



Rising-IT

Dream IT, build IT

Automatische bemaling

Hilde Hemelaers
Viktor Nagels
Wouter Vandueren
Jan Vanhove
Taro Verschueren
Lander Wuyts

Academiejaar 2022-2023

Inhoud

Inleiding	3
Terminologie	4
Context	5
Hooyberghs	5
De huidige situatie	5
ZD 900 - 600 OPEN FRAME met elektromotor	5
Grundfos SP 5A8	5
Probleemstelling	6
Het doel/ shared vision	6
Webapp	6
Overzicht van de gegevens	7
Automatische aansturing	7
Preventief afsluiten	7
Robuust en foutenbestendig	7
Business value	7
Plan	8
Frontend	10
Database	13
Backend	14
Inlezen van sensor- en pompdata	14
Automatisch aansturen van de pompen	15
Realisatie	16
Webapp	16
Specificaties	16
Overzicht	17
Infrastructuur	21
Server	22
Toonmodel	22
Temperatuurbestendigheid	23
Totale kost	24
Conclusie	25
Bijlagen	26

Inleiding

Als onderdeel van het eindtraject van de opleidingen **Elektronica ICT**¹ en **Toegepaste Informatica**² te Thomas More Geel, hebben wij in opdracht van **Hooyberghs**³ een prototype ontworpen voor het automatiseren van hun grondwater-bemaling op de werf.

Met dit prototype willen wij aantonen dat deze automatische bemaling een meerwaarde is voor Hooyberghs en hun zowel kosten als tijd kan besparen.

In dit document wordt een overzicht gegeven van het probleem op de werven van Hooyberghs, het plan om dit op te lossen en het resultaat van dit project waarin we deze oplossing tot leven brachten. Meer informatie over de precieze uitwerking is te vinden in de folder “Bijlagen”, bijgevoegd aan dit document. Onder elke afbeelding staat een verwijzing naar diens locatie in de folder “Bijlagen”. Als de afbeelding te klein is in dit document, kan men hier een grotere versie vinden.

Dit project liep van oktober 2022 tot februari 2023.

¹ <https://www.thomasmore.be/opleidingen/professionele-bachelor/elektronica-ict>

² <https://www.thomasmore.be/opleidingen/professionele-bachelor/toegepaste-informatica>

³ <https://hooyberghsbouw.be/>

Terminologie

Term	Verklaring
<i>Frontend</i>	<p>Gebruikersomgeving voor een applicatie</p> <p>Bijvoorbeeld: een webapp of website</p>
<i>Backend</i>	<p>Programma dat onzichtbaar is voor de gebruiker, maar veel van de onderliggende logica bevat. Bestaat vaak uit API's.</p> <p>Bijvoorbeeld: een server</p>
<i>Database</i>	<p>Gegevensbank, databank.</p> <p>Een verzameling gegevens, gericht op flexibele raadpleging en gebruik. Speelt een belangrijke rol in het archiveren en actueel houden van gegevens.</p>
<i>Mockup</i>	<p>Het toonmodel waarmee de werf wordt voorgesteld</p>
<i>API</i>	<p>Application Programming Interface.</p> <p>Verzameling van methodes waarmee een computerprogramma kan communiceren met een ander programma over het internet (HTTP).</p> <p>Vaak zijn er verschillende methodes om gegevens op te vragen, te uploaden, aan te passen, of te verwijderen.</p>
<i>Hosting</i>	<p>Het online beschikbaar maken van een website.</p> <p>Men kan een website lokaal doen draaien, op de eigen computer, maar dan is deze niet beschikbaar via een browser over het internet.</p>
<i>Repository</i>	<p>Een collectie van files en folders die online via Git⁴ worden opgeslagen. Voordelen aan het gebruiken van Git repositories (in vergelijking met shared drives) zijn betere samenwerking, makkelijkere rollback (oudere versies ophalen indien iets breekt), makkelijk lokaal coderen, ...</p> <p>Voor dit project zijn de verschillende onderdelen (backend, frontend, ...) onderverdeeld in verschillende repositories.</p>
<i>README.md</i>	<p>De README-file wordt in softwareontwikkeling vaak gebruikt om informatie te geven over de code en scripts aanwezig in een repository. Het is een vorm van documentatie, gericht naar programmeurs.</p> <p>De repositories van dit project hebben elk een eigen README-file, geschreven in Markdown⁵.</p>

⁴ <https://git-scm.com/book/nl/v2/Aan-de-slag-Wat-is-Git%3F>

⁵ <https://nl.wikipedia.org/wiki/Markdown>

Context

Dit project is gestart in september 2022. Omdat dit project over een lange tijdspannen verspreid was, is dit project onderverdeeld in 2 grote “fases”. Oktober tot december zijn besteed aan het onderzoeken van het probleem en diens mogelijke oplossingen. Januari tot en met februari is besteed voor de realisatiefase. In dit document vindt u de volledige samenvatting van het volledig traject, samen met de nodige informatie om op dit project verder te bouwen.

Een project staat echter niet op zichzelf. In dit hoofdstuk wordt een overzicht gegeven van de klant (Hooyberghs) en diens probleem, alsook de oplossing die wij bieden en de meerwaarde van deze oplossing.

Hooyberghs

De klant van dit project is Hooyberghs NV. Als contactpersoon is Frederik Mostmans toegewezen, assistent-projectleider.



Het bedrijf is een bouwgroep gevestigd in Arendonk. Ze realiseren projecten in heel België, met een focus op alle soorten openbare gebouwen, commerciële woonprojecten, sociale woonprojecten en zorginstellingen. Zij leggen de nadruk op het correct opvolgen van de planning en een duurzame mentaliteit. Bij velen van hun bouwprojecten gebeurt de bemaling echter nog vrij inefficiënt en niet duurzaam.

De huidige situatie

Momenteel wordt er gebruikt gemaakt van 2 verschillende systemen voor de bemaling. Als eerste is er een systeem met **ZD 900 - 600 OPEN FRAME met elektromotor** -type pompen. Dit is een oud systeem dat vooral nog gebruikt wordt op landelijke locaties. Er is ook een nieuwer systeem met **Grundfos SP 5A8** -type pompen dat gebruikt wordt in steden. Dit is een duurder systeem dat veel minder ruimte inpakt en geluidloos werkt. Deze werven hebben vaak een oppervlakte van 3000m² tot 3500m².

ZD 900 - 600 OPEN FRAME met elektromotor⁶

Bij dit systeem is er minstens één grote pomp aanwezig. Deze wordt naast de werf geplaatst. Vanuit dit systeem vertrekken er twee grote leidingen die rond de werf gelegd worden. Vanuit deze leidingen vertrekken er dan weer kleinere leidingen die in de grond geboord worden. Deze pomp staat **aan of uit**. Het is dus niet regelbaar hoeveel water er weggepompt wordt. Ook is er bij deze opstelling **niet altijd een sensor aanwezig** die meet hoe diep het grondwater gezakt is. Voor meer info, zie “**Bijlagen\TF - GEHO ZD 900.pdf**”

Grundfos SP 5A8⁷

Dit systeem bestaat uit sensors en pompen. Als eerste zijn er **minstens twee sensors aanwezig** op een werf (idealiter 4, 1tje in elke hoek). Deze worden diagonaal ten opzichte van elkaar geplaatst en meten dan hoe diep het grondwater zit. De pompen zijn **frequentie gestuurd** en worden rechtstreeks in de grond geplaatst. Hieruit loopt een leiding naar de afvoerplaats voor het water. Als er iets mis is met een pomp (er wordt vb. een leiding doorgesneden) valt het **volledige systeem uit**. Het systeem zal zich dan terug proberen op te starten pomp per pomp. Als het systeem de defecte pomp probeert op te zetten, kan dit ervoor zorgen dat het volledige systeem opnieuw uitvalt.

⁶ Voor meer informatie, zie “Bijlagen\TF - GEHO ZD 900.pdf”

⁷ Voor meer informatie, zie “Bijlagen\TF – Grundfos SP 5A8.pdf”

Probleemstelling

Beide pompsystemen hebben een aantal beperkingen. Deze zorgen ervoor dat het pompen minder efficiënt en accuraat gebeurt. Dankzij ons systeem, zullen volgende problemen omzeild worden:

Bij de ZD 900 -types pompen is de huidige waterstand niet gekend, er zijn immers geen sensoren. Hierdoor is het mogelijk dat men pas bij de start van de werken (wanneer het water weggepompt moet zijn) weet of het water eigenlijk laag genoeg staat. In dit geval zullen de **werken uitgesteld** moeten worden.

Bij de Grundfos -types pompen staan wel sensoren, maar komen de metingen slechts dagelijks binnen. Dit zorgt ervoor dat men de pompen dagelijks (manueel) moet bijsturen, gebaseerd op een grote hoeveelheid data in een CSV-formaat (Excel). Dit zorgt ook weer voor **vertraging** en **minder accuraat** pompen.

Wanneer één pomp defecten vertoont, kan dit **het hele systeem plat leggen**. De pompen geraken vaak niet meer van zichzelf opgestart, waardoor er een techniker moet komen, vaak op onregelmatige uren. Dit zorgt nogmaals voor veel tijdsverlies.

Bij beide systemen wordt er dus bijna altijd te weinig of te veel water weggepompt, wat voor **vertraging** en **extra kosten** (elektriciteits- en lozingskosten) zorgt. Ook kunnen de systemen enkel op de werf gemonitord worden, waardoor men geen centraal beheer heeft.

Kortom samengevat: men weet niet hoe goed het pompen verloopt en moet vaak achteraf nog bijpompen. Dit kost het bedrijf tijd en geld.

Het doel/ shared vision

Het doel van dit project is het ontwikkelen van een **efficiënter, milieuvriendelijker & kosten besparend** bemalingssysteem. De volgende onderdelen zijn belangrijke aspecten van dit systeem: een overzicht van de waterstand op de werven, de mogelijkheid tot het automatisch aansturen van de pompen, robuustheid en foutbestendigheid. Dit alles moet beschikbaar zijn op een gebruiksvriendelijke webapplicatie.

Webapp

De webapp biedt een centraal interface dat werfleiders en -medewerkers een overzicht kan bieden van de gegevens op werven, waaronder de waterstand, debiet, energieverbruik van pompen en sensoren, ... Via grafieken en tabellen kan de gebruiker in een oogopslag een beeld krijgen van de vooruitgang van de bemaling op de werf.

Op deze webapp kan de gebruiker ook een gewenste diepte en periode ingeven, waarna het systeem automatisch de pompen zal aansturen om deze diepte over deze periode te bereiken. Om misbruik of menselijke fouten te voorkomen, worden deze waarden gelimiteerd. De diepte moet tussen 5 en -15 meter liggen (TAW-waarden), de periode tussen 5 dagen en 2 weken.

In geval van nood kan het nodig zijn om pompen handmatig aan of uit te zetten. In dit geval kan men op de webapp deze ook manueel aansturen.

Tot slot is het belangrijk dat de handelingen van de gebruikers op deze website worden bijgehouden in een logboek. Men kan dit immers gebruiken als een "paper trail" om de oorzaak en verantwoordelijke te vinden van problemen die kunnen optreden, zoals bijvoorbeeld een verkeerd aangestuurde pomp.

Overzicht van de gegevens

De volgende gegevens worden getoond per werf:

- De huidige waterstand per sensor (staafdiagram)
- Het verloop van de waterstand over tijd, per sensor (grafiek)
- Het verloop van het debiet over tijd (grafiek)
- Het verloop van het energieverbruik per sensor en pomp over tijd (grafiek)
- Logs met de laatste relevante handelingen op de werf, zoals het automatisch aansturen of het manueel afsluiten van pompen (tabel)

Automatische aansturing

De gebruiker heeft op de webapp de optie om in te geven welke op welke diepte het grondwaterniveau zich moet bevinden op een gekozen dag. Met deze informatie zullen de pompen aangestuurd worden om dit niveau gestaag te bereiken over deze periode. Zo wordt overbodig energieverbruik en waterverspilling vermeden.

Om deze berekening te maken, moet het systeem weten wat het verschil is in het niveau van de waterstand wanneer de pompen aan of uitstaan gedurende een uur. De eerste week wanneer een werf is aangesloten op het systeem worden deze standaardwaarden bepaald.

Preventief afsluiten

Om te voorkomen dat het hele systeem stop wordt gezet bij een defecte pomp, moet het systeem detecteren wanneer een defect optreedt en de relevante pomp(en) preventief afsluiten. Zo kunnen de overige pompen hun werking blijven uitvoeren, en hoeft de techniker niet buiten de werkuren aanwezig te zijn.

Wanneer dit gebeurt, moet een verantwoordelijk persoon een melding krijgen, zowel via e-mail als op de webapp.

Robuust en foutenbestendig

Dit systeem zal op een werf gebruikt worden. De kabels en onderdelen moeten blijven werken onder slechte weersomstandigheden en wisselende temperaturen. Ook rijden er vaak zware voertuigen en graafmachines rond op de werf. Het systeem moet dus tegen een stootje en wat vuil kunnen.

Hiernaast kan het zijn dat de gemeten standaardwaarden niet meer accuraat zijn op het moment van het pompen. Weer en temperatuur kunnen dit beïnvloeden, waardoor het opgestelde plan niet meer accuraat is. Het systeem moet hier rekening mee houden en zichzelf dus kunnen bijsturen waar nodig.

Business value

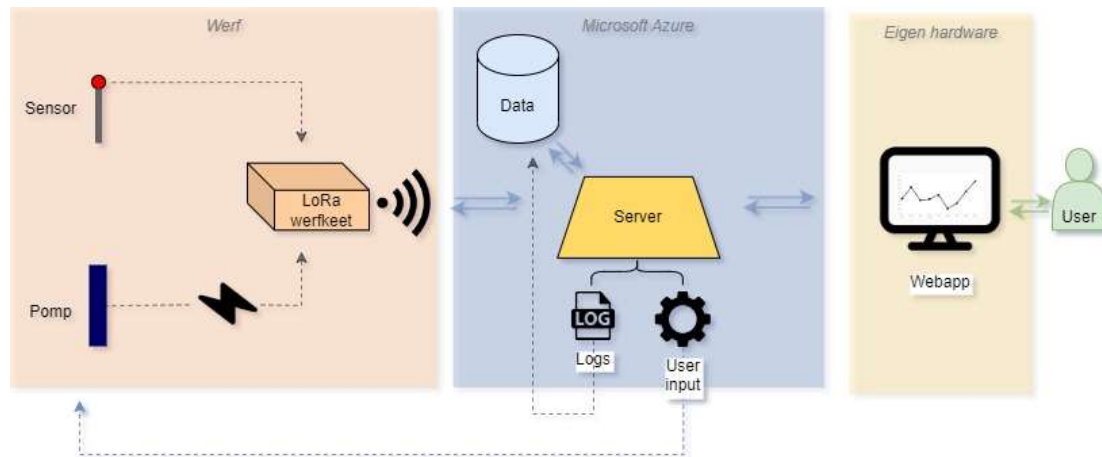
Een project staat niet op zichzelf; het is een onderdeel van het bedrijf. Logischerwijze moet elk project dan ook waarde leveren aan de klant. De geautomatiseerde grondbemaling van Rising-IT zal deze waarde door het vermijden van overbodig pompen.

Het huidige systeem pompt vaak meer grondwater weg dan nodig is, wat resulteert in hoge elektriciteits- en lozingskosten (€0,15 per m³ weggepompt water). Het nieuwe systeem moet de grondwatertafel opvolgen en bijsturen waar nodig. Zo wordt het minimaal vereiste volume aan grondwater weggepompt. Dit resulteert in een meer efficiënte en economische manier om een bemaling te doen.

Plan

In dit hoofdstuk wordt de theorie en denkwijzen achter de werking van het systeem beschreven. Er wordt dus niet ingegaan op de concrete uitvoering en code nodig om het te doen werken, maar eerder de concepten en structuren die gebruikt worden.

Het onderstaande schema geeft een globaal overzicht van de werking van het systeem:

