2022

Thijmen Hogenkamp & Stefan Verkuijlen

O2GO

28-10-2022

Technisch Ontwerp

Afbeelding met zwart

Automatisch gegenereerde beschrijving

# Voorwoord

Dit document is geschreven in het kader van de specialisatie Advanced App Development als 4e leerjaarstudent van de opleiding bachelor Software Engineering aan het Saxion. Het document is geschreven voor de opdracht die geld van week drie tot en met week acht. De auteurs van het document zijn Thijmen Hogenkamp en Stefan Verkuijlen.

Inhoudsopgave

[Voorwoord 1](#_Toc119175410)

[Inleiding 3](#_Toc119175411)

[Requirements 4](#_Toc119175412)

[User Stories 5](#_Toc119175413)

[Entity Relationship Diagram 13](#_Toc119175414)

[Uitleg keuzes & complexere stukken 18](#_Toc119175415)

# Inleiding

In dit document komen de technische aspecten aan de orde. Hier komen de requirements, user stories, het entity relationship diagram en een uitleg van keuzes en complexere stukken van de applicatie aan bod.

Even wat meer over het idee. Het idee is om een gemakkelijkere manier te hebben om scooters te huren. Momenteel zijn er veel van deelscooters te zien op de weg, maar daar wilt men vanaf stappen. Het doel is om mensen die geen geld hebben voor een eigen scooter of het geld niet waard vinden te helpen toch op een gemakkelijkere manier te laten ‘reizen’. O2GO is de app gemaakt voor hun. Er is een mogelijkheid om een scooter te huren voor een bepaalde tijd en prijs, de mogelijkheid om het toevoegen van pakketten met wellicht korting of verzekeringen. Zit er in een pakket een bepaalde verzekering die je wilt, maar vindt je de andere verzekeringen nutteloos? Geen probleem, dan kun je alsnog die verzekering toevoegen.

# Doel applicatie

Het doel van de applicatie is voor de gebruikers om scooters te kunnen huren. Deze scooters zijn bedoelt om voor een langere periode te huren, waardoor het eigenlijk ‘in bezit’ van de huurder komt. Als een scooter gehuurd is, is deze ook alleen te gebruiken door de huurder. De pakketten en de verzekeringen zijn er om de gehuurde scooter te verzekeren. Een pakket is eigenlijk een groep verzekeringen bij elkaar, maar dan goedkoper dan alle verzekeringen los af te sluiten. De mogelijkheid om een verzekering ook los af te sluiten is er, omdat als iemand een bepaalde verzekering wilt terwijl deze niet in een juist pakket voor diegene zit, kan hij de verzekering ook los afsluiten. In de applicatie moet er ook een mogelijkheid zijn om service aan te vragen, bijvoorbeeld een zomer- of wintercheck. Ook mocht er bijvoorbeeld iets motorisch mis gaan, moet dit kunnen worden geregeld door middel van het aanvragen van service.

# Requirements

Hieronder staan de requirements die opgesteld zijn. Alle requirements worden op basis van de MoSCoW methode op prioriteit geschaald.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Beschrijving** | **MoSCoW** | **Type** |
| **R01** | De gebruiker moet kunnen registreren | Must | Functioneel |
| **R02** | De gebruiker moet kunnen inloggen | Must | Functioneel |
| **R03** | De gebruiker moet kunnen uitloggen | Must | Functioneel |
| **R04** | De gebruiker moet een scooter plan kunnen afsluiten | Must | Functioneel |
| **R05** | De gebruiker moet een scooter plan kunnen annuleren | Must | Functioneel |
| **R06** | De gebruiker moet service kunnen aanvragen | Must | Functioneel |
| **R07** | Als admin moet ik scooters kunnen aanmaken, aanpassen en verwijderen | Must | Functioneel |
| **R08** | Als admin moet ik verzekeringen kunnen aanmaken, aanpassen en verwijderen | Must | Functioneel |
| **R09** | Als admin moet ik pakketten kunnen aanmaken, aanpassen en verwijderen | Must | Functioneel |
| **R10** | Als admin moet ik admin accounts kunnen aanmaken, aanpassen en verwijderen | Should | Functioneel |
| **R11** | De gebruiker moet meerdere opties hebben voor betaalmethodes | Should | Non-functioneel |
| **R12** | De gebruiker moet een overzicht van kosten kunnen inzien | Should | Functioneel |
| **R13** | De gebruiker moet app feedback kunnen geven | Could | Functioneel |
| **R14** | De gebruiker moet het kleuren thema kunnen aanpassen | Would | Functioneel |

