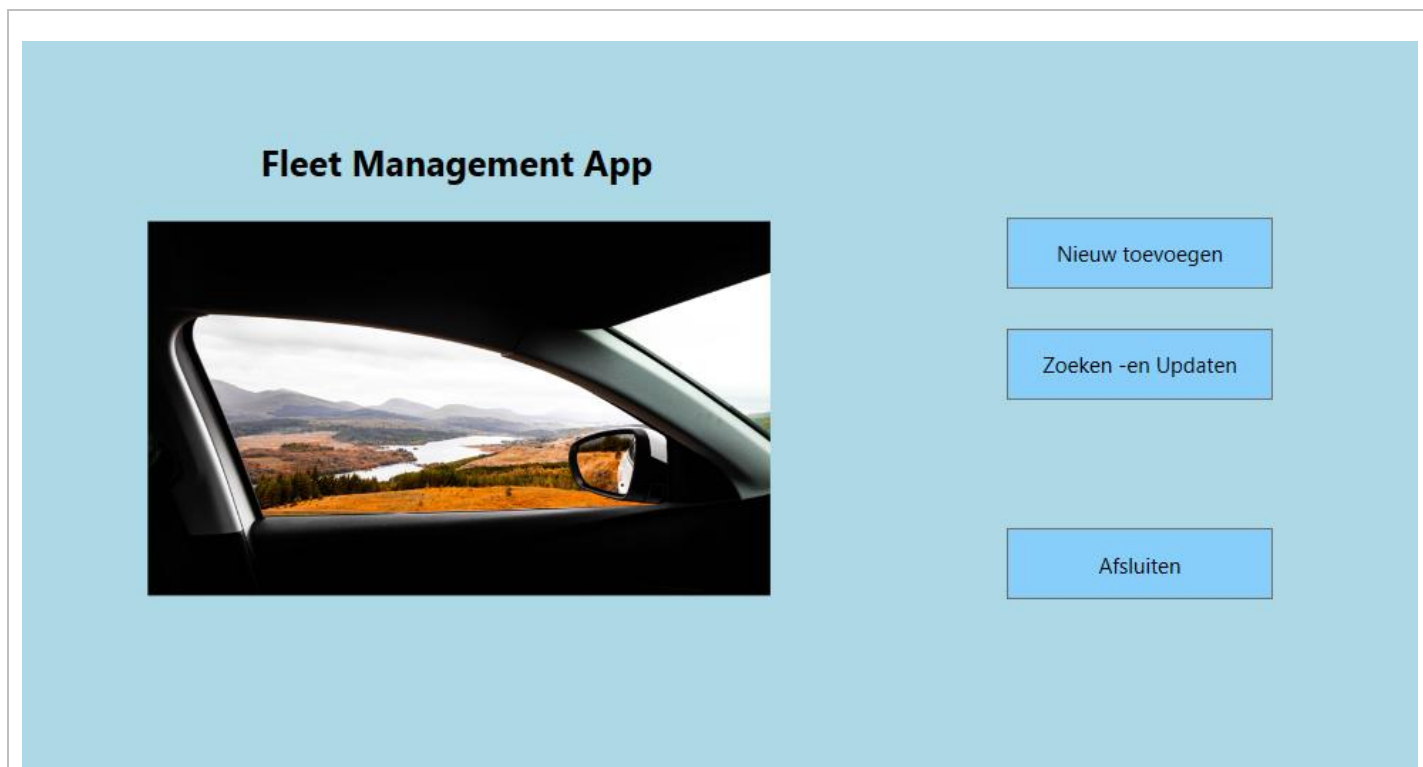
        //Manager instantie aamaken & Autotypes inladen
        Managers = new Managers(repositories)
        {
            AutoTypes = config.GetSection("autotypes").AsEnumerable().Where(a => !string.IsNullOrEmpty(a.Value))
        };

        0 references | Filip Rigoir, 22 days ago | 1 author, 2 changes
        protected override void OnStartup(StartupEventArgs e)
        {
            MainWindow = new MainWindow(Managers) { Title = "FleetManagement App" };
            MainWindow.Show();
        }
    }
}
```

Stappen	Hoe werkt het juist
Connection string	<p>De configurationbuilder gaat de config file inladen die in het projet WPF staat onder de naam app.json. Alle gegevens inclusief Autotypes worden ingeladen.</p> <p>Deze stap gaat op een veilige manier de connectionString ophalen en bij zich houden.</p>
ADO instantie	<p>Daarna wordt een instantie gemaakt van de implementatie van repositories. Een interface dat voorschrijft alle afzonderlijke interfaces aan te bieden in één class. De connectie string word meegegeven als argument.</p>
Managers	<p>Wat voor ADO geldt, geldt ook voor Managers. Een gezamenlijke class dat alle afzonderlijke managers in zich heeft. Elke Window kan zijn eigen Manager injecteren.</p> <p>Managers verwacht als argument de interface van de repositories.</p>
AutoTypes	<p>Na instantie van de managers wordt de autotypes uit de json lijst gehaald via property GetSection. Van de dictionary word een IEnumerable gemaakt.</p>
OnStartrUp()	<p>Tot slot werd in de App.xaml de opstart uitgezet om deze handmatig in code behind aan te geven. Dit laat ons toe om de MainWindow op te starten met dependency injection van de managers.</p>
Kleur, brandstoffen en Aantal deuren	<p>Lijst van aantal deuren, brandstoffen en kleuren worden éénmalig ingeladen in de constructor van de manager.</p>



De User Interfaces per scherm en uitleg:



Button	Woordje uitleg:
Nieuw toevoegen	<p>Deze Window geeft een venster weer met 4 tabs:</p> <ul style="list-style-type: none"> <li>- Nieuw voertuig</li> <li>- Bestuurder aanmaken</li> <li>- Tankkaart Ingeven</li> <li>- Automodel invoegen</li> </ul>
Zoeken -en Updaten	<p>Deze Window dient om bestaande gegevens te zoeken en te updaten. Bestaande data kan eerst worden opgezocht. Deze heeft eveneens dezelfde 4 tabs:</p> <ul style="list-style-type: none"> <li>- Voertuigen</li> <li>- Bestuurders</li> <li>- Tankkaarten</li> <li>- Automodellen</li> </ul>
Afsluiten	Om de applicatie op een eenvoudige wijze te beëindigen en terug te keren naar de code.

Hieronder vind je de schermen die hierboven worden vernoemd.

Nieuw voertuig
Bestuurder aanmaken
Tankkaart ingeven
Automodel invoegen

## Nieuw voertuig aanmaken

**Automodel: \***
Automodel kiezen

**Chassisnummer: \***

**Nummerplaat: \***

**Hybride: \***
☒ Neen ☐ Ja

**Brandstof: \***
Selecteer

**Kleur:**
Selecteer

**Aantal deuren:**
Selecteer

**Bestuurder:**
Bestuurder selecteren

Voertuig aanmaken
Reset
Sluiten

Subject	Woordje uitleg:
Automodel	Dit leidt naar een Window scherm om een bestaande automodel de kiezen. Wanneer dit wordt ingevoegd komt het automerk & automodel naast de button terecht.
Bestuurder	<p>Eveneens geldt hetzelfde voor button bestuurder. Er kan een bestaande bestuurder gekozen worden uit de lijst. Deze Window representeert enkel een lijst met bestuurder die nog geen voertuig hebben.</p> <p>Dit scherm wordt ook gebruikt in Tankkaart ingeven. Meer uitleg bij tankkaart.</p>
Reset	Button Reset wist alle gegevens zodat opnieuw kan begonnen worden met een leeg formulier
Sluiten	Window scherm wordt gesloten en keert terug naar het vorige.
De extra Window schermen kom hieronder nog aan bod.	

Nieuw voertuig
Bestuurder aanmaken
Tankkaart ingeven
Automodel invoegen

### Bestuurder aanmaken

**Voornaam: \***

**Achternaam: \***

**Geboortedatum: \*** Dag:  Maand:  Jaar:

**Rijksregisternummer: \***

**Adres:**

**Rijbewijs: \***

**Tankkaart:**

**Voertuig:**

Button	Woordje uitleg:
Adres	<div> <div>Nieuw adres ingeven</div> <div> <b>Straat:</b> <input type="text"/> </div> <div> <b>Nummer:</b> <input type="text"/> </div> <div> <b>Postcode:</b> <input type="text"/> </div> <div> <b>Gemeente:</b> <input type="text"/> </div> <div> <input type="button" value="Aanmaken"/> <input type="button" value="Annuleer"/> </div> </div> <p>Adres kan ingegeven worden via een extra Window scherm. Zo wordt de pagina niet te druk. Dit kan eerder belangrijke rol spelen bij uitbreiding van het project. Button Annuleer maakt alles weer blank. Een adres is optioneel en dus niet verplicht.</p>
Geboortedatum:	<p>Dit is een basic view. Dit is een ToDo om te optimaliseren. Momenteel kan een vrije datum worden ingegeven inclusief met de 00-00-0001 als datum. Ook het bisnummer kan dus worden ingegeven. Dat is het format zoals in de database. In principe kan alles gescheiden worden zodat het bisnummer en nul waarde in geboortedatum in code-behind plaatsvinden.</p>

Tankkaart en Voertuig selecteren	<div> <div> <b>Tankkaart:</b> <div>Annuleer tankkaart</div> 5695565965654562006 </div> <div> <b>Voertuig:</b> <div>Annuleer voertuig</div> Toyota Celica T </div> </div> <p>Via deze buttons kan er optioneel een Voertuig of tankkaart ingevoegd worden. Wanneer een object is gekozen wordt de tekst aan de rechterkant getoond. Ook de button verandert mee. Met een eenvoudige klik verwijder je de ingevoegde bestuurder terug uit het formulier.</p>
Info tekst	<p>Links onderaan, naast de buttons kan een melding in het rood of groen tevoorschijn komen. Dit naargelang de actie. Een bericht dat meer vertelt welke actie moet worden ondernomen. Dit zijn meestal de exceptions van de businesslaag die opgevangen worden.</p>

Nieuw voertuig
Bestuurder aanmaken
Tankkaart ingeven
Automodel invoegen

## Tankkaart ingeven

**Uitgeefdatum:**

**Tankkaartnummer: \***

**Geldigheidsdatum: \***

**Pincode:**

**Brandstoffen:**

**Bestuurder:**

**Brandstoffen:**  
Geen brandstoffen

Button/veld	Woordje uitleg:
Bestuurder selecteren	Dit gaat op dezelfde wijze, een extra Window scherm opent en er kan een bestuurder worden gekozen dat nog geen tankkaart heeft.
Brandstof toevoegen	Je kiest een brandstof in meerkeuzemenu en klik dan op "Brandstof toevoegen". Het gekozen brandstof type komt rechts bovenaan te staan terwijl de gekozen brandstof verdwijnt uit het meerkeuzemenu. Op deze wijze kan geen tweede worden ingevoegd. Maar mocht de UI dat niet checken is de manager die een melding zal geven.
Brandstoflijst legen	<div> <div> <b>Brandstoffen:</b>  Benzine, Elektrisch </div> <div> <input type="button" value="Brandstoflijst legen"/> </div> </div> <p>Nadat er één is toegevoegd krijg je altijd de keuze de lijst te wissen. In dit geval wordt het meerkeuzemenu terug gezet met alle keuzes.</p>
Geldigheidsdatum en uitgeefdatum	Wanneer deze alle twee zijn ingegeven wordt een check gedaan dat alles klopt. Een geldigheidsdatum ingeven die verstreken is krijgt ook de melding vervallen te zijn.

Fleet Management Toevoegbeheer

Nieuw voertuig

Bestuurder aanmaken

Tankkaart ingeven

Automodel invoegen

Nieuw automodel aanmaken

**Merknaam: \***

**Automodel: \***

**Autotype: \***

Selecteer ▾

Automodel aanmaken

Reset

Sluiten

Button/veld	Woordje uitleg:
Automdol aanmaken	Dit is de aanvulling van automodellenlijst dat gebruikt wordt om een voertuig samen te stellen. Elke automodel kan één of meerdere keren gekozen worden. Een uniek automodel wordt gezien als één van de drie velden verschillend zijn.
AutoType	Dit is dé lijstweergave dat werd ingeladen vanaf de config file tijdens het opstarten.

Voertuig	Autotype	Brandstof	Hybride	Kleur	Nummerplaat	Chassisnummer	
BMW iX3	Sedan	Benzine	Neen		2-EFG-364	55632015469875546	
Toyota Celica T	Sportwagen	Benzine	Neen	Groen	1-ZFV-123	DCNBSZEDG8ZB45239	
Lexus NX	Sedan	Benzine	Neen		Z-FZE-256	DMNSDRAT8VFSE2039	

Voertuig:

Merk + Automodel

Kies

Annuleer

**Werking****Woordje uitleg:**

Window voertuigen

Een lijst in een aparte Window scherm om optioneel een voertuig te selecteren voor bestuurder. Er komen enkel voertuigen tevoorschijn die nog geen bestuurder hebben.

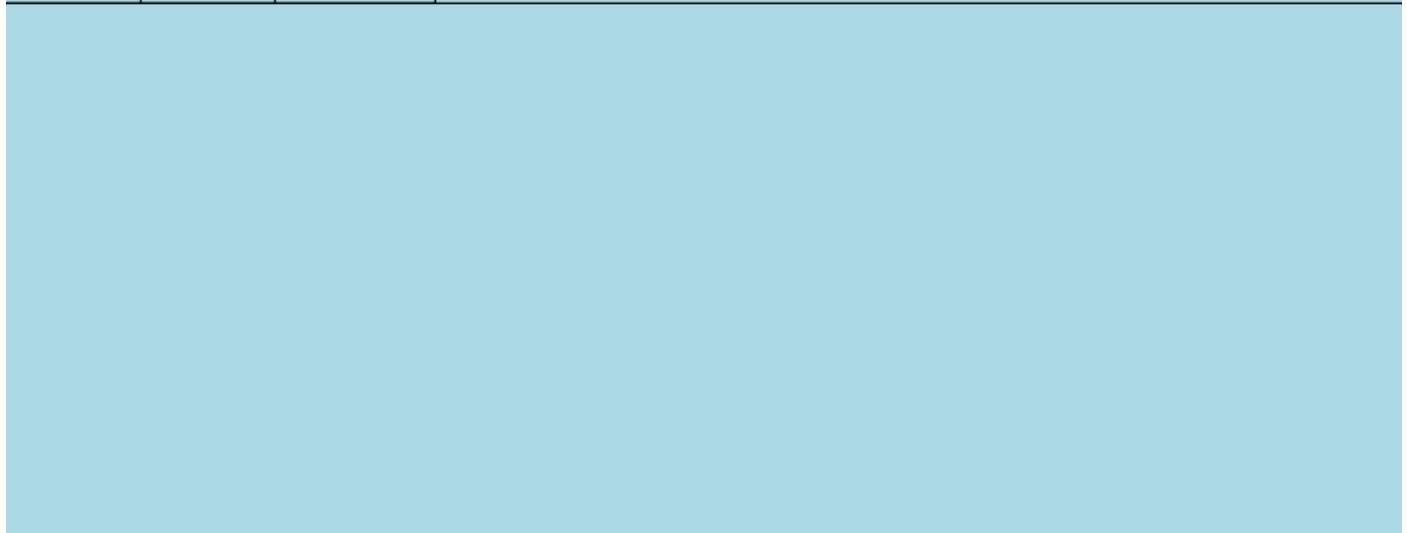
IValueConverter

Hybride & Nummerplaat hebben een IValueConverter om de gegevens per rij weer te geven zoals de klant het wil. Zo is Hybride oorspronkelijk true & false. Maar via deze implementatie is daar niets van te merken.

Filter

Via filter kan gezocht worden op voertuig. Eerst wordt het merk verwacht gevolgd door automodel.

Merk	Automodel	Autotype	
BMW	i4	Coupé	
BMW	iX	Coupé	
BMW	iX3	Sedan	
Land Rover	Defender	Terreinwagen	
Lexus	NX	Sedan	
Toyota	Celica T	Sportwagen	
Toyota	Supra S	Sportwagen	



Autonaam: Merk + Automodel

Kies

Annuleer

**Werking****Woordje uitleg:**

Window Automodellen

Een extra Window met allemaal automodellen om te selecteren voor een voertuig samen te stellen. Alle modellen worden getoond tot 50 resultaten. Daarna is filteren aangeraden. Dit om de snelheid niet in gedrang te brengen. Maar dat geldt voor alle resultaten van objecten.

Filter

Via filter kan gezocht worden op voertuig. Eerst wordt ook het merk verwacht daarna gevolgd door automodel.



Naam	Geboortedatum	Rijbewijs	Rijksregisternummer	Postcode	
Brink Jos	1999-03-01	B	99.03.01-653.59		
Dambryune Anna	1944-02-03	C, D	44.02.03-263.03		
De Man Jef	1972-11-28	B	72.11.28-268.93	9040	
De Phoo Winnie	1922-00-00	A, B	22.00.00-268.97		
Derdean Jean-Pol	1944-11-11	B7	44.11.11-111.75		
Huisman Josje	2002-12-01	D	02.12.01-236.83		
Leeuwentemmer Tanguy	1987-05-22	B	87.05.22-667.50		
Rinoir Jos	1976-00-00	A, C, B+	76.00.00-019.25		
Schoenmaker Deen	1962-11-02	C	62.11.02-536.07		
Talpe Louis	2001-12-01	B, C	01.12.01-563.26		

Naam:

Kies

Annuleer

Werking	Woordje uitleg:
Window Bestuurders	Extra Window scherm om een bestuurder te selecteren voor Tankkaart en Voertuig. Via dependency injection wordt deze info meegegeven of het voor tankkaart of voor voertuig is.
Voor Tankkaart	Alleen bestuurders komen tevoorschijn die geen tankkaart hebben.
Voor Voertuig	Alleen bestuurders komen tevoorschijn die geen voertuig hebben.
Filter	Via filter kan gezocht worden op volledige naam. Eerst wordt de achternaam verwacht daarna de voornaam.
Kies/Dubbelklik	<p>Wanneer in de lijst een bestuurder wordt geselecteerd wordt deze opgeslagen. Invoegen kan op twee manier:</p> <ol style="list-style-type: none"> <li>1). Via dubbelklik op de bestuurdersrij</li> <li>2). Selecteer een bestuurder en klik op Kies.</li> </ol> <p>Dit geldt voor alle selecteerschermen. Alle filters zoeken ook rechtstreeks in de database . Er is geen button nodig.</p>

Tankkaartnummer	Geldigheidsdatum	Pincode	Actief	Uitgeefdatum	
1111111111111111121	29/03/2023	9999	Ja		
1111111111111111122	06/02/2023		Ja		
5230525214589785620	16/01/2022		Ja		
5236599887564556355	05/02/2022		Ja		
5321458975632589725	25/12/2021		Ja	09/12/2021	
5411202312089765415	29/01/2022	2006	Ja		
5642525630154795666	17/03/2022		Ja		
5695565965654562006	19/02/2022	9637	Ja	15/12/2021	
8563202145698765298	22/01/2022		Ja	09/12/2021	
9653210478956402864	07/01/2022		Ja		
9878542130589754201	17/03/2022	1523	Ja	01/12/2021	

Tankkaart:





Button	Woordje uitleg:
Window Tankkaarten	Een bestuurder kan optioneel een tankkaart kiezen. Alle tankkaarten komen tevoorschijn die geldig zijn en nog geen bestuurder hebben. Een lambda zorgt er nog eens extra voor dat enkel tankkaarten tevoorschijn komen die actief zijn. Dit naargelang de geldigheidsdatum.
Zoekoptie	Er kan gezocht worden op tankkaartnummer. Het nummer ingeven en op zoek klikken.

Voertuigen	Bestuurders	Tankkaarten	Automodellen				
Automodel	Autotype	Brandstof	Hybride	Kleur	Nummerplaat	Chassisnummer	Bestuurder
Toyota Celica T	Sportwagen	Benzine	Neen	Groen	1-ZFV-123	DCNBSZEDG8ZB45239	
Toyota Celica T	Sportwagen	Diesel	Neen		A-BNJ-123	KDZSARCD4569VBN56	
Land Rover Defender	Terreinwagen	Benzine	Ja	Grijs	1-KJH-236	5356978452FGPM4Z5	
Land Rover Defender	Terreinwagen	Diesel	Neen		Z-FGB-136	DNBSFAZYKM4125695	Rigoir Ingrid
BMW i4	Coupé	Benzine	Neen	Blauw	1-GKL-169	GHJKPMAS8JKLPM965	Van Den Heuvel Loes
BMW iX	Coupé	Elektrisch	Neen	Beige	Z-ABC-123	58632SCVZA95DLM67	Rigoir Filip
Lexus NX	Sedan	Benzine	Neen		Z-FZE-256	DMNSDRAT8VFSE2039	Frimout Dirk
Lexus NX	Sedan	Diesel	Ja	Blauw	Z-ZER-340	DSREKGJHF5KDH6589	

Voertuig:

Chassisnummer of Nummerplaat:

Werking	Woordje uitleg:
Tabs	Telkens een andere tab wordt geselecteerd zal dit een refresh opleveren met de recentste gegevens. Alle entiteiten gerelateerd aan voertuig zitten in het object.
Naar detailweergave	Scherf zoeken heeft meerdere aanleidingen. Eén daarvan is een voertuig zoeken met intentie alle details weer te geven.
Update	Na het weergeven van detail kan via knop wijzigen de gegevens worden gewijzigd.
Filter	Eveneens als op voertuig wordt gefilterd, is dat eerste merk en daarna automodel.
Zoek op chassisnummer	Zoeken op deze twee gebeurt in hetzelfde veld. Het is slim genoeg om te herkennen welke nummer gevraagd wordt.
Zoek op nummerplaat	Wanneer een update plaatsvindt (zoals verwijderingen of aanpassing) wordt de filter instelling altijd bewaard.
Extra filter	Er kan een extra filter worden ingezet om op brandstof, autotype en/of kleur te filteren. Hieronder meer uitleg.

**Filteren van voertuigen**

**Brandstoffen:**  
☐ Hybride  
☐ Benzine  
☐ Diesel  
☐ Elektrisch  
☐ Gas  
☐ Waterstof

**Auto types:**  
☐ Sportwagen  
☐ SUV  
☐ Stationwagen  
☐ Sedan  
☐ Terreinwagen  
☐ GT  
☐ Coupé  
☐ Cabriolet

**Voertuig kleuren:**  
☐ Beige  
☐ Blauw  
☐ Bruin  
☐ Geel  
☐ Grijs  
☐ Groen  
☐ Oranje  
☐ Paars  
☐ Rood  
☐ Wit  
☐ Zwart

Werking	Woordje uitleg:
Dynamisch	Alle checkboxes worden dynamisch in code behind samengesteld. Dit wil zeggen als de lijst langer wordt, deze ook automatisch worden weergegeven en kan op gefilterd worden.
Brandstoffen	Als een of meerdere van deze is geselecteerd, worden enkel deze weergegeven die aan dit criterium/criteria voldoen.  Als je op toevoegen klikt keer je terug naar voertuigen met alle resultaten die hieraan voldoen. Daarna kan er verder gefilterd worden op Merk + automodel om nog eens te verfijnen.
Auto types	
Voertuig Kleuren	

Voertuigen	Bestuurders	Tankkaarten	Automodellen				
Naam	Geboortedatum	Rijbewijs	Rijksregister	Tankkaart	Voertuig	Nummerplaat	
Brink Jos	1999-03-01	B	99.03.01-653.59	Ja			
Dambryune Anna	1944-02-03	C, D	44.02.03-263.03	Neen			
De Man Jef	1972-11-28	B	72.11.28-268.93	Neen			
De Phoo Winnie	1922-00-00	A, B	22.00.00-268.97	Ja			
Derdean Jean-Pol	1944-11-11	B7	44.11.11-111.75	Ja			
Frimout Dirk	1950-12-05	D	50.12.05-236.69	Ja	Lexus NX	Z-FZE-256	
Huisman Josje	2002-12-01	D	02.12.01-236.83	Neen			
Leeuwentemmer Tanguy	1987-05-22	B	87.05.22-667.50	Ja			
Rigoir Filip	1976-03-31	B	76.03.31-019.86	Ja	BMW iX	Z-ABC-123	
Rigoir Ingrid	1976-00-31	B,C	76.00.31-019.65	Ja	Land Rover Defender	Z-FGB-136	
Rinoir Jos	1976-00-00	A, C, B+	76.00.00-019.25	Neen			
Schoenmaker Deen	1962-11-02	C	62.11.02-536.07	Neen			
Talpe Louis	2001-12-01	B, C	01.12.01-563.26	Ja			
Van Den Heuvel Loes	1965-01-03	B, V	65.01.03-538.65	Ja	BMW i4	1-GKL-169	

Naam: 

Kies

Rijksregisternummer:

Zoek

Sluiten

Actie	Woordje uitleg:
Bestuurders	Alle bestuurders worden weergegeven met hun relaties van entiteiten. Dit betekent: 1). bestuurder + adres 2). Mogelijk voertuig (ook deze heeft extra entiteiten), 3). Mogelijk tankkaart
Voertuig	Het is overzichtelijk te zien wanneer een bestuurder een tankkaart heeft en/of aan een voertuig gelinkt is. Alle formats zoals rijksregisternummer worden in de lijst weergegeven zoals op een paspoort zelf via IValueConverter.
Nummerplaat	
Filter	Er kan gezocht worden op naam. Eerst achternaam daarna voornaam.
Rijksregisternummer	Er kan ook een rijksregisternummer worden ingegeven en op zoeken klikken. Deze komt eveneens (als enige dan) in de lijst bovenaan te staan.

Voertuigen		Bestuurders		Tankkaarten		Automodellen	
Tankkaartnummer	Geldigheidsdatum	Actief	Pincode	Bestuurder			
1111111111111111121	29/03/2023	Ja	9999				^
1111111111111111122	06/02/2023	Ja					
11202023656545897456	20/02/2022	Neen		Rigoir Filip			
5230525214589785620	16/01/2022	Ja					
5236599887564556355	05/02/2022	Ja					
532098756363208564	05/02/2022	Neen	5864				
5321458975632589725	25/12/2021	Ja					
5345643553212154541	06/01/2022	Ja					
5411202312089765415	29/01/2022	Ja	2006				
5489654425554515226	18/06/2022	Ja	1111				
5642525630154795666	17/03/2022	Ja					
5695565965654562006	19/02/2022	Ja	9637				
5698646313202598756	28/12/2021	Ja	1234	Talpe Louis			
6980002123059874219	19/12/2021	Neen	1258				
8563202145698765298	22/01/2022	Ja					v

Zoekcriteria: Alle tankkaarten

Tankkaartnummer:

Button	Woordje uitleg:
Tankkaarten	Alle tankkaarten die beschikbaar zijn. Bij default alle kaarten ongeacht hun status. Zoals al eerder vernoemd worden alle resultaten beperkt tot 50 resultaten.
Filter	Dit is een meerkeuzemenu met: 1). Alle tankkaarten 2). Enkel tankkaarten die geldig zijn op dat moment 3). Alleen tankkaarten weergeven die vervallen of geblokkeerd zijn
Tankkaartnummer	Geef een tankkartnummer in en klik op Zoek. Wanneer niets is gevonden wordt dat aangegeven.

Voertuigen	Bestuurders	Tankkaarten	Automodellen
Merk	Autmodel	Autotype	
BMW	i4	Coupé	
BMW	iX	Coupé	
BMW	iX3	Sedan	
Land Rover	Defender	Terreinwagen	
Lexus	NX	Sedan	
Toyota	Celica T	Sportwagen	
Toyota	Supra S	Sportwagen	

Autotype:

Autonaam:

Actie	Woordje uitleg:
Automodellen	Een lijst met aangemaakte automodellen. Bij dubbelklik of button “Kies” gaat het updatescherm open.
Autotypes	Deze dropdown geeft de mogelijkheid te zoeken op autotype (+ filter als deze wordt opgegeven).
Filter	Eveneens kan er gefilterd worden op autonaam zonder autotype, dat bestaat uit merk + automodel.

Detail Voertuig

Chassisnummer : GHJKPMAS8JKLPM965

AutoModel : BMW i4 Coupé

Brandstof: Benzine

Type Voertuig : Coupé

Nummerplaat: 1-GKL-169

Kleur : Blauw

Hybride: Neen

Bestuurder:

Naam: Van Den Heuvel Loes  
Rijksregister:  
65.01.03-538.65

Verwijder

Wijzigen

Sluiten

Actie	Woordje uitleg:
Profielweergave	Een luchtige weergave zonder toeters en bellen. Je kan op één oogopslag alles zien over dit voertuig. Wat zijn de kenmerken en is dit voertuig aan een bestuurder gelinkt. Zo ja, wordt de bestuurder rechts weergegeven.
Verwijderen	Wanneer dit voertuig wordt verwijderd wordt er eerst een bevestiging gevraagd. Wanneer dit succesvol is uitgevoerd, wordt dit scherm automatisch gesloten en het zoekscherm geüpdatet.  Het voertuig kan pas verwijderd worden als het geen bestuurder (meer) heeft.
Wijzigen	Een button om alles te wijzigen opent een nieuw Window scherm.