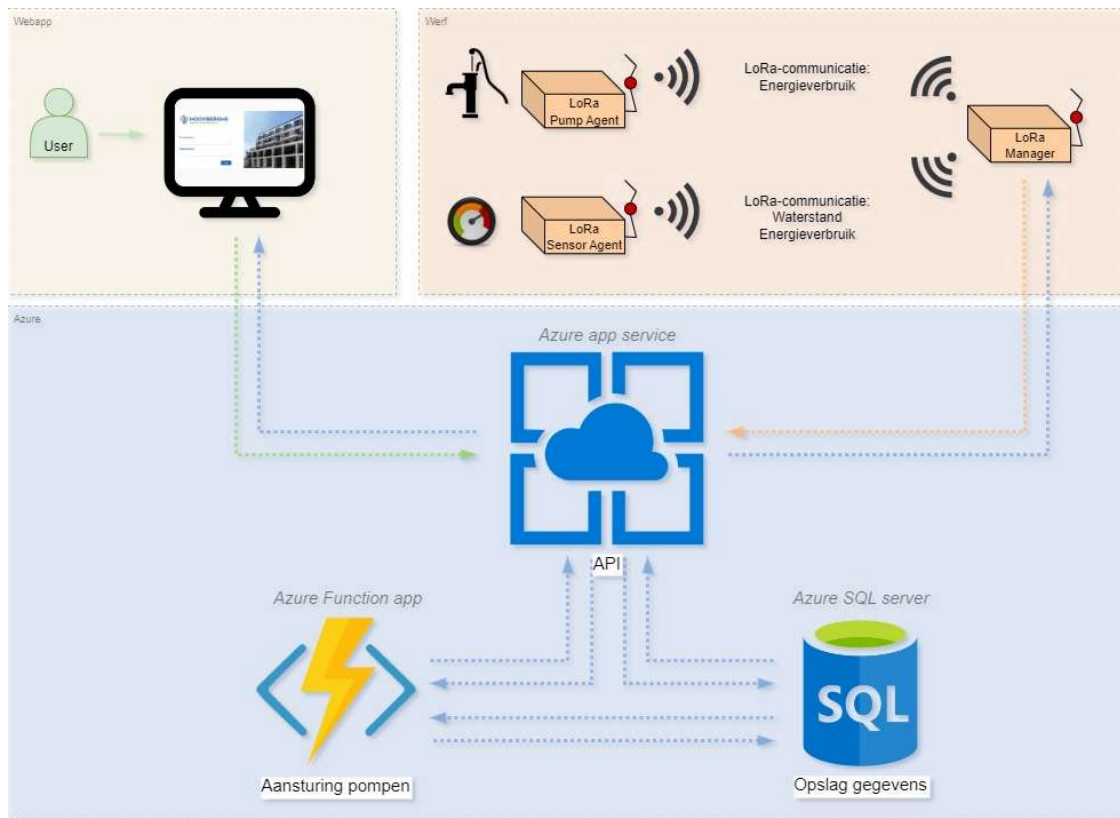


Bijlagen/Plan/Concept – schema.jpg

De gegevens van de pompen en sensoren worden via een LoRa-module⁸ in de werfkeet verstuurd naar de server, die deze in de database zet. De gebruiker kan de gegevens opvragen op de webapp, die deze dan via de server uit de database kan halen. De server gebruikt de user input om de pompen aan te sturen en schrijft van alle acties logs uit naar de database.

Een meer uitgebreid overzicht van de onderdelen en hun technologieën staat op de volgende pagina.

⁸ <https://circuitdigest.com/article/introduction-to-lora-and-lorawan-what-is-lora-and-how-does-it-work>




Bijlagen/Plan/Concept – schema - advanced.jpg

In de volgende hoofdstukken wordt dieper ingegaan op de werking van elk van deze onderdelen.

Frontend


De frontend is de webapp waar de gebruiker mee in interactie gaat. Deze moet dus overzichtelijk en gebruiksvriendelijk zijn. De volgende afbeeldingen geven een indicator van hoe de webapp er uit kan zien. Deze kunnen worden gezien als een ruwe schets. De uiteindelijke webpagina's zijn te zien in het hoofdstuk "Realisatie", onder "Webapp".




HOOYBERGHS
Algemene bouwonderneming

E-mail adres:

Wachtwoord:




Bijlagen/Plan/Frontend/Aanmeldpagina.png



HOOYBERGHS
Algemene bouwonderneming

Adres	Start bemaling	Einde bemaling	Active	View	Settings werk
Straat 126 2734 Stad	23/11/2022	12/02/2023	✓	➔	⚙️
Laan 63 8634 Dorp	03/11/2022	03/01/2023	✓	➔	⚙️
Laan 64 8634 Dorp	18/09/2022	03/11/2022		➔	⚙️
Straat 12 2767 Stad	18/09/2022	02/11/2022		➔	⚙️