# User Stories

Hier worden de user stories gemaakt en beschreven.

|  |  |
| --- | --- |
| **Titel: Registreren gebruiker** | **R01** |
| *User Story:*  Een gebruiker moet met een email, wachtwoord, voornaam, achternaam en adresgegevens kunnen registreren. | |
| *Criteria:*   * Als gebruiker kan ik op de knop ‘Account aanmaken’ drukken * Als gebruiker kan ik een email invullen * Als gebruiker kan ik een wachtwoord invullen * Als gebruiker kan ik nogmaals een wachtwoord invullen ter bevestiging * Als gebruiker kan ik een voornaam invullen * Als gebruiker kan ik een achternaam invullen * Als gebruiker kan ik een straatnaam invullen * Als gebruiker kan ik een huisnummer invullen * Als gebruiker kan ik een postcode invullen * Als gebruiker kan ik een woonplaats invullen * Alle velden moeten correct ingevuld worden, als dit niet het geval is moet er netjes een foutmelding komen te staan met de juiste informatie * Als gebruiker kan ik, als alle velden correct zijn ingevuld, op de knop ‘Account aanmaken’ drukken om het registreren af te ronden | |

|  |  |
| --- | --- |
| **Titel: Inloggen gebruiker** | **R02** |
| *User Story:*  Een gebruiker moet met een email en wachtwoord kunnen inloggen. | |
| *Criteria:*   * Als gebruiker kan ik een email invullen * Als gebruiker kan ik een wachtwoord invullen * Alle velden moeten correct ingevuld worden, als dit niet het geval is moet er netjes een foutmelding komen te staan met de juiste informatie * Als gebruiker kan ik, als alle velden correct zijn ingevuld, op de knop ‘Inloggen’ drukken om in te loggen | |

|  |  |
| --- | --- |
| **Titel: Uitloggen gebruiker** | **R03** |
| *User Story:*  Een gebruiker moet kunnen uitloggen. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker kan ik op de knop ‘Uitloggen’ drukken om uit te loggen | |

|  |  |
| --- | --- |
| **Titel: Scooterplan afsluiten** | **R04** |
| *User Story:*  Een gebruiker moet een scooterplan af kunnen sluiten. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik op de knop ‘Scooterplan afsluiten’ kunnen drukken * Als gebruiker moet ik een van de scooters kunnen selecteren * Als gebruiker moet ik een pakket kunnen selecteren * Als gebruiker moet ik een losse verzekering kunnen selecteren * Als gebruiker moet ik een periode kunnen selecteren * Als gebruiker moet ik kunnen betalen * Als gebruiker moet ik het proces kunnen afronden | |

|  |  |
| --- | --- |
| **Titel: Scooterplan annuleren** | **R05** |
| *User Story:*  Een gebruiker moet een scooterplan kunnen annuleren. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik meer dan een week tot de startperiode van het contract ingelogd zijn * Als gebruiker moet ik op de knop ‘Mijn account’ kunnen drukken * Als gebruiker moet ik op de knop ‘Bekijk contracten’ kunnen drukken * Als gebruiker moet ik op het betreffende contract kunnen drukken * Als gebruiker moet ik op de knop ‘Annuleer contract’ kunnen drukken * Als gebruiker moet ik het annuleren van een contract kunnen bevestigen | |

|  |  |
| --- | --- |
| **Titel: Problemen melden** | **R06** |
| *User Story:*  Een gebruiker moet problemen kunnen melden. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik in de periode van het betreffende contract zitten * Als gebruiker moet ik op de knop ‘Service’ kunnen drukken * Als gebruiker moet ik het contract kunnen selecteren * Als gebruiker moet ik een beschrijving kunnen toevoegen * Als gebruiker moet ik een foto kunnen toevoegen * Als gebruiker moet ik een datum kunnen toevoegen * Als gebruiker moet ik een servicepunt kunnen selecteren * Als gebruiker moet ik de service kunnen bevestigen | |

|  |  |
| --- | --- |
| **Titel: Onderhoud inplannen** | **R07** |
| *User Story:*  Een gebruiker moet problemen kunnen melden. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik in de periode van het betreffende contract zitten * Als gebruiker moet ik op de knop ‘Service’ kunnen drukken * Als gebruiker moet ik het contract kunnen selecteren * Als gebruiker moet ik een beschrijving kunnen toevoegen * Als gebruiker moet ik een datum kunnen toevoegen * Als gebruiker moet ik een servicepunt kunnen selecteren * Als gebruiker moet ik de service kunnen bevestigen | |

|  |  |
| --- | --- |
| **Titel: Scooter aanmaken** | **R08** |
| *User Story:*  Een admin moet een scooter kunnen aanmaken. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Scooters’ kunnen drukken * Als admin moet ik op de knop ‘Scooter toevoegen’ kunnen drukken * Als admin moet ik een kenteken kunnen invullen * Als admin moet ik een maximum snelheid kunnen invullen * Als admin moet ik een bouwjaar kunnen invullen * Als admin moet ik een beschrijving kunnen invullen * Als admin moet ik een prijs (per maand) kunnen invullen * Als admin moet ik de scooter kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Scooter aanpassen** | **R08** |
| *User Story:*  Een admin moet een scooter kunnen aanpassen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Scooters’ kunnen drukken * Als admin moet ik, bij de betreffende scooter, op het potloodje kunnen drukken * Als admin moet ik het veld wat aangepast moet worden kunnen aanpassen * Als admin moet ik de scooter kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Scooter verwijderen** | **R08** |
| *User Story:*  Een admin moet een scooter kunnen verwijderen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Scooters’ kunnen drukken * Als admin moet ik, bij de betreffende scooter, op het prullenbakje kunnen drukken * Als admin moet ik het verwijderen kunnen bevestigen * Als admin moet ik de scooter kunnen verwijderen | |

|  |  |
| --- | --- |
| **Titel: Verzekering aanmaken** | **R09** |
| *User Story:*  Een admin moet een verzekering kunnen aanmaken. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Verzekeringen’ kunnen drukken * Als admin moet ik op de knop ‘Verzekering toevoegen’ kunnen drukken * Als admin moet ik een naam kunnen invullen * Als admin moet ik een beschrijving kunnen invullen * Als admin moet ik een prijs kunnen invullen * Als admin moet ik de verzekering kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Verzekering aanpassen** | **R09** |
| *User Story:*  Een admin moet een verzekering kunnen aanpassen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Verzekeringen’ kunnen drukken * Als admin moet ik, bij de betreffende verzekering, op het potloodje kunnen drukken * Als admin moet ik het veld wat aangepast moet worden kunnen aanpassen * Als admin moet ik de verzekering kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Verzekering verwijderen** | **R09** |
| *User Story:*  Een admin moet een verzekering kunnen verwijderen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Verzekeringen’ kunnen drukken * Als admin moet ik, bij de betreffende verzekering, op het prullenbakje kunnen drukken * Als admin moet ik het verwijderen kunnen bevestigen * Als admin moet ik de verzekering kunnen verwijderen | |

|  |  |
| --- | --- |
| **Titel: Pakket aanmaken** | **R10** |
| *User Story:*  Een admin moet een pakket kunnen aanmaken. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Pakketten’ kunnen drukken * Als admin moet ik op de knop ‘Pakket toevoegen’ kunnen drukken * Als admin moet ik een naam kunnen invullen * Als admin moet ik een beschrijving kunnen invullen * Als admin moet ik een prijs kunnen invullen * Als admin moet ik verzekeringen kunnen selecteren * Als admin moet ik de verzekering kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Pakket aanpassen** | **R10** |
| *User Story:*  Een admin moet een pakket kunnen aanpassen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Pakketten’ kunnen drukken * Als admin moet ik, bij het betreffende pakket, op het potloodje kunnen drukken * Als admin moet ik het veld wat aangepast moet worden kunnen aanpassen * Als admin moet ik het pakket kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Pakket verwijderen** | **R10** |
| *User Story:*  Een admin moet een pakket kunnen verwijderen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Pakketten’ kunnen drukken * Als admin moet ik, bij het betreffende pakket, op het prullenbakje kunnen drukken * Als admin moet ik het verwijderen kunnen bevestigen * Als admin moet ik het pakket kunnen verwijderen | |

|  |  |
| --- | --- |
| **Titel: Admin account aanmaken** | **R11** |
| *User Story:*  Een admin moet een ander admin account kunnen aanmaken. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Admins’ kunnen drukken * Als admin moet ik op de knop ‘Admin toevoegen’ kunnen drukken * Als admin moet ik een email kunnen invullen * Als admin moet ik een wachtwoord kunnen invullen * Als admin moet ik het account kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Admin account aanpassen** | **R11** |
| *User Story:*  Een admin moet een ander admin account kunnen aanpassen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Admins’ kunnen drukken * Als admin moet ik, bij het betreffende account, op het potloodje kunnen drukken * Als admin moet ik het veld wat aangepast moet worden kunnen aanpassen * Als admin moet ik het account kunnen opslaan | |

|  |  |
| --- | --- |
| **Titel: Admin account verwijderen** | **R11** |
| *User Story:*  Een admin moet een ander admin account kunnen verwijderen. | |
| *Criteria:*   * Als admin moet ik ingelogd zijn * Als admin moet ik op de knop ‘Admins’ kunnen drukken * Als admin moet ik, bij het betreffende account, op het prullenbakje kunnen drukken * Als admin moet ik het verwijderen kunnen bevestigen * Als admin moet ik het account kunnen verwijderen | |

|  |  |
| --- | --- |
| **Titel: Overzicht van kosten inzien** | **R13** |
| *User Story:*  Een gebruiker moet een overzicht van kosten kunnen inzien. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik op de knop ‘Mijn account’ kunnen drukken * Als gebruiker moet ik op de knop ‘Kostenoverzicht’ kunnen drukken * Als gebruiker moet ik de data en kosten kunnen zien die gemaakt zijn of nog worden gemaakt | |

|  |  |
| --- | --- |
| **Titel: Feedback geven** | **R14** |
| *User Story:*  Een gebruiker moet een feedback kunnen geven. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik op de knop ‘Feedback’ kunnen drukken * Als gebruiker moet een feedbackveld kunnen invullen * Als gebruiker moet ik de feedback kunnen versturen | |

|  |  |
| --- | --- |
| **Titel: Notificaties ontvangen** | **R15** |
| *User Story:*  Een gebruiker moet notificaties kunnen ontvangen. | |
| *Criteria:*   * Als gebruiker moet ik notificaties kunnen ontvangen wanneer er belangrijke mededelingen zijn | |

|  |  |
| --- | --- |
| **Titel: Kleurenthema aanpassen** | **R16** |
| *User Story:*  Een gebruiker moet het kleurenthema kunnen aanpassen. | |
| *Criteria:*   * Als gebruiker moet ik ingelogd zijn * Als gebruiker moet ik op de knop ‘Instellingen, kunnen drukken * Als gebruiker moet ik een ander kleurenthema kunnen aanklikken * Als gebruiker moet ik de instellingen kunnen opslaan | |

# Entity Relationship Diagram

Hier worden alle tabellen genoteerd en wordt het ERD getoond. Van de tabellen worden de de keys getoond, en waar deze naar toe linken, de namen van de velden, de data types ervan en of ze null mogen zijn. De eerste tabellen zijn zelf bedacht, onder ‘Automatisch gegenereerd’ worden de tabellen getoond die automatisch gegenereerd zijn door ASP zelf.

## Tabellen

**Scooter:**

|  |  |  |  |
| --- | --- | --- | --- |
| Scooter | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Max\_kmh | Int | No |
|  | Brand | Nvarchar(100) | No |
|  | Year | Int | No |
|  | Description | Text | No |
|  | Price | Money | No |

**Service:**

|  |  |  |  |
| --- | --- | --- | --- |
| Service | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Description | Text | No |
|  | Date | DateTime | No |
| FK -> Scooter (id) | Scooter\_id | Int | No |
| FK -> Servicepoint (id) | Servicepoint\_id | Int | No |

**Address:**

|  |  |  |  |
| --- | --- | --- | --- |
| Address | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Street | Nvarchar(100) | No |
|  | Number | Nvarchar(100) | No |
|  | Postal\_code | Nvarchar(8) | No |
|  | City | Nvarchar(100) | No |

**Servicepoint:**

|  |  |  |  |
| --- | --- | --- | --- |
| Servicepoint | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Name | Nvarchar(50) | No |
| FK -> Address (id) | Address\_id | Int | No |

**ApplicationUser:**

|  |  |  |  |
| --- | --- | --- | --- |
| ApplicationUser | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | First\_name | Nvarchar(50) | No |
|  | Last\_name | Nvarchar(50) | No |
| FK -> AspNetUsers (id) | User\_id | Nvarchar(450) | No |
| FK -> Address (id) | Address\_id | Int | No |

