

Technisch ontwerp

Project CCSB

15 september 2021

ROc van Twente – Scrumgroep

GIETERIJ 200, HENGELO

|  |  |
| --- | --- |
| Auteur | Lucas Huls |
| Datum | 15-9-2021 |
| Versie | 1.0 |

# 1. Versiegeschiedenis

|  |  |  |  |
| --- | --- | --- | --- |
| ***Versie*** | ***Datum*** | ***Naam*** | ***Omschrijving*** |
| 0.1 | 7-9-2021 | Lucas Huls | Opmaak technisch ontwerp |
| 0.2 | 13-9-2021 | Lucas Huls | ERD + Databases |
|  |  |  |  |
| 1.0 | 30-9-2021 | Lucas Huls |  |

# 2. Titels

|  |  |
| --- | --- |
| ***Naam*** | ***Functie*** |
| Sam Elfring | Scrum Master |
| Lucas Huls | Notulist |
| Bram van Nek | Deelnemer |
| Guus Hinrichs | Deelnemer |

Inhoudsopgave

[1. Versiegeschiedenis 1](#_Toc83885740)

[2. Titels 1](#_Toc83885741)

[3. De opdracht 3](#_Toc83885742)

[3.1 Achtergrond 3](#_Toc83885743)

[3.2 Standaards & richtlijnen 3](#_Toc83885744)

[4. Ontwikkelomgeving 3](#_Toc83885745)

[4.1 Hardware 3](#_Toc83885746)

[4.2 Software 4](#_Toc83885747)

[4.3 Communicatie 4](#_Toc83885748)

[4.4 Gegevensopslag 4](#_Toc83885749)

[12 Bijlagen 5](#_Toc83885750)

[12.1 Schermafbeeldingen 5](#_Toc83885751)

[6 Data Dictionary 8](#_Toc83885752)

[5. Gegevensmodel 10](#_Toc83885753)

[5.1 Applicationuser 10](#_Toc83885754)

[7 Database 11](#_Toc83885755)

[9 Gebruikersinterface 11](#_Toc83885756)

[9.1 Schermen 11](#_Toc83885757)

[9.2 Menu’s 11](#_Toc83885758)

[10 Functionaliteiten in diagrammen 12](#_Toc83885759)

[11 Planning 14](#_Toc83885760)

[11 Bronnen en bestanden 15](#_Toc83885761)

# 3. De opdracht

Op 2 Sept 2021 kregen wij de opdracht van de heer van der Stal om een website te maken voor zijn caravan/camper stalling. Momenteel regelt hij de afspraken zelf. Maar aangezien hij niet altijd bereikbaar is leek een website een praktische oplossing. In de opdracht staat duidelijk wat de wensen en eisen zijn. Deze wensen en eisen zijn meegenomen in het beschrijven van dit het plan. Namens onze scrum groep willen we de heer van der Stal bedanken voor de opdracht. In het document zijn de onderwerpen zoveel mogelijk uiteen gezet.

Hengelo, 6 Sept 2021.

## 3.1 Achtergrond

Dit document bevat het technisch ontwerp voor de te realiseren webapplicatie voor Camper en Caravan stalling Bentelo. Op de website kunnen klanten hun afspraken voor het stallen van hun caravan en camper inplannen. De beheerders kunnen de afspraken inzien en aapassen indien dit nodig is. De website gaat gerealiseerd worden met Visual Studio in ASP.NET Core. Daarbij wordt gebruik gemaakt van het MVC ontwerp model. De gekozen programmeertaal is C#. Voor het beheer van de sourcecode wordt gebruik gemaakt van Git en GitHub, waarbij de GIT repositories worden gehost op Azure DevOps. Voor het opslaan van de gegevens is gekozen voor Microsoft SQL Server, versie 2018. Tijdens de testfase wordt gebruik gemaakt van een database op [*https://www.smarterasp.net/*](https://www.smarterasp.net/) Ook de webapplicatie zal worden gehost op Smarter ASP. In een later stadium zullen de kosten van het hosten van de site in kaart worden gebracht.