TankkaartDetails

Detail Tankkaart

Uitgeefdatum:

8/12/2021

Tankkaartnummer:

532098756363208564

Geldigheidsdatum:

5/02/2022

Pincode:

5864

Actief:

Geblokkeerd

Brandstoffen:

Geen brandstoffen

Bestuurder:

Geen bestuurder

Blokkeren

Wijzigen

Sluiten

Actie	Woordje uitleg:
Profielweergave	Een luchtige weergave zonder toeters en bellen. Je kan op één oogopslag alles bekijken van deze tankkaart. Wat zijn de kenmerken en is deze tankkaart gelinkt aan een bestuurder. Zo ja, wordt de bestuurder rechts weergegeven.
Brandstoffen	Optioneel kan een tankkaart meerdere brandstoffen hebben. Deze worden allemaal rechts bovenaan weergegeven als die er zijn.
Blokkeren	Wanneer op blokkeren wordt geklikt, wordt eerst een bevestiging gevraagd. Zo ja, dan wordt de kaart direct geblokkeerd. Via wijzigen kan alles beheert worden, ook deze status. Er is een volledige controle hierover.

Detail Bestuurder

**Voornaam:** Loes  
**Achternaam:** Van Den Heuvel  
**Geboortedatum:** 1965-01-03  
**Rijksregisternummer:** 65.01.03-538.65  
**Rijbewijs:** B, V  
**Adres:** Kampioenenstraat 95  
Brussel

**Voertuig:**  
BMW i4  
Chassis: GHJKPMAS8JKLPM965  
Nummerplaat: 1-GKL-169  
  
**Tankkaart:**  
Nr: 5411202312089765415  
Geldig tot: 29/01/2022  
Tankkaart is actief

Verwijder

Wijzigen

Sluiten

Actie	Woordje uitleg:
Profielweergave	Een luchtige weergave zonder toeters en bellen. Je kan op één oogopslag alles bekijken van deze bestuurder. Wat zijn de kenmerken en heeft deze bestuurder een tankkaart en/of een voertuig. De weergave is rechts te zien.
Verwijderen	<p>Een bestuurder kan eveneens verwijderd worden. De voorwaarde is dat deze bestuurder geen tankkaart en geen voertuig (meer) heeft. Er wordt een bevestiging gevraagd.</p> <div> <div>Ben je zeker dat je wilt verwijderen?</div> <div> <div>JA</div> <div>NEEN</div> </div> </div> <p>Voorbeeld van zo'n bevestiging. Wordt meerdere keren gebruikt in de applicatie. De koptekst wordt met dependency injection weergegeven.</p>
Wijzigen	Bestuurder kan gewijzigd worden via deze button. Hieronder een overzicht van de mogelijkheden.

## Update Bestuurder

**Voornaam:**

**Achternaam:**

**Geboortedatum:**

Dag:  Maand:  Jaar:

**Rijksregisternummer:**

**Rijbewijs:**

**Adres:**

**Voertuig:**

Land Rover Defender  
Chassis: DNBSFAZYKM4125695  
Nummerplaat: Z-FGB-136

**Tankkaart:**

Nr: 9654258645455652004  
Geldig tot: 14/12/2021  
Tankkaart is vervallen




Button/Veld	Woordje uitleg:
Update scherm	Deze weergave is ietwat meer drukker door alle knoppen en velden. Daarom ook een onderscheiding van detail.
Adres	Hier kan ook een adres aangemaakt worden als er nog geen bestaat; bestaat er al een adres wordt deze overschreven. Dit mogen lege velde zijn.
Verwijder voertuig	Wanneer een bestuurder een voertuig of tankkaart heeft. Kan dit via deze knop snel verwijderd worden. Er komt dan een andere knop tevoorschijn om terug een andere bestuurder te geven.
Verwijder tankkaart	De veranderingen zijn nog niet definitief. Alleen wanneer er op Updaten wordt geklikt zijn de veranderingen definitief.
Reset	Reset plaatst de originele gegevens inclusief de containments weer.
Rijksregisternummer	Dit kan gewijzigd worden maar wordt eerst gecheckt dat deze wel uniek is.

## Tankaart Updaten

**Uitgeefdatum:**

**Tankaartnummer: \***

**Geldigheidsdatum: \***

**Actief: \*** ☒ Neen ☐ Ja

**Pincode:**

**Brandstoffen:**

Selecteer ▼

[Brandstof toevoegen](#)

**Brandstoffen:**

Geen brandstoffen

**Bestuurder:**

Geen Bestuurder

[Bestuurder selecteren](#)

Tankaart updaten

Reset

Sluiten

Button/Veld	Woordje uitleg:
Brandstof toevoegen	Brandstof toevoegen werkt op dezelfde manier zoals de toevoegscherm.  Wanneer de brandstof wordt toegevoegd en de lijst wordt gewist is het nog niet definitief aangepast. Enkel wanneer updaten wordt uitgevoerd wordt de manager ingeschakeld.
Bestuurder	Dit is hetzelfde als het scherm hierboven. De werking is overal gelijkaardig gemaakt wat voor de gebruiker of klant aangenaam is.
Tankaartnummer	Tankaartnummer kan gewijzigd worden want deze kan namelijk verkeerd ingegeven worden. Let op het feit dat deze nummer uniek moet blijven, daarom wordt dat altijd eerst gecontroleerd door de manager.
Reset	Alle gegevens inclusief containments worden teruggeplaatst; wat gebruiksgemak is.  Wanneer het updaten niet kan voltooid worden zal en drawback plaatsvinden in de achtergrond. De klant of gebruiker merkt daar niets van. Dit om mogelijk te maken dat alles kan teruggezet worden in zijn oorspronkelijk staat als op reset wordt geklikt.

## Update AutoModel

Merknaam: \*

AutoModelNaam: \*

AutoType: \*

Sedan ▼

Verwijder

Updaten

Reset

Sluiten

Actie	Woordje uitleg:
Updaten	Wanneer hier niets wordt gewijzigd wordt er geen update gedaan. De boodschap zegt dat alles up2date is. De drie types worden terug gecontroleerd dat ze verschillend zijn. Deze behoort tot de settings van de app. De automodellen kunnen oneindig keer geselecteerd worden.
Verwijderen	Verwijderen is mogelijk na confirmatie.
