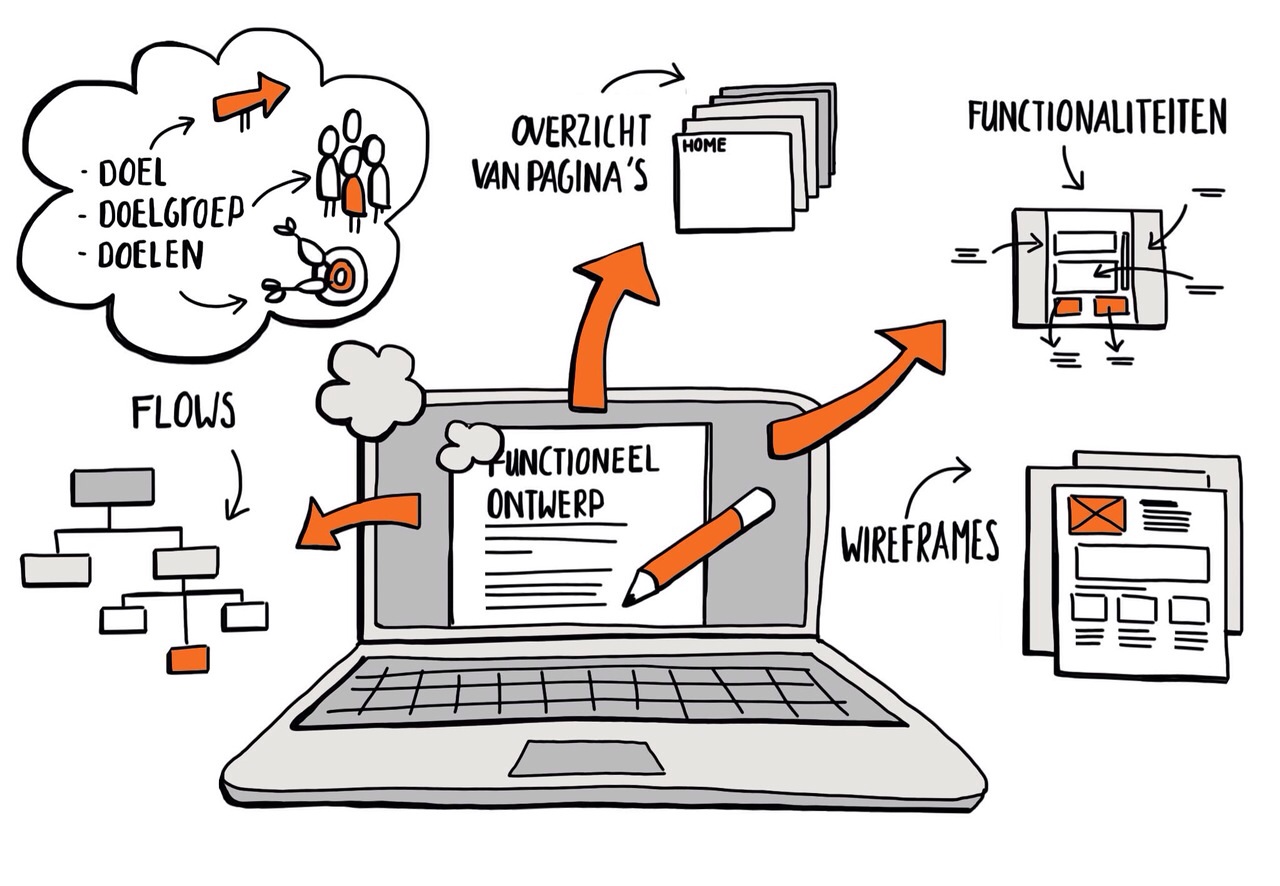
Functioneel ontwerp



Inhoudopgave

[Inleiding 3](#_Toc506795210)

[Use Cases 4](#_Toc506795211)

[Use Case Diagram 4](#_Toc506795212)

[Actor Templates 5](#_Toc506795213)

[Use Case Beschrijvingen 6](#_Toc506795214)

[Use Case Templates + Wireframes 8](#_Toc506795215)

[Wireframe UC001: 9](#_Toc506795216)

[Wireframe UC002: 11](#_Toc506795217)

[Wireframe UC003: 13](#_Toc506795218)

[Wireframe UC004: 15](#_Toc506795219)

[Wireframe UC005: 17](#_Toc506795220)

[Wireframe UC006: 19](#_Toc506795221)

[Datamodel 20](#_Toc506795222)

[Tabel 20](#_Toc506795223)