## 3.2 Standaards & richtlijnen

GIT Commits:  
[+] = Toevoegingen  
[-] = Verwijdering  
[/] = Aanpassing

De code word in zijn geheel gecommentarieerd in het Engels.  
De GitHub commits worden geschreven in het Engels.  
De Microsoft SQL database tables worden geschreven in het Engels

# 4. Ontwikkelomgeving

## 4.1 Hardware

Er word binnen onze groep gebruik gemaakt van 4 Windows computers en een scrumbord waar onze planning op word weergegeven.

## 4.2 Software

We maken het meest gebruik van “Microsoft Visual Studio 2019”. Dit omdat we website gaan realiseren in .NET CORE. Ook word “Microsoft SQL Server Management Studio” Ook gebruiken we GitHub om samen te werekn aan het project. We hebben hiervoor twee branches gemaakt. De branch ‘main’ is het uiteindelijke product. De branch develop is voor het ontwikkelen. De DevOps is voor onze planning. De DevOps word elke dag in de gaten gehouden en elke dag word onze agenda aangepast.

## 4.3 Communicatie

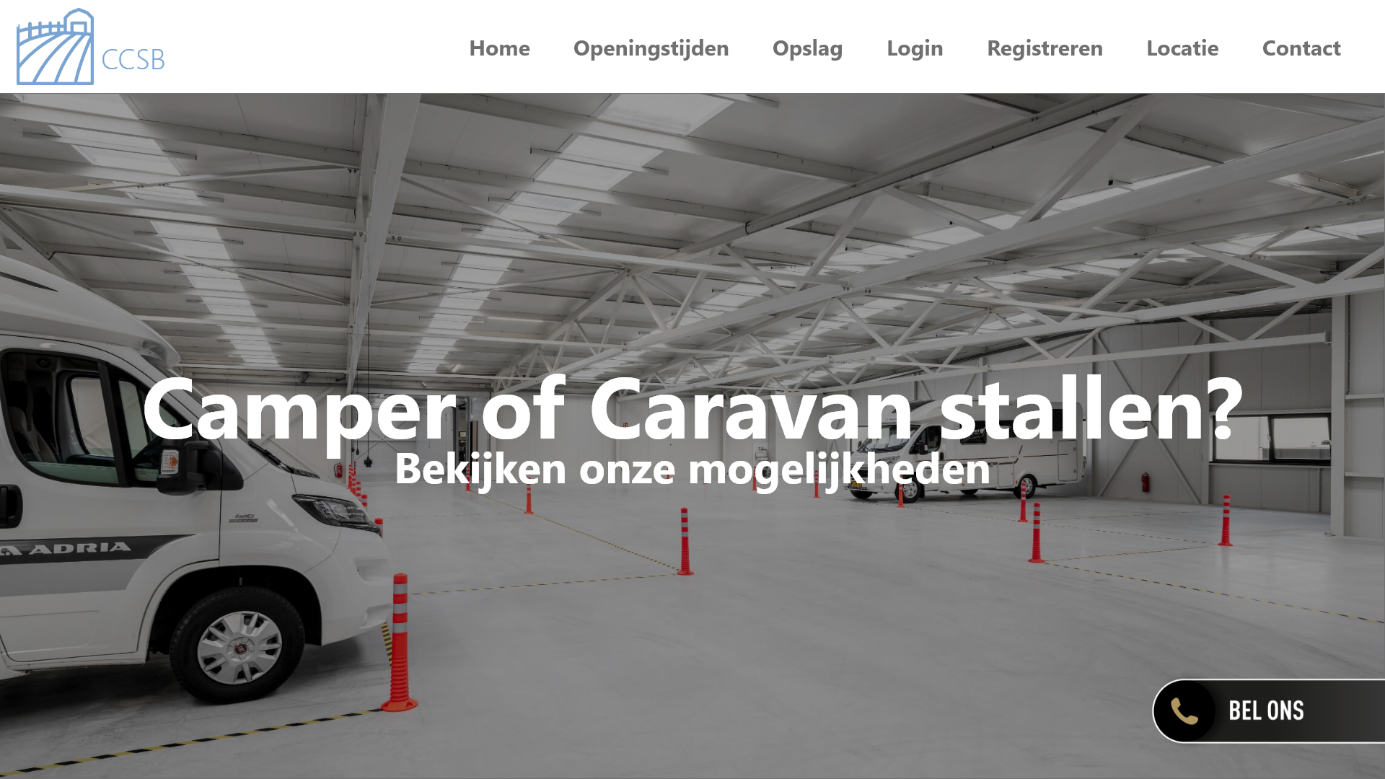
Buiten onze SCRUM uren communiceren we via Discord en Whatsapp. Iedere dag hebben we de dagelijkse stand-up. Dit doen we zodat we duidelijkheid hebben en vragen we de volgende vragen. Wat heb ik bereikt sinds de vorige daily standup? Wat ga ik vandaag bereiken? Verwacht ik obstakels, en kan het team me daarbij helpen?

