9-4-2024

GoodGarden

LO2E-SWD2

**Justin Doekhi - 6027529**

**Mohammed Çifçi- 6028570**

**Renzo van Putten - 6025850**

**Atilla Oomen - 6028968**

**Burak Diker - 6028083**

Inhoud

[Onderzoek 2](#_Toc163584326)

[Planning 3](#_Toc163584327)

[Moodboard & Logo 4](#_Toc163584328)

[Wireframes 5](#_Toc163584329)

[Design 5](#_Toc163584330)

[Gerealiseerd Product 6](#_Toc163584331)

# Onderzoek

Het onderzoek dat wij hebben verricht was eerst gefocust op de STACK. We waren het er allemaal over eens dat we wel een uitdaging zochten en kozen voor Python als de backend taal. Verder hebben we geen programmeertalen gebruikt die ons vreemd waren. De andere talen zijn namelijk HTML & CSS, ondersteund met JavaScript. Ook de database hebben wij redelijk simpel gehouden, die is namelijk gewoon gemaakt in SQL.

Toen het onderzoek naar de verschillende programmeertalen af was, hebben wij ons onderzoek gericht op informatie inwinnen over planten en de verschillende sensoren die aanwezig zijn in de kas. Voor de planten hadden wij een plan om een API toe te voegen waar je dan mee zou kunnen verbinden op het moment dat je een plant wilt toevoegen aan de kas.

Als laatste hebben we onderzoek gedaan naar algoritmes in zijn geheel. We kwamen er al vrij snel achter dat algoritmes op zichzelf eenvoudig beginnen, maar al snel heel ingewikkeld kunnen worden. Wij hebben, voor nu, gekozen voor een vrij eenvoudig algoritme die berekent hoeveel dagen de batterij nog mee gaat.

Hieronder een korte beschrijving over de gebruikte programmeertalen.

* **Python**

Wij hebben voor de backend Python gebruikt. De reden hiervan, Python wordt steeds populairder wat betekent dat het ook steeds vaker compatible is met verschillende systemen. Ook in de wereld van web ontwikkeling wordt Python steeds vaker gebruikt.

Door de vele mogelijkheden die Python ons te bieden heeft, dachten we dat het een leuk, maar bovenal leerzame en waardevolle toevoeging zou zijn voor onze “stack”.

Wij gebruikte Python vooral voor de verbindingen die gemaakt werden met de database. Ook hebben wij MQTT toegevoegd, wat in Python is geschreven. Met MQTT fetchen wij de data van de API.

We hebben een Flask API gemaakt die ons in staat stelt om data vanuit Python in JSON-formaat te versturen. Met JavaScript fetchen wij dan die data in JSON-formaat om het te tonen op de Frontend.

* **MySQL**

Wij hebben gekozen voor MySQL, omdat het voor ons een betrouwbare en vrij eenvoudige manier voor het database beheer is. Onze database heet “goodgarden” en bestaat uit 11 tafels, waarvan veel gebruikt kunnen worden om de data van het fetchen van de API in op te slaan.

* **Electron**

Onze frontend zal worden gemaakt in Electron. Dit is een Framework gebaseerd op JavaScript en werkt samen met HTML & CSS, waardoor het vrij eenvoudig is om de data te tonen en om het geheel te stylen.

Omdat we Electron gebruiken en dit geen “browser” omgeving is, waren we genoodzaakt om Flask te gebruiken om data vanuit Python direct in Electron weer te geven. Via JSON hebben wij data verwisseld tussen de verschillende programmeertalen.

* **MQTT**

Het fetchen van de sensordata wordt gedaan in MQTT. In het “publish” fetchen wij de data die dan wordt verdeeld in verschillende “topics”. Op die topics kan je dan weer “subscriben” om de data vervolgens in een ander script verder te kunnen verwerken.

# Planning

Uiteraard maken we gebruik van de eerder geleerde technieken. Als eerste gingen we allemaal onderzoek doen van wat er nou precies allemaal nodig was om onze applicatie te realiseren. Nadat het onderzoek was afgerond, hadden we een redelijk overzicht van de mogelijkheden. We hebben, doormiddel van “democratie” de taken verdeelt.

Zo wilden Renzo & Justin het front-end gedeelte doen en Burak & Mohammed gingen aan de slag met Python. Atilla ging overal een beetje bijspringen, maar hield zich vooral bezig met het MQTT gedeelte. Nadat Python helemaal was ingebouwd en werkend was, gingen Burak, Mohammed en Atilla verder om de data te verwerken en de berekeningen/algoritme te maken die nodig waren voor ons project.

We hebben voor dit project geen scrumbord gebruikt, achteraf niet zo heel handig en we hebben ons lesje geleerd. Doordat we dit niet hadden gedaan liep de taakverdeling wat rommelig en had het dus een stuk efficiënter gekund. Dus volgende keer wordt er weer gewoon een scrumbord gemaakt met de bijbehorende onderdelen.

# Moodboard & Logo





# Wireframes

Afbeelding met tekst, schermopname, diagram, ontwerp

Automatisch gegenereerde beschrijving

# Design

Nadat we het logo en het moodboard hebben gemaakt, kwam het design. Er werden twee designs gemaakt, zodra die klaar waren gingen we stemmen welke we zouden gaan gebruiken.

Voor beide designs is er gekozen voor een “dashboard” idee, waar de klant een overzicht krijgt te zien over allerlei verschillend aspecten waarvan wij dachten dat handig zou zijn.

Afbeelding met tekst, schermopname, diagram, ontwerp

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, diagram, ontwerp

Automatisch gegenereerde beschrijving

https://xd.adobe.com/view/84403d73-8cc2-4622-8dfc-f84a63f81ad8-d113/

# Gerealiseerd Product

Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijving