Team De druppel – O-S-OOS3-O-S31

Peter Askey  
Nick Kersten  
Inass Madhoun  
Derek Meuwissen  
Bas van Olphen  
Tygo van Ommen

Functioneel Ontwerp

De Druppel

Inhoud

[Versiebeheer 2](#_Toc51599764)

[Project omschrijving 3](#_Toc51599765)

[Concept 3](#_Toc51599766)

[Doel 3](#_Toc51599767)

[Functionele eisen 4](#_Toc51599768)

[Features 4](#_Toc51599769)

[Functionele eisen 4](#_Toc51599770)

[Usecases 5](#_Toc51599771)

[Usecase diagram 5](#_Toc51599772)

[Fully dressed usecases 5](#_Toc51599773)

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Auteur | Datum | Wijzigingen |
| 0.1 | Nick Kersten | 21-09-2020 | Eerste opzet document na oplevering en bespreken individuele documenten teamleden. |
| 0.2 | Nick Kersten | 23-09-2020 | Kleine aanpassingen use cases na uitlijning met werkgroep tijdens virtuele bijeenkomst. |

# Project omschrijving

## Concept

Dit project is voor een druppel-irrigatie systeem. Dit systeem is gericht op klein thuisgebruik voor het gecontroleerd bewateren van planten in bijvoorbeeld een moestuin, bloemenperk of potplanten. Het systeem automatiseert de irrigatie door middel van een microcontroller. Deze meet het vochtgehalte in de grond middels een vochtsensor, en stuurt afhankelijk van minimale en maximale drempelwaardes een klepventiel of pomp aan.

De data over bewateringscycli en vochtgehaltes worden door de microcontroller over het WIFI-netwerk middels een API ge-upload naar een database. Deze data zijn met de API op te vragen.

## Doel

We leven in een steeds drukkere maatschappij waarin duurzaamheid een alsmaar groeiende rol speelt. Met het oog op duurzaam leven willen wij laten zien dat het mogelijk is om met bereikbare middelen op kleine schaal planten en bloemen te verbouwen op een water, tijd -en energie-efficiënte manier.

# Functionele eisen

## Features

Aan de hand van bovenstaand concept zijn de volgende features vastgesteld, gerangschikt volgens MoSCoW:

|  |
| --- |
| Must have |
| Een vochtsensor welke zijn data middels een microcontroller regelmatig opstuurt naar een API. |
| Een API welke de data ontvangt en opslag regelt met een database. Het moet ook mogelijk zijn om deze data via de API op te vragen. |

|  |
| --- |
| Should have |
| Een dashboard met visualisatie van de live en –historische data. |
| Via de microcontroller een klepventiel of pomp kunnen aansturen om planten water te geven. |

|  |
| --- |
| Could have |
| Via het dashboard individuele parameters kunnen instellen waarmee de microcontroller moet werken, bijvoorbeeld voor de frequentie van water geven. |
| Via het dashboard de microcontroller kunnen aansturen om de planten direct water te geven voor een bepaalde periode. |
| Via het dashboard de bewateringscyclus tijdelijk uitschakelen of inschakelen |
| Communiceren met een externe weer API om op basis van de weersverwachting de irrigatie te reguleren. |
| Een temperatuursensor kunnen aansluiten en de data daarvan uitlezen in het dashboard. |
| Een lichtsensor kunnen aansluiten en de data daarvan uitlezen in het dashboard. |
| Een PH-sensor kunnen aansluiten en de data daarvan uitlezen in het dashboard. |

## Functionele eisen

Aan de hand de bovenstaande MoSCoW lijst zijn de onderstaande functionele eisen geïdentificeerd:

* Het bewateringssysteem geeft planten automatisch water als minimale grondvochtigheidswaarden zijn bereikt.
* Het bewateringssysteem stopt met het bewateren van planten als maximale grondvochtigheidswaarden zijn bereikt.
* De gebruiker kan live data over bewateringscycli en grondvochtigheidswaarden inzien via het dashboard.
* De gebruiker kan historische data over bewateringscycli en grondvochtigheidswaarden inzien via het dashboard.
* De gebruiker kan via het dashboard parameters instellen van minimale en maximale bewatering drempelwaardes.
* De gebruiker kan via het dashboard middels een knop handmatig de planten bewateren.

# Usecases

## Usecase diagram

Om de functionele eisen te visualiseren is aan de hand hiervan het onderstaande use-case diagram opgesteld:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Fully dressed usecases

De usecases zijn opgemaakt volgens het fully dressed format.

|  |  |
| --- | --- |
| Use case UC1: plantgegevens inzien | |
| Primary actor: Gebruiker | |
| Stakeholders and interests: gebruiker  Wil snelle, accurate inzage in de plantgegevens. | |
| Brief description:  De gebruiker moet op het dashboard zowel de live plantgegevens als de historische plantgegevens kunnen inzien.  De gegevens bestaan uit het vochtgehalte, de temperatuur en de PH-waarden. | |
| Preconditions: Gebruiker moet verbonden zijn met het dashboard | |
| Postconditions: Geen | |
| Main success scenario: | |
| Actor action | **System responsibility** |
| Gebruiker arriveert op het dashboard | Dashboard toont live en historische gegevens |
| Alternate scenario: systeem kan geen historische gegevens opvragen | |
| Actor action | **System responsibility** |
| Gebruiker arriveert op het dashboard | Dashboard toont alleen live gegevens en meldt dat historische gegevens niet beschikbaar zijn. |
| Alternate scenario: systeem kan geen live gegevens opvragen | |
| Actor action: | **System responsibility:** |
| De gebruiker arriveert op het dashboard. | Dashboard toont de laatste beschikbare historische gegevens en meldt dat live gegevens niet beschikbaar zijn. |

|  |  |
| --- | --- |
| Use case UC2: plant bewateren | |
| Primary actor: Gebruiker | |
| Stakeholders and interests: Gebruiker  Wil met een druk op de knop onmiddellijk bewatering van de planten starten. | |
| Brief description  De gebruiker kan op het dashboard een opdracht geven aan het systeem om de planten te bewateren. | |
| Preconditions:  Gebruiker moet verbonden zijn met het dashboard | |
| Postconditions:  Plant wordt bewaterd. Bewateringsgegevens worden geaggregeerd met historische plantgegevens. | |
| Main success scenario: | |
| Actor action | **System responsibility** |
| De gebruiker arriveert op het dashboard. |  |
| De gebruiker drukt op de knop om de planten te bewateren. | Het systeem geeft weer dat de plant wordt bewaterd voor een bepaalde tijdsduur. |
| Alternate scenario: Systeem kan de pomp niet aanspreken om de plant te bewateren | |
| Actor action | **System responsibility** |
| De gebruiker arriveert op het dashboard. |  |
| De gebruiker drukt op de knop om de planten te bewateren. | Het systeem geeft een contextafhankelijke foutmelding weer. |

|  |  |
| --- | --- |
| Use case UC3: parameters intellen | |
| Primary actor: Gebruiker | |
| Stakeholders and interests: Gebruiker | |
| Brief description:  De gebruiker kan op het dashboard de volgende parameters van het bewateringssysteem kunnen inzien en instellen: de bewateringsfrequentie en de minimale en maximale vochtwaarden alvorens bewatering automatisch start. | |
| Preconditions:  De gebruiker moet verbonden zijn met het dashboard. | |
| Postconditions:  De parameters worden opgeslagen en aangepast in het bewateringssysteem. | |
| Main success scenario: | |
| Actor action | **System responsibility** |
| De gebruiker arriveert op het dashboard. |  |
| De gebruiker drukt op de knop parameters instellen. | Het dashboard geeft de velden van de parameters weer. |
| De gebruiker vult de gegevens in. | Het dashboard slaat de gewijzigde gegevens op |
|  | Het dashboard toont parameters instellen geslaagd. |
| Alternate scenario: blanco parameter invoer | |
| Actor action | **System responsibility** |
| De gebruiker arriveert op het dashboard. |  |
| De gebruiker drukt op de knop parameters instellen. | Het dashboard geeft de velden van de parameters weer. |
| De gebruiker laat de velden blanco. | Het dashboard geeft een foutmelding weer en vraagt om opnieuw invoeren van de gegevens. |