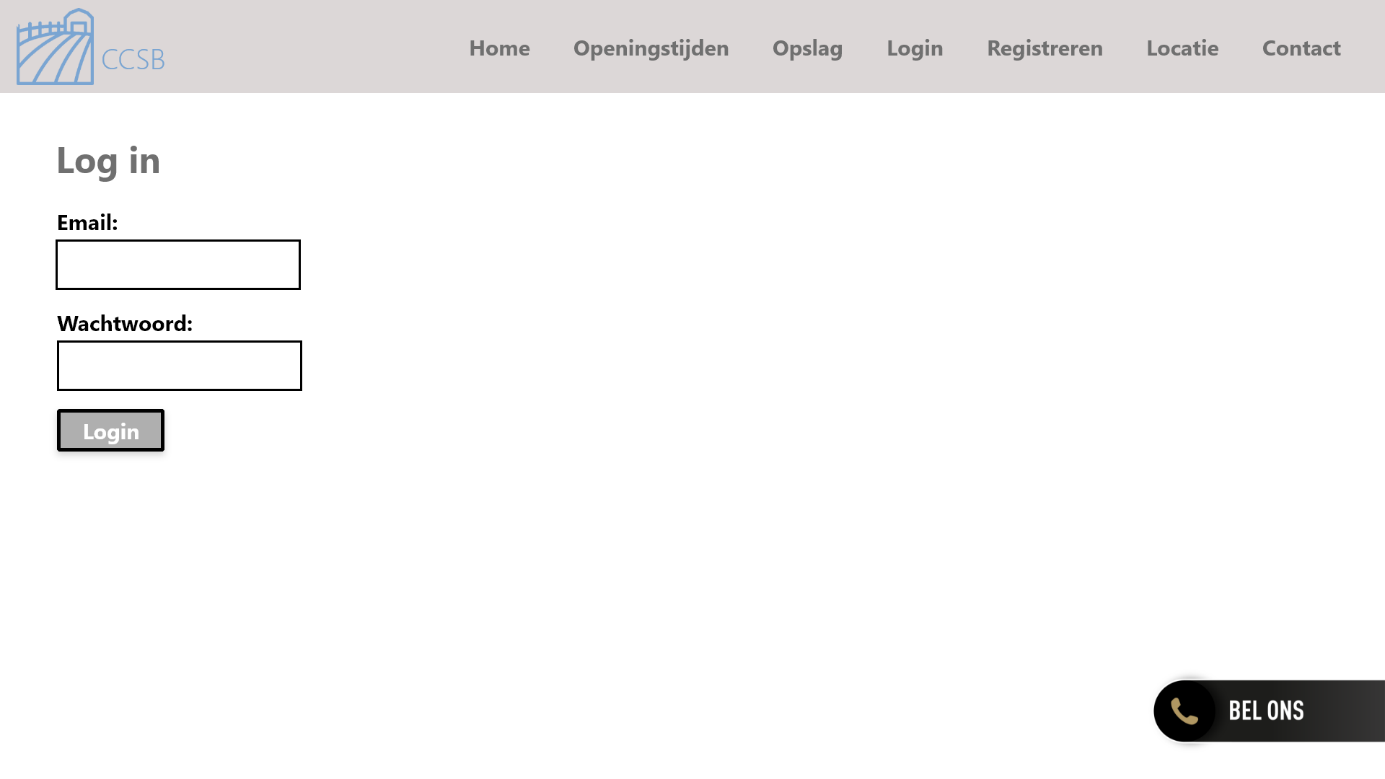
## 4.4 Gegevensopslag

Alle gegevens die opgeslagen dienen te worden in MS SQL. We maken gebruik van een externe hosting omdat we in vorige projecten niet tevreden waren met de SQL hosting van Azure.

# 5 Bijlagen

## 5.1 Schermafbeeldingen

Home  


Beheer inloggen  


Klantenpaneel  


Agenda  
Afbeelding met tafel

Automatisch gegenereerde beschrijving

Registratie  
6 Normaalvormen

Hieronder ziet u de alle normaalvormen van de tabellen die we gaan gebruiken

# 7 Data Dictionary

|  |  |
| --- | --- |
| Tabel: Customer | |
| Deze tabel bevat alle users: klanten & beheerders | |
|  | |
| R1 | Een “customer” is een klant |
| R2 | Een “customer” is een beheerder |
| Opmerkingen | CustomerNumber is een automatisch door SQL Server gegenereerde identifier |

|  |  |
| --- | --- |
| Tabel: Appointment | |
| Deze tabel bevat alle geplande afspraken | |
|  | |
| R1 | AppointmentType is ‘ophalen’ of ‘brengen’ |
| Opmerkingen | Vehicle\_LicensePlate is een foreign key van de table Vehicle |

|  |  |
| --- | --- |
| Tabel: Vehicle | |
| Deze tabel bevat alle opgeslagen voertuigen | |
|  | |
| R1 | Type is ‘Camper’ of ‘Caravan’ |
| R2 | Power is een Boolean met ‘true’ en ‘false’ |
| Opmerkingen | Custome\_CustomerNumver is een foreign key van de table Customer |

|  |  |
| --- | --- |
| Tabel: Contract | |
| Deze tabel bevat alle contracten | |
|  | |
| R1 | Price is staat niet vast |
| Opmerkingen | Vehicle\_\_LicensePlate is een foreign key van de table Vehicle |

# 8. Gegevensmodel

Voor de interactie tussen de gebruiker en de website dienen schermen gebouwd te worden voor het uitvoeren van de functies. In grote lijnen worden de volgende classes onderscheiden:

## 8.1 Applicationuser

Aangezien gebruik gemaakt zal worden van een ASP.NET MVC model met user accounts zal deze class **ApplicationUser**. Aan de class voegen we de volgende properties toe:

|  |  |  |
| --- | --- | --- |
| Properties | Type | Permissies |
| Klantnummer | INT | (Read only) |
| Voornaam | STRING | (Read/Write) |
| Tussenvoegsels | STRING | (Read/Write) |
| Achternaam | STRING | (Read/Write) |
| Voorletters | STRING | (Read/Write) |
| Email | STRING | (Read/Write) |
| Telefoonnummer | INT | (Read/Write) |
| Woonplaats | STRING | (Read/Write) |
| Adres | STRING | (Read/Write) |
| Postcode | STRING | (Read/Write) |
| Geboortedatum | INT | (Read/Write) |
| Rekeningnummer | STRING | (Read/Write) |

Voor het opslaan van de afspraken hebben een class gemaakt genaamd “**Afspraken**”. Aan deze class voegen we de volgende properties toe:

|  |  |  |
| --- | --- | --- |
| Properties | Type | Permissies |
| Datum | INT | (Read/Write) |
| Tijd | INT | (Read/Write) |
| Kenteken | STRING | (Read/Write) |
| SoortAfspraak | STRING | (Read/Write) |

Voor het opslaan van de voertuig gegevens hebben een class gemaakt genaamd “**Voertuigen**”. Aan deze class voegen we de volgende properties toe:

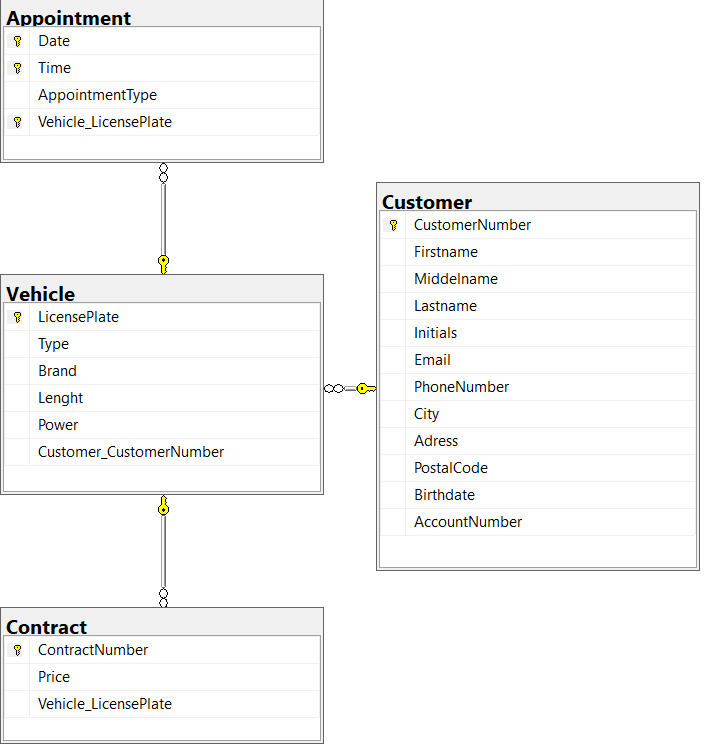
|  |  |  |
| --- | --- | --- |
| Properties | Type | Permission |
| Klantnummer | INT | (Read only) |
| Type | STRING | (Read/Write) |
| Kenteken | STRING | (Read/Write) |
| Merk | STRING | (Read/Write) |
| Lengte | INT | (Read/Write) |
| Stroom | Boolean | (Read/Write) |

Voor het opslaan van de contracten hebben een class gemaakt genaamd “**contracten**”. Aan deze class voegen we de volgende properties toe:

|  |  |  |
| --- | --- | --- |
| Properties | Type | Permission |
| Contractnummer | INT | (Read only) |
| Prijs | STRING | (Read/Write) |
| Kenteken | STRING | (Read/Write) |

# 9 Database

Als we het ERD vertalen naar een SQL Server database, dan kunnen we het volgende database diagram genereren:



*Het script om de database te genereren is als bijlage toegevoegd.*

# 10 Gebruikersinterface

Voor de interactie tussen de gebruiker en de website dienen schermen gebouwd te worden voor het uitvoeren van de functies.