**Package:**

|  |  |  |  |
| --- | --- | --- | --- |
| Package | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Name | Nvarchar(50) | No |
|  | Description | Text | No |
|  | Price | Money | No |

**Insurance:**

|  |  |  |  |
| --- | --- | --- | --- |
| Insurance | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Name | Nvarchar(50) | No |
|  | Description | Text | No |
|  | Price | Money | No |

**PackageInsurance:**

|  |  |  |  |
| --- | --- | --- | --- |
| PackageInsurance | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
| FK -> Package (id) | Package\_id | Int | No |
| FK -> Insurance (id) | Insurance\_id | Int | No |

**ContractInsurance:**

|  |  |  |  |
| --- | --- | --- | --- |
| ContractInsurance | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
| FK -> Contract (id) | Contract\_id | Int | No |
| FK -> Insurance (id) | Insurance\_id | Int | No |

**Contract:**

|  |  |  |  |
| --- | --- | --- | --- |
| Contract | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
|  | Start\_date | Nvarchar(50) | No |
|  | End\_date | Nvarchar(50) | No |
| FK -> Package (id) | Package\_id | Int | No |
| FK -> Scooter (id) | Scooter\_id | Int | No |
| FK -> ApplicationUser (id) | User\_id | Int | No |

**Automatisch gegenereerd**

**AspNetUsers:**

|  |  |  |  |
| --- | --- | --- | --- |
| AspNetUsers | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Nvarchar(450) | No |
|  | UserName | Nvarchar(256) | Yes |
|  | NormalizedUserName | Nvarchar(256) | Yes |
|  | Email | Nvarchar(256) | Yes |
|  | NormalizedEmail | Nvarchar(256) | Yes |
|  | EmailConfirmed | Bit | No |
|  | PasswordHash | Nvarchar(max) | Yes |
|  | SecurityStamp | Nvarchar(max) | Yes |
|  | ConcurrencyStamp | Nvarchar(max) | Yes |
|  | PhoneNumber | Nvarchar(max) | Yes |
|  | PhoneNumberConfirmed | Bit | No |
|  | TwoFactorEnabled | Bit | No |
|  | LockoutEnd | Datetimeoffset(7) | Yes |
|  | LockoutEnabled | Bit | No |
|  | AccessFailedCount | Int | No |

**AspNetUserClaims:**

|  |  |  |  |
| --- | --- | --- | --- |
| AspNetUserClaims | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
| FK -> AspNetUsers (id) | UserId | Nvarchar(450) | No |
|  | ClaimType | Nvarchar(max) | Yes |
|  | ClaimValue | Nvarchar(max) | Yes |

**AspNetRoles:**

|  |  |  |  |
| --- | --- | --- | --- |
| AspNetRoles | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Nvarchar(450) | No |
|  | Name | Nvarchar(256) | Yes |
|  | NormalizedName | Nvarchar(256) | Yes |
|  | ConcurrencyStamp | Nvarchar(max) | Yes |

**AspNetRoleClaims:**

<toelichting>

|  |  |  |  |
| --- | --- | --- | --- |
| AspNetRoleClaims | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | Id | Int | No |
| FK -> AspNetRoles (id) | RoleId | Nvarchar(450) | No |
|  | ClaimType | Nvarchar(max) | Yes |
|  | ClaimValue | Nvarchar(max) | Yes |

**AspNetUserLogins:**

|  |  |  |  |
| --- | --- | --- | --- |
| AspNetUserLogins | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | LoginProvider | Nvarchar(450) | No |
| PK | ProviderKey | Nvarchar(450) | No |
|  | ProviderDisplayName | Nvarchar(max) | Yes |
| FK -> AspNetUsers (id) | UserId | Nvarchar(450) | No |

**AspNetUserRoles:**

|  |  |  |  |
| --- | --- | --- | --- |
| AspNetUserRoles | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK, FK -> AspNetUsers (id) | UserId | Nvarchar(450) | No |
| PK, FK -> AspNetRoles (id) | RoleId | Nvarchar(450) | No |

