# Software ontwerp document

## Algemene beschrijving

Hier beschrijf je wat het doel is van de applicatie. Globale beschrijving van wat het doet.

## Gebruikers rollen

Meeste applicaties hebben meerdere gebruikersrollen. Bijvoorbeeld een administrator en een gebruiker. Administrators kunnen nieuwe objecten aanmaken, bewerken en verwijderen. Normale gebruikers kunnen alleen objecten opvragen.

## Functionele eisen

Voor elke gebruikersrol maak je een lijst van wat deze gebruikers moet kunnen in je applicatie. Bijv:

* Alle gebruikers moeten kunnen inloggen
* Administrators moeten zonnepanelen moeten aanmaken

Pas wanneer de applicatie aan al deze eisen voldoet kan je hem bestempelen als “afgerond”

## Database entiteiten

Hier beschrijf je alle entiteiten in je database en welke velden deze heeft. Bijv:

|  |  |
| --- | --- |
| **Entiteit naam** | **Properties** |
| User | firstName: string  lastName: string  age: number |
| Role | name: ‘admin’ | ‘user’ |

## Endpoints

De endpoints die moet maken om te kunnen voldoen aan je functionele eisen en wat het returned.

|  |  |
| --- | --- |
| **Endpoint** | **Return data** |
| GET /user/me | Returned de huidige ingelode user |

## Input Data transfer object

Hier beschrijf je alle input van de gebruikers.

|  |  |
| --- | --- |
| Name | Data |
| LoginDto | Email: string  Password: string |

## Response objects

Hier beschrijf je de objecten die de API output, deze kunnen afwijken van de entiteiten in de database. Bijvoorbeeld: in de database heb je een user met firstName en lastName, maar de API returned een fullName.

|  |  |
| --- | --- |
| **Name** | **Data** |
| UserResponse | firstName: string  lastName: string  fullName: string  age: number |