Reset	Terugplaatsen van de oorspronkelijk gegevens die zijn binnengekomen via dependency injection.

Het principe is voor alle Windows hetzelfde.

Zoeken geeft via dependency injection door aan detail, en detail aan updaten,

Update geeft dan terug acties door aan detail, en detail terug aan zoeken.

Alles is reverse. Dit zorgt ervoor dat enkel een nieuwe aanvraag van gegeven plaatsvinden wanneer het echt nodig is. Geen onnodige connecties naar een gegevensbank.

Update voertuig

Chassisnummer: \*

KDZSARCD4569VBN56

Nummerplaat: \*

ABNJ123

Hybride: \*

☒ Neen
 ☐ Ja

Brandstof: \*

Diesel

Kleur:

Selecteer

Aantal deuren:

Vier

Automodel: \*

Toyota Celica T Sportwagen

Automodel wijzigen

Bestuurder:

Naam: Frimout Dirk  
 Rijksregister:  
 50.12.05-236.69

Annuleer bestuurder

Update

Reset

Sluiten

Actie	Woordje uitleg:
Voertuig updaten	Tot slot, het updaten van voertuig, gaat op dezelfde wijze zoals andere updates.
Chassisnummer	Bij het wijzigen wordt eerst gecontroleerd dat deze niet in de gegevensbank zit.
Nummerplaat	
Automodel	Dit is altijd dezelfde Window scherm die hergebruikt wordt.
Bestuurder	

## 4. Exceptions & foutmeldingen

Om ervoor te zorgen dat de domeinregels worden afgedwongen zoals de opdrachtgever het wenst worden de nodige exceptions gebruikt met try & catch om foutmeldingen te krijgen voor zowel clientprogrammeur als eindgebruiker.

Elke class heeft zijn eigen exceptions met meldingen als iets fout loopt. Wat de reden ook kan zijn het einddoel is dat het programma stopt en de melding weergeeft. Daarom is het belangrijk overal Try & Catch te gebruiken om deze meldingen op te vangen. De volgende classes hebben een Exception:

AutoModel	Filter
AutoType	Geboortedatum
Bestuurder	Nummerplaat
Brandstof	Pincode
Chassismummer	Rijksregisternummer
CheckFormat	Tankkaart
Voertuig	Voertuigkleur
AutomodelManager	KleurManager
BestuurderManager	TankkaartManager
BrandstofManager	VoertuigManager
BestuurderNepManager	VoertuigNepManager
TankkaartNepManager	AantalDeuren

## 5. Correctheid -en robuustheid:

Een applicatie moet uiteraard goed werken. Het zit soms in kleine details die het verschil maken. Daarom is het een must om de applicatie aan een test te onderwerpen. De xUnit test is een nieuw project binnenin dezelfde solution en heeft als doel de business laag met zijn entiteiten te corrigeren en een bewijs op te leveren van de goede werking ervan.

Daarnaast bestaat er ook een tweede test in de solution, bestaande uit een console applicatie die het bewijs oplevert te kunnen posten, opvragen, updaten en te verwijderen.

We geven eerst een beknopte weergave weer van de test voor de business laag, die bij elke class terugkomt:

Test	Beschrijving
Verplichte velden	De velden die verplicht zijn in te vullen staan in de constructor. We instantiëren een class en controleren dat alle ingegeven argumenten een match is.
ID's	<p>We gaan nu dat een ID nummer gelijk is aan nul als het gaat om een nieuw instantie. Een uitzondering hierop is tankkaart omdat deze gegenereerd wordt door derden.</p> <p>Een negatief getal wordt ook getest en tenslotte een geldige ID nummer behoort ook tot de standaard test situatie.</p>
Unieke nummers & keys	Volgende zaken worden getest of deze wel aan de juiste opgelegde normen voldoen: Chassisnummer, nummerplaat, tankkaartnummer, pincode en rijksregisternummer
Instantie vergelijking	Om instanties van een object te kunnen vergelijken is het vereist equals te override. Twee dezelfde instanties worden vergeleken want deze wordt gebruikt om een relatie los te koppelen.
Relaties testen	<p>Wanneer containment wordt geplaatst, is het belangrijk dat de geplaatste entiteit ook het object herkent waarin het terechtkomt. Deze tests worden als volgt uitgevoerd:</p> <ol style="list-style-type: none"> <li>1). Via method de containment plaatsen (bv.: VoegVoertuigToe)</li> <li>2). Test of deze aanwezig is (bv.: HeeftBestuurderVoertuig)</li> <li>3). Test ook de relatie tussen beide (bv.: HeeftVoertuigBestuurder)</li> <li>4). Daarna wordt deze terug verwijderd (bv.: VerwijderVoertuig)</li> <li>5). Controleer dat het object is verwijderd (bv.: HeeftBestuurderVoertuig)</li> <li>6). Het geplaatste object wordt ook getest via reference type (bv.: HeeftVoertuigBestuurder)</li> <li>7). Object verwijderen wanneer nog geen object is toegevoegd (bv.: VerwijderVoertuig())</li> </ol> <p>Zowel voor toevoegen als verwijderen wordt er ook getest op NULL Deze worden opgesplitst in 8 aparte testen per entiteit dat ingevoegd wordt.</p>
Ingegeven argumenten	Na het instantiëren worden alle velden die ingegeven zijn gecontroleerd dat deze aanwezig zijn en ook een match zijn.
Null / Leeg	Alle argumenten in constructors en methods worden gecontroleerd op null en leeg zijn. Met uitzondering op pincode. Dit kan een lege string representeren



Daarnaast zijn er ook testen voor de business laag die specifiek voor een bepaalde class belangrijk zijn:

Properties	Interface
CheckFormat	<p>Alle formats dat gecontroleerd worden, bevinden zich in een aparte file. Er werd gekozen voor een datumstring te gebruiken voor geboortedatum. Via reguliere expressie wordt deze op een eenvoudige wijze volledig afgedwongen.</p> <p>Elk mogelijk variant dat goed is en fout is wordt getest.</p>
Geldigheidsdatum tankkaart	Een geldigheidsdatum wordt gecheckt dat deze op dezelfde dag nog geldig is. Het is immers geldig tot de dag zelf en niet vervallen op de dag zelf. Eén seconde terug naar gisteren, en de tankkaart is niet meer actief. Precisie tot op de seconde.