**AspNetUserTokens:**

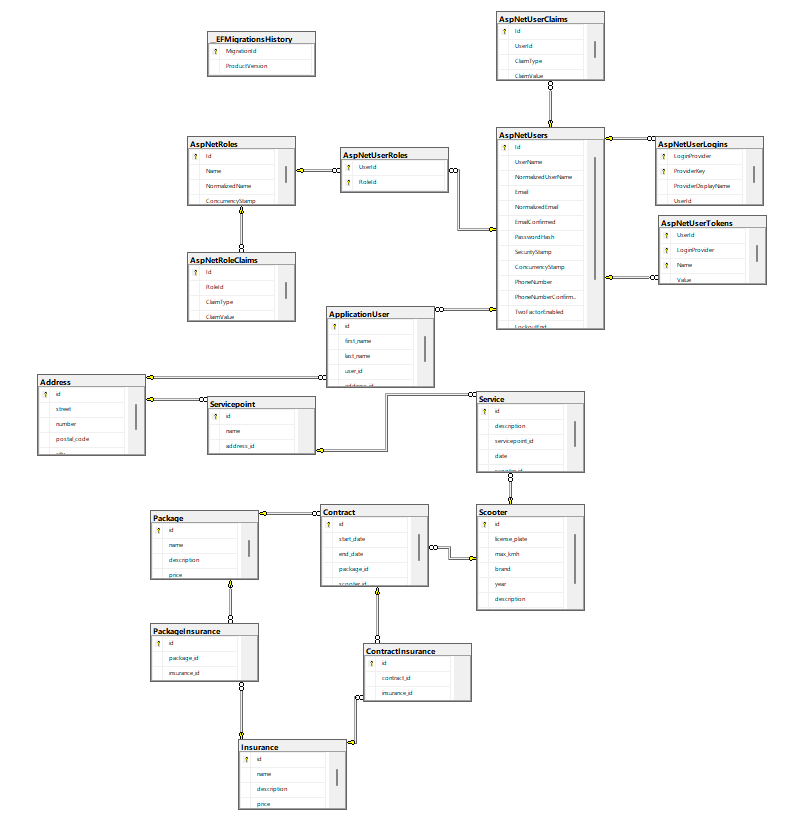
|  |  |  |  |
| --- | --- | --- | --- |
| AspNetUserTokens | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK, FK -> AspNetUsers (id) | UserId | Nvarchar(450) | No |
| PK | LoginProvider | Nvarchar(450) | No |
| PK | Name | Nvarchar(450) | No |
|  | Value | Nvarchar(max) | Yes |

**\_EFMigrationsHistory:**

|  |  |  |  |
| --- | --- | --- | --- |
| \_EFMigrationsHistory | | | |
| Key | **Naam** | **Data type** | **Allow nulls** |
| PK | MigrationId | Nvarchar(150) | No |
|  | ProductVersion | Nvarchar(32) | No |

## ERD

Het ERD bestaat een groot gedeelte uit AspNetUser data. Namelijk de AspNetUsers, AspNetUserLogins, AspNetUserTokens, AspNetUserRoles, AspNetRoleClaims, AspNetRoles en AspNetUserClaims. Deze worden alleen gebruikt voor de authenticatie van de api. Er is hiervoor ook een eigen user aangemaakt, namelijk ApplicationUser. Hierin wordt de voornaam, achternaam, het adres en de AspNetUser gekoppeld. Het adres wordt ook in een aparte tabel bijgehouden, hierin wordt de straat, nummer, postcode en stad bijgehouden. Het adres wordt in een aparte tabel opgeslagen, omdat deze ook gebruikt wordt in de Servicepoint tabel, dit is een tabel met alleen een naam en adres. Deze dient ervoor om voor de gebruiker te bepalen waar de service geleverd moet worden. De Service tabel is er om service aan te vragen, deze tabel heeft een beschrijving, datum, scooter en servicepoint. De Scooter tabel heeft een kenteken, maximale snelheid, merk, bouwjaar, beschrijving en prijs. Om een scooter te huren, moet er een contract afgesloten worden. De tabel Contract heeft daarom ook een begin- en einddatum, scooter, user en package. Er wordt alleen een package gekoppeld en geen losse verzekeringen, omdat een verzekering bij meerdere contracten kan horen en een contract meerdere verzekeringen kan hebben. De Insurance tabel heeft daarom ook alleen een naam, beschrijving en prijs. Hetzelfde geld voor de Package tabel. Omdat de contract en insurance, maar ook de package en insurance, een many-to-many relatie hebben worden er automatisch twee tabellen bij gemaakt, namelijk een PackageInsurance en een ContractInsurance. Deze hebben alleen de id’s van beide kanten. Op de volgende pagina zal de ERD getoond worden, waarin we proberen het zo groot mogelijk af te beelden.