## 10.1 Schermen

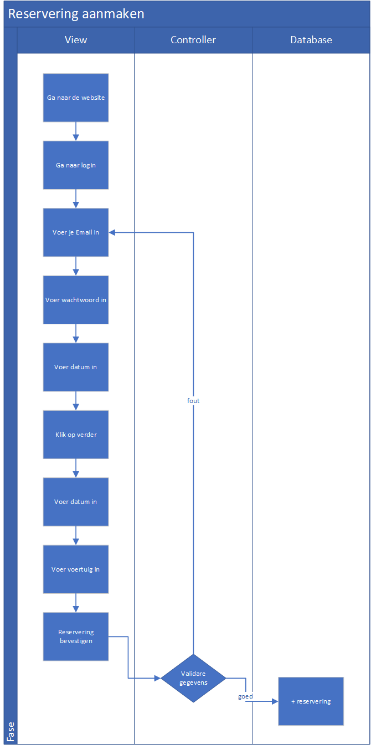
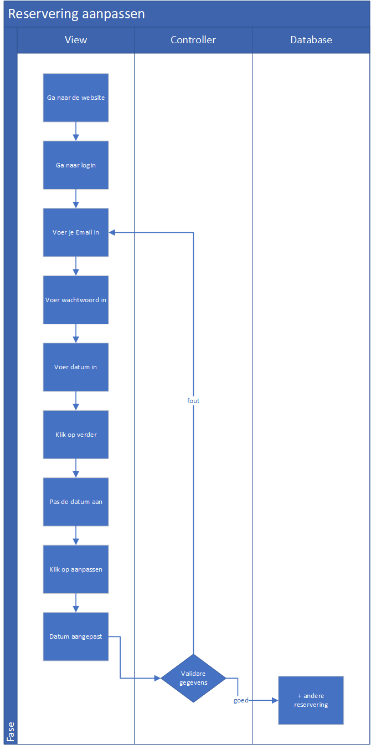
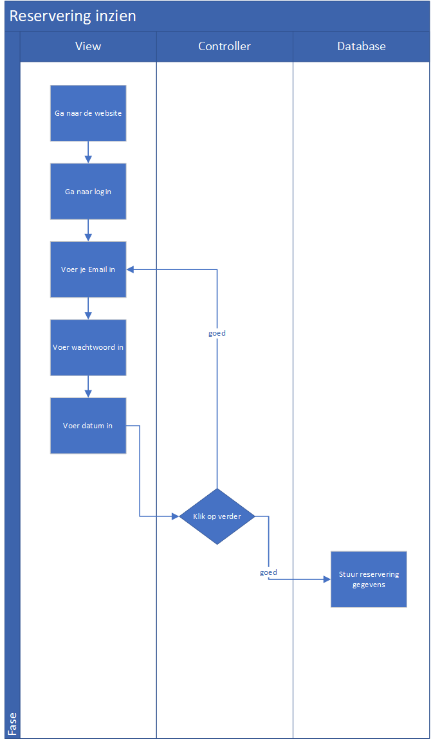
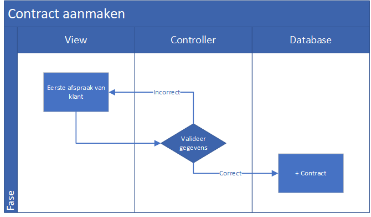
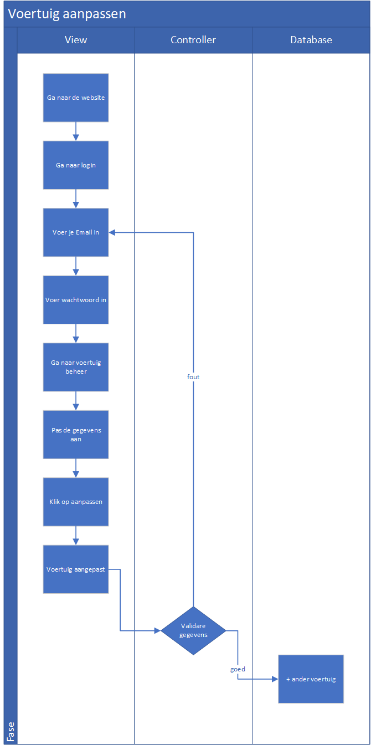
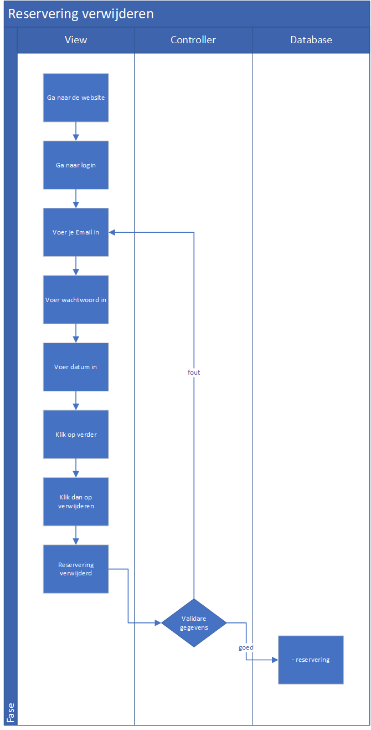
Voor de complete omschrijving van de verschillende schermen verwijzen we u door naar het [functioneel ontwerp](Functioneel%20Ontwerp.docx).