Tankkaart blokkeren	Tankkaart direct blokkeren is een must. Er wordt getest dat een tankkaart niet automatisch terugkeert wanneer dat op ongeldig staat.
Tankkaart Actief/Inactief	<p>Wanneer in database een tankkaart geldig is, maar toch blijkt vervallen te zijn, wordt dit opgevangen door het object zelf.</p> <p>Dit is nodig om geen gegevensanomalieën te bekomen in het Tankkaart. Anders staat property Actief op "true" en method IsGeldigheidsdatumVervallen() op "true". Een test bewijst dat de twee altijd en elkaar gelijk zijn.</p>
Pincode	Pincode kan bijgewerkt worden totdat een tankkaart geldig is.
Niet verplichte velden	Niet verplichte velden worden eveneens getest bij verplichte velden. Deze lopen parallel. De ene mag niet leeg zijn en de andere moet kunnen leeg zijn.

Om het testen te vereenvoudigen werd een nep manager geïntegreerd. Dat heeft als doel een geldige instantie op te leveren voor andere class-testen. In het kort de voordelen ervan:

Properties	Interface
Het doel?	Zodra een teamlid een class heeft getest kon hij een geldige instantie maken die door de andere class (lees ander teamlid) kan gebruikt worden.
BestuurderNepManager()	Een 3-tal instanties die door andere classes kunnen gebruikt worden Vb.: VoegBestuurderToe().
TankkaartNepManager()	Vb.: VoegTankkaartToe() Vb.: VoegVoertuigToe()
VoertuigNepManager()	<p>Aan de hand van chassisnummer, tankkaartnummer en rijksregisternummer kan een geldige instantie worden bekomen.</p> <p>Het bleek nuttig te zijn want er moest iets gewijzigd worden en dat moest welgeteld maar drie keer gedaan worden en alle testen maakten daar direct gebruik van. Handig!</p>

Tot slot, wanneer de business laag is voltooid, wordt de data laag voorzien. Uiteraard is het ook voor deze laag interessant te weten of de werking juist verloopt en doet wat het moet doen. De volgende zaken werden op de rooster gelegd:

Actie	Beschrijving
ConnectieString	Connectie string wordt bovenaan in private variabel opgeslagen zodat dependency injection voor ADO kan plaatsvinden.
Aparte Methods	Om het overzicht te behouden staan de codes in verschillende aparte static methods per repository: 1). BestuurderRepoTest() 2). TankkaartRepoTest() 3). VoertuigRepoTest()
Object aanmaken	Instantie met verplichte velden wordt aangemaakt en gepost in de database
Object opvragen	Eenzijds worden de lijsten eerst opgevraagd omwille van FK's en anderzijds worden de nieuwe toegevoegde objecten terug opgevraagd om te testen dat deze daadwerkelijk in de database staan.
Object updaten	Na het checken dat het kan geüpdatet worden, worden alle verplichte velden veranderd. Ook de unieke rijen in database.
Object verwijderen	Tot slotte wordt gecreëerde data altijd verwijderd en gecontroleerd op verwijdering. Met uitzondering van tankkaart want dat mag niet verwijderd worden.

## 6. Documentatie & Bronvermeldingen:

Een lijst naar alle tutorial 's die handig zijn:

Configuratie	Officiële Website naar de configuratie
Visual studio	<a href="https://docs.microsoft.com/nl-nl/visualstudio/windows/?view=vs-2019&amp;preserve-view=true">https://docs.microsoft.com/nl-nl/visualstudio/windows/?view=vs-2019&amp;preserve-view=true</a>
C#	<a href="https://docs.microsoft.com/en-us/dotnet/csharp/">https://docs.microsoft.com/en-us/dotnet/csharp/</a>
.Net Core 5.0	<a href="https://docs.microsoft.com/en-us/dotnet/">https://docs.microsoft.com/en-us/dotnet/</a>
ADO.NET	<a href="https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/">https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/</a>
XUnit Test	<a href="https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-dotnet-test">https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-dotnet-test</a>
Configuration Manager	<a href="https://docs.microsoft.com/en-us/dotnet/api/system.configuration.configurationmanager?view=dotnet-plat-ext-6.0">https://docs.microsoft.com/en-us/dotnet/api/system.configuration.configurationmanager?view=dotnet-plat-ext-6.0</a>
Microsoft SQL-server	<a href="https://docs.microsoft.com/en-us/sql/ssms/release-notes-ssms?view=sql-server-ver15">https://docs.microsoft.com/en-us/sql/ssms/release-notes-ssms?view=sql-server-ver15</a>
Miscrosoft SQL Server Management	<a href="https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15">https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15</a>
WPF	<a href="https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-5.0">https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-5.0</a>
Github Repository	<a href="https://docs.github.com/en/repositories">https://docs.github.com/en/repositories</a>

Een lijst met alle nuttige informatie:

Chassisnummer	<a href="https://nl.wikipedia.org/wiki/Framenummer">https://nl.wikipedia.org/wiki/Framenummer</a>
Rijksregisternummer	<a href="https://nl.wikipedia.org/wiki/Rijksregisternummer">https://nl.wikipedia.org/wiki/Rijksregisternummer</a>
AutoTypes	<a href="https://nl.wikipedia.org/wiki/Lijst_van_autotypen">https://nl.wikipedia.org/wiki/Lijst_van_autotypen</a>

Dankwoord

Met dank aan de wekelijkse coaching van Tom Vande Wiele

**Projectwerk:** Fleet Management App  
**Start -en einde:** 10/2021 – 01/2022  
**Studenten:** Ali Maes, Ahmet Yilmaz, Filip Rigoir  
**Onderwijsinstelling:** HoGent  
**Vakgebied:** Handelswetenschappen -en Bedrijfskunde  
**Opleiding:** Graduaatsopleiding (Go5)  
**Richting:** Programmeren