Bijlagen/Plan/Frontend/OverzichtWerven.png


HOOYBERGHS
 Algemene bouwonderneming

Adres:
 Straat Nr.
 Postcode Stad

Pompen +
 Kies pomptype
 Locatieomschrijving pomp

Adres:
 Start bemaling
 Einde bemaling

Sensor +
 Locatieomschrijving sensor

Werknemers +
 Kies werknemer
 Kies rol werknemer

Bijlagen/Plan/Frontend/VoegWerfToe.png



Bijlagen/Plan/Frontend/Trendlijn1.png



Bijlagen/Plan/Frontend/Trendlijn2.png

HOOYBERGHS
Algemene bouwonderneming

Grondwaterstand Weggepompt debiet Status Logs Uitloggen

Pomp/sensor	Uur	Status
Sensor	Links	Working
Pomp	Links boven	Working 10%
Pomp	Rechts boven	Working 23%
Pomp	Rechts onder	Not working

Bijlagen/Plan/Frontend/PompStatus.png

HOOYBERGHS
Algemene bouwonderneming

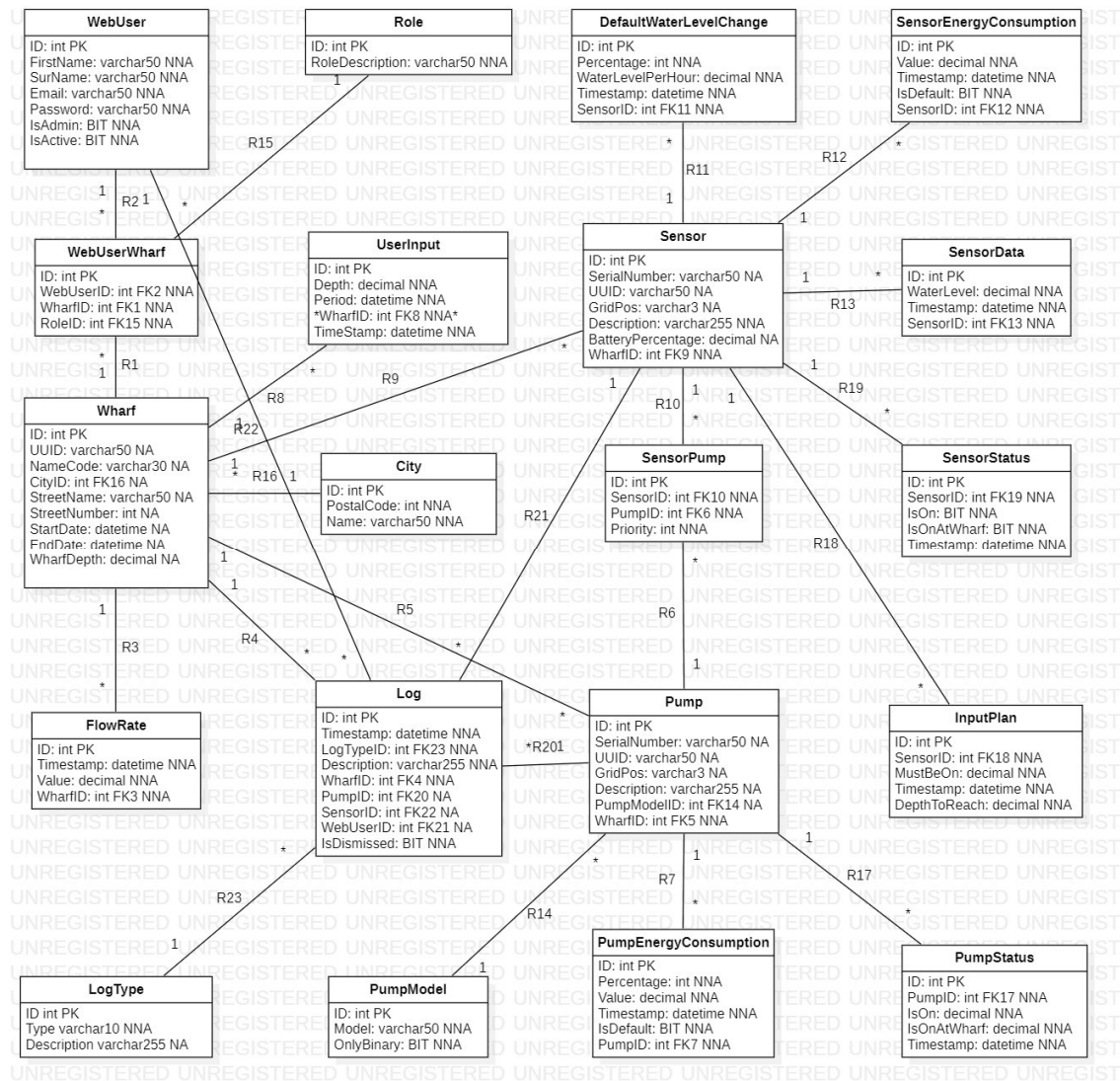
Grondwaterstand Weggepompt debiet Status Logs Uitloggen