# Inleiding

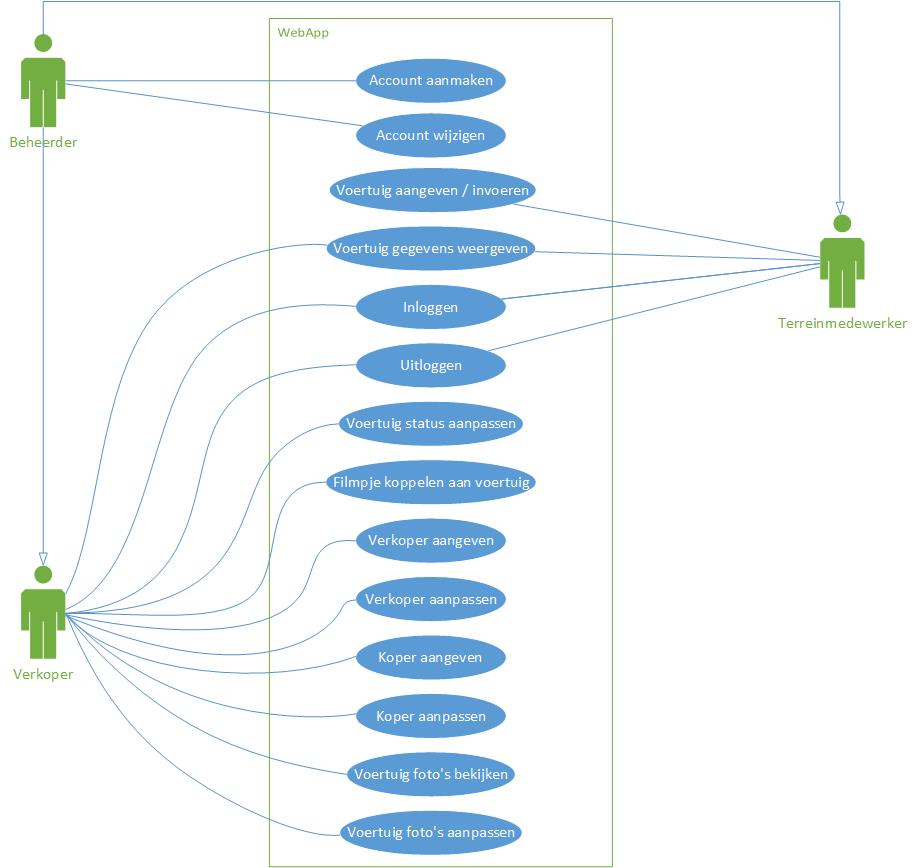
Door Kleyn Group is aan Devin Minnaard, Dustin van Hal en Justin van der Laar gevraagd om een webapp te ontwikkelen die moet voldoen aan een aantal features.

Voor het beginnen met de webapp te maken heb je een functioneel ontwerp nodig. Deze is als volgt opgebouwd. Vanuit het bedrijf zijn er een aantal features bekend gemaakt die in ieder geval in de webapp moeten komen. Daar bovenop zijn er door ons ook nog een aantal features bedacht die volgens ons in de app moeten komen. Na het verzinnen en binnen krijgen van de features zijn wij begonnen met het maken van Use Cases, hierbij is ook een Use Case Diagram gemaakt. Vervolgens zijn de Use Cases uitgewerkt naar Use Case beschrijvingen. Met de Use Case beschrijvingen zijn gemakkelijk de Use Case Templates gemaakt. En uiteindelijk zijn de Wireframes gemaakt.

Natuurlijk kan een Datamodel niet ontbreken aan een functioneel ontwerp. Daarom is er zo snel mogelijk een Datamodel ontwikkelt, deze is gemaakt met behulp van de Use Case Templates.

# Use Cases

## Use Case Diagram



## Actor Templates

|  |  |
| --- | --- |
| **Rolnaam** | Beheerder |
| **Functienaam** | Beheerder Kleyn Group |
| **Omschrijving**  **van de rol** | Beheerder van Kleyn Group, die verantwoordelijk is voor het in de gaten houden van alles wat er op het terrein gebeurt. De beheerder kan een account aanmaken. En deze ook weer inzien en wijzigen. Daarbovenop kan een beheerder alles wat een ‘Terreinmedewerker’ en ‘Verkoper’ kan ook. |

|  |  |
| --- | --- |
| **Rolnaam** | Terreinmedewerker |
| **Functienaam** | Terreinmedewerker Kleyn Group |
| **Omschrijving**  **van de rol** | Terreinmedewerker van Kleyn Group, die verantwoordelijk is voor het aangeven en invoeren van voertuigen. Ook moet de terreinmedewerker de mogelijkheid hebben om de gegevens van het desbetreffende voertuig in te zien. Dit is allemaal mogelijk nadat het desbetreffende persoon is ingelogd met zijn/haar account gegevens. |

|  |  |
| --- | --- |
| **Rolnaam** | Verkoper |
| **Functienaam** | Verkoper Kleyn Group |
| **Omschrijving**  **van de rol** | Verkoper van Kleyn Group, die verantwoordelijk is voor het verkopen van de trucks die op het terrein staan. De verkoper moet de gegevens van een voertuig kunnen inzien en hiervan de status kunnen aanpassen naar, verkocht of te koop. De verkoper moet foto’s / video’s van het desbetreffende voertuig kunnen toevoegen / aanpassen bij het voertuig. Ook moet een verkoper kunnen aangeven / aanpassen wie de koper / verkoper is. |

## Use Case Beschrijvingen

Voertuig aangeven / invoeren, Voertuig gegevens weergeven, Voertuig status aanpassen, Filmpje koppelen aan voertuig, Voertuig foto’s bekijken, Voertuig foto’s aanpassen, Verkoper aangeven, Verkoper aanpassen, Koper aangeven, Koper aanpassen

|  |
| --- |
| **Account aanmaken (UC001)**   * **Gegevens invoeren (Naam, wachtwoord, email)** * **Account wordt aangemaakt** |
| **Account wijzigen (UC002)**   * **Inloggen** * **Selecteer het te wijzigen account** * **Account is gewijzigd** |
| **Voertuig aangeven / invoeren (UC003)**   * **Inloggen** * **Selecteer voertuig aangeven / invoeren** * **Vult gegevens in** * **Voertuig is aangegeven / ingevoerd** |
| **Voertuig gegevens weergeven (UC004)**   * **Inloggen** * **Selecteer voertuig weergeven** * **Lijst met gegevens wordt weergeven** |
| **Voertuig status aanpassen (UC005)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Status wijzigen** |
| **Filmpje koppelen aan voertuig (UC006)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Voeg filmpje toe** |
| **Voertuig foto’s bekijken (UC007)**   * **Inloggen** * **Selecteer voertuig** * **Lijst met gegevens wordt weergeven** |
| **Voertuig foto’s aanpassen (UC008)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Lijst met gegevens wordt weergeven** * **Selecteer foto’s** * **Foto’s worden aangepast** |
| **Verkoper aangeven (UC009)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Verkoper wordt aangegeven** |

|  |
| --- |
| **Verkoper aanpassen (UC010)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Verkoper wordt aangepast** |
| **Koper aangeven (UC011)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Koper wordt aangegeven** |
| **Koper aanpassen (UC012)**   * **Inloggen** * **Selecteer voertuig aanpassen** * **Koper wordt aangepast** |

## Use Case Templates + Wireframes

|  |  |
| --- | --- |
| Use Case nummer | UC001 |
| Use Case naam | Account aanmaken |
| Actor | Gebruiker |
| Samenvatting | Systeem toont een pagina met een aantal input fields**(o).** De actor vult een naam**(i)**, wachtwoord**(i)** en email**(i)**. Het account wordt aangemaakt |
| Pre-Conditie | - |
| Main scenario | 1. Systeem toont een pagina met een aantal input fields**(o)**. 2. Actor vult naam**(i)**, wachtwoord**(i)** en email**(i)** in. 3. Systeem maakt account aan. 4. Systeem beëindigt de Use Case. |
| Post-Conditie | Systeem is niet veranderd. |
| Alternatief scenario | |  |  | | --- | --- | | Trigger | Bij het alternatieve scenario “Actor vult naam, wachtwoord en email in “.  Email is al in gebruik. | | Scenario | 1. Systeem toont een foutmelding. 2. Systeem gaat terug naar stap 1. | |

### Wireframe UC001:

|  |  |
| --- | --- |
| Use Case nummer | UC002 |
| Use Case naam | Afspraak aanmaken |
| Actor | Gebruiker |
| Samenvatting | Systeem toont een pagina met een kalender**(o)**. De actor selecteert een datum**(i)**, hier wordt dan gevraagd naar een begintijd**(i)** en het ras van het hond**(i).** De afspraak wordt aangemaakt. |
| Pre-Conditie | - |
| Main scenario | 1. Systeem toont een pagina met een kalender**(o)**. 2. Actor selecteert een datum**(i)**. 3. Actor selecteert een begintijd**(i)**. 4. Actor selecteert een ras van zijn/haar hond**(i)**. 5. Systeem maakt afspraak aan. 6. Systeem beëindigt de Use Case. |
| Post-Conditie | Systeem is niet veranderd. |
| Alternatief scenario | - |

### Wireframe UC002:

|  |  |
| --- | --- |
| Use Case nummer | UC003 |
| Use Case naam | Afspraak wijzigen |
| Actor | Gebruiker |
| Samenvatting | Systeem toont een pagina met een kalender**(o)**. De actor selecteert een datum**(i)** waar zij/hij een afspraak op heeft gepland. De actor klikt op afspraak wijzigen**(i)**. De actor selecteert een (andere) datum**(i)**, tijd**(i)** en of ras**(i)**. De afspraak wordt gewijzigd. |
| Pre-Conditie | - |
| Main scenario | 1. Systeem toont een pagina met een kalender**(o)**. 2. Actor selecteert de afspraak. 3. Als actor selecteert een datum**(i)**.   3.1 Systeem wijzigt datum.   1. Actor selecteert een begintijd**(i)**.   4.1 Systeem wijzigt begintijd.   1. Actor selecteert een ras van zijn/haar hond**(i)**.   5.1 Systeem wijzigt ras.   1. Systeem wijzigt de afspraak. 2. Systeem beëindigt de Use Case. |
| Post-Conditie | Systeem is niet veranderd. |
| Alternatief scenario | - |

### Wireframe UC003:

|  |  |
| --- | --- |
| Use Case nummer | UC004 |
| Use Case naam | Afspraak verwijderen |
| Actor | Gebruiker |
| Samenvatting | Systeem toont een pagina met een kalender**(o)**. De actor selecteert een datum**(i)** waar zij/hij een afspraak op heeft gepland. De actor klikt op afspraak verwijderen**(i)**. Het system verwijdert de afspraak. |
| Pre-Conditie | Systeem moet een afspraak bevatten. |
| Main scenario | 1. Systeem toont een pagina met een kalender**(o)**. 2. Actor selecteert een datum**(i)**. 3. Actor selecteert verwijder afspraak**(i)**. 4. Systeem verwijdert afspraak. 5. Systeem beëindigt de Use Case. |
| Post-Conditie | Systeem is niet veranderd. |
| Alternatief scenario | - |

### Wireframe UC004:

|  |  |
| --- | --- |
| Use Case nummer | UC005 |
| Use Case naam | Afspraak inzien |
| Actor | Eigenaar |
| Samenvatting | Systeem toont een pagina met een kalender**(o)**. De actor selecteert een datum**(i)**. Het system toont alle afspraken van de dag die de actor selecteert**(o)**. De actor kan alle data van de gebruiker(s) zien. |
| Pre-Conditie | - |
| Main scenario | 1. Systeem toont een pagina met een kalender**(o).** 2. Actor selecteert een datum**(i).** 3. Systeem toont alle afspraken van de dag die de actor selecteert**(o)**. 4. Actor ziet data in. 5. Systeem beëindigt de Use Case. |
| Post-Conditie | Systeem is niet veranderd. |
| Alternatief scenario | |  |  | | --- | --- | | Trigger | Bij het alternatieve scenario “Actor selecteert een datum“.  Er is geen informatie over de afspraken op die dag aanwezig. | | Scenario | 1. Systeem toont een  foutmelding.   1. Systeem gaat terug naar stap 1. | |

### Wireframe UC005:

|  |  |
| --- | --- |
| Use Case nummer | UC006 |
| Use Case naam | Bepaalde datums blokkeren |
| Actor | Eigenaar |
| Samenvatting | Systeem toont een pagina met een kalender**(o)**. De actor selecteert een datum**(i)** waar zij/hij geen afspraak op wilt/kan hebben. Het systeem blokkeert deze datums. |
| Pre-Conditie | - |
| Main scenario | 1. Systeem toont een pagina met een kalender**(o).** 2. De actor selecteert een datum**(i)**. 3. Systeem blokkeert de datum voor verdere afspraken. 4. Systeem beëindigt de Use Case. |
| Post-Conditie | Systeem is niet veranderd. |
| Alternatief scenario | |  |  | | --- | --- | | Trigger | Bij het alternatieve scenario “De actor selecteert een datum “.  De datum is al geblokkeerd | | Scenario | 1. Systeem toont een foutmelding. 2. Systeem gaat terug naar stap 1. | |

### Wireframe UC006:

# Datamodel

## Tabel