# Swagger documentatie

Als de applicatie door middel van bijvoorbeeld Visual Studio geopend wordt, wordt de swagger automatisch geopend. Als de applicatie door middel van bijvoorbeeld Docker gestart wordt, kan de swagger documentatie gevonden worden via de volgende link: *localhost:<port>/swagger/index.html*.

# Uitleg keuzes & complexere stukken

In dit hoofdstuk worden twee keuzes en twee complexere stukken toegelicht en uitgelegd.

## Complexere stukken

Hier worden de complexere stukken beschreven. Er is gekozen voor deze stukken, omdat de overige stukken van de applicatie makkelijker en minder complex zijn dan deze stukken, want dat is voornamelijk het ophalen en opslaan van data.

### Register

Afbeelding met tekst

Automatisch gegenereerde beschrijving

In de functie wordt een UserRegisterResource meegegeven vanuit de front-end. Hierin staan een e-mailadres (string), wachtwoord (string), voornaam (string), achternaam (string), straat (string), huisnummer (int), postcode (string), stad (string) en admin (boolean). Aan het begin van de functie wordt er een user geinitialiseerd als null, daarna wordt er gekeken of het e-mailadres al bestaat in het huidige userbestand (database). Als hij bestaat wordt hij erin gezet en is hij dus geen null en klopt de registratie niet. Als hij niet bestaat wordt er een nieuwe user aangemaakt met het e-mailadres en het wachtwoord. Als hij een Admin is, krijgt hij ook nog de rol ‘Admin’ toegewezen. Hierna wordt de automatisch gegenereerde gebruiker nog een keer opgezocht om het ID ervan te krijgen en een nieuwe door ons gegenereerde gebruiker toegevoegd (ApplicationUser). Hierna wordt het opgeslagen en teruggegeven, met als wachtwoord ‘Hidden’ om de beveiliging voorop te stellen.

### Login

Afbeelding met tekst

Automatisch gegenereerde beschrijving

In de functie wordt een LoginResource meegegeven vanuit de front-end. Hierin staat een e-mailadres (string) en wachtwoord (string). Aan het begin van de functie kijkt hij eerst of er een gebruiker bestaat met het e-mailadres. Is dit al niet het geval dan komt er een Unauthorized terug. Ook kijkt hij of bij die user het gegeven wachtwoord hoort, is dit ook niet het geval dan komt er weer een Unauthorized terug. Als de user en het wachtwoord bij elkaar horen, worden de rollen opgehaald en de claims toegevoegd. Voor elke rol wordt ook weer een claim toegevoegd. Na dat de rollen als claims zijn toegevoegd, wordt de bearer token aangemaakt en teruggeven. De bearer token is goed voor 8 uur lang, waarna hij expired en de gebruiker dus opnieuw in moet loggen.

#### Keuze

Er is hiervoor gekozen, omdat de gebruiker eigenlijk nooit langer dan 8 uur in de app bezig zal zijn. Het concept is er namelijk voor, samen met het eenvoudige design, dat de gebruiker binnen minimaal 30 minuten al zijn acties kan uitvoeren.

## Uitleg keuzes

Hier worden de keuzes beschreven en toegelicht.

### Ophalen van scooters (en andere modellen)

Het is alleen mogelijk om alle scooters en andere modellen op te halen, zodat er ‘zo weinig mogelijk’ calls naar een externe API gemaakt worden. De scooters en andere modellen worden bij inladen opgehaald en daarna wordt alles afgevangen in de front-end. Zodra er iets geüpdatet moet worden in de database, dus de gebruiker heeft zijn een actie uitgevoerd wat in de database toegepast moet worden, dan wordt de API weer aangeroepen om de actie uit te voeren.

### Waarom pakketten en verzekeringen samen of los kunnen nemen

Het is mogelijk om zowel verzekeringen los, als in pakketten toe te voegen aan een contract. Er is hiervoor gekozen, omdat het mogelijk moet zijn om een pakket naar behoren te kiezen en een verzekering extra toe te kunnen voegen aan het contract. In het ERD zal ook terug te zien zijn dat hier een many-to-many relatie tussen zit, waardoor er automatisch een tussentabel te zien zal zijn.