Datum	Uur	Melding
29/11/2022	03:02	Sensor 2 automatisch uitgeschakeld
	03:02	Melding vreemde waarden sensor 2 naar werfleider
28/11/2022	23:02	Vreemde waarden sensor 2 hersteld binnen 2 seconden
27/11/2022	14:56	Pomp 1 handmatig ingeschakeld

Bijlagen/Plan/Frontend/PompLogs.png

Database

De database is een SQL-database⁹ in Azure. Zij volgt het volgende database-schema:



Bijlagen/Plan/Database/Database – model.jpg

Voor deze database is gekozen om gebruik te maken van een gestructureerd SQL-model. SQL-data is geschikt voor grote hoeveelheid data op te slaan die een specifieke structuur volgt (denk Excel-tabellen) en in relatie staat met elkaar.

Deze database bevat alle relevante informatie voor dit project. Gebruikers (WebUser), werven (Wharf), pompen (Pump), sensoren (Sensor), al hun onderlinge relaties en alle relevante gegevens verkregen van deze pompen. Deze data wordt opgeslagen in een Azure SQL server.

⁹ <https://youtu.be/5L7TpWkHw-o>

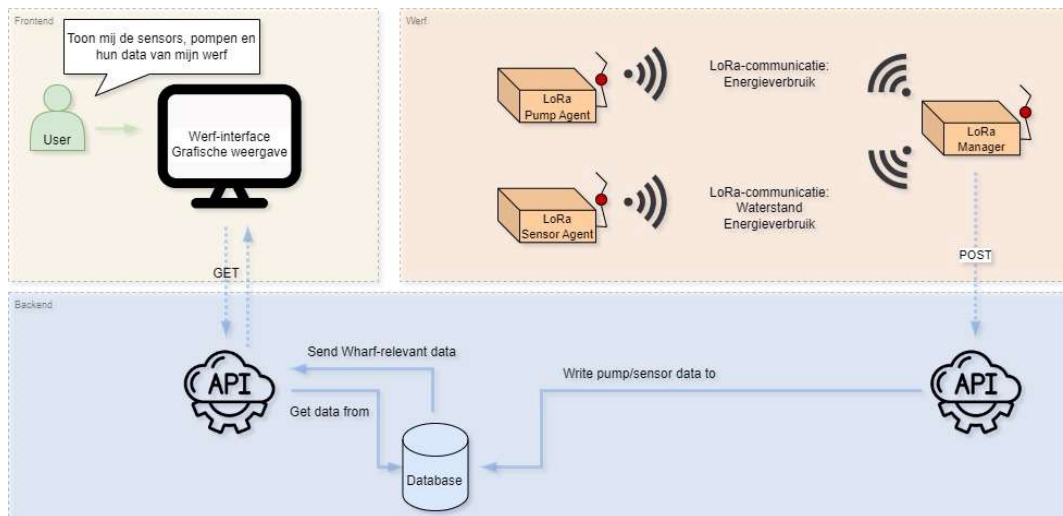
Backend

De server communiceert met de mockup en frontend via API-requests¹⁰. Hiermee worden records aangemaakt of uitgelezen in de database, het automatisch aansturen berekend, ... De werking van deze functionaliteiten is hieronder schematisch afgebeeld.

Inlezen van sensor- en pompdata

Wanneer de gebruiker de laatst gemeten sensorwaarden wil bekijken, gaat deze via de website een API-request maken naar de server. Deze haalt de waarden op uit de database en geeft deze als antwoord terug mee naar de website.

Wanneer de LoRa manager gegevens van de server wil versturen naar de database, doet deze dat ook via een API-request naar de server. De server valideert dit request en schrijft de waarden uit naar de database.



Bijlagen/Plan/Backend/Process flow-View wharf status1.jpg

¹⁰ <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>