## 10.2 Menu’s

We gebruiken de volgende koppen: Home, Openingstijden, Opslag, Login, Registreren, Locatie, Contact.

# 11 Functionaliteiten in diagrammen

In het functionele ontwerp zijn de gewenste functionaliteiten uitgetekend in een USE-CASE diagram. Het toevoegen van een afspraak aan de kalender kunnen we als volgt schematisch weergeven in een activiteitendiagram:



Reservering verwijderen

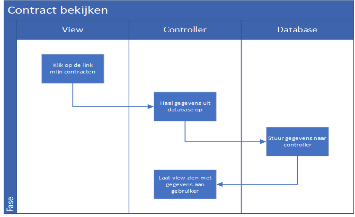
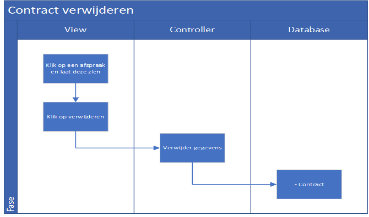
Voertuig aanpassen

Contract aanmaken

Reservering inzien

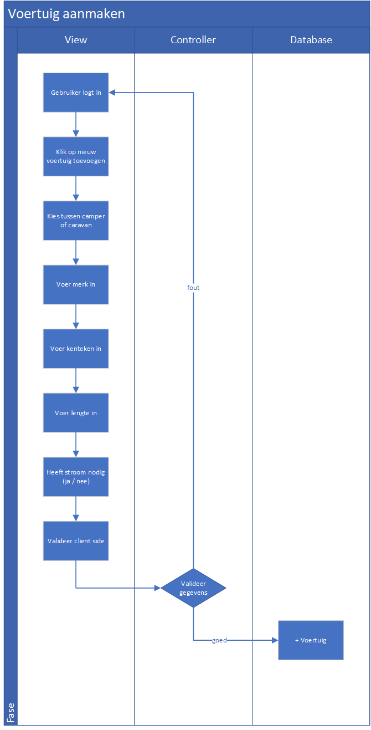
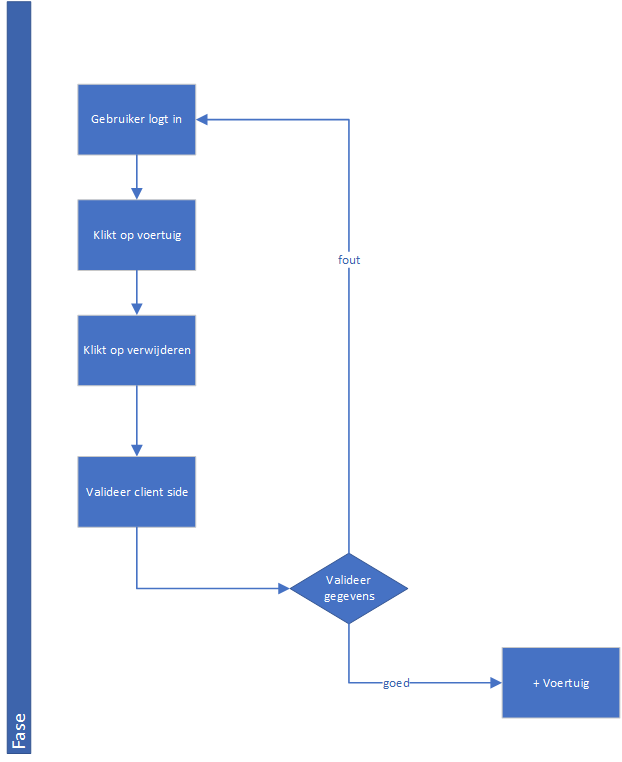
Reservering aanpassen

Reservering aanmaken



Contract verwijderen

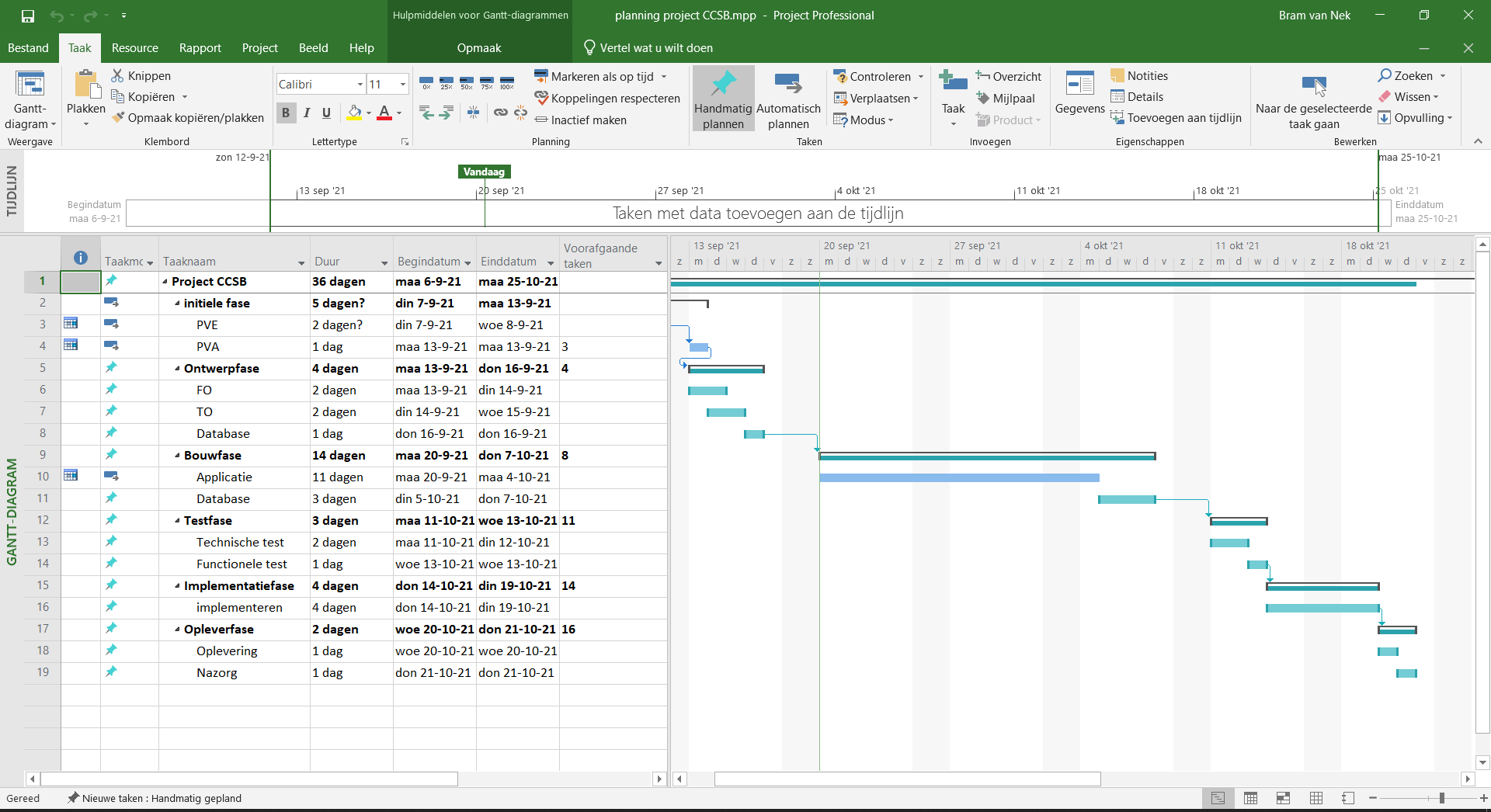
Contract bekijken



Voertuig aanmaken

Voertuig verwijderen

# 12 Planning

De globale planning van het project is reeds weergegeven in het functioneel ontwerp. In dit rapport wordt ingezoomd op de planning van de realisatie, welke staat gepland van 6 september 2021 tot en met 25 oktober 2021.

# 13 Bronnen en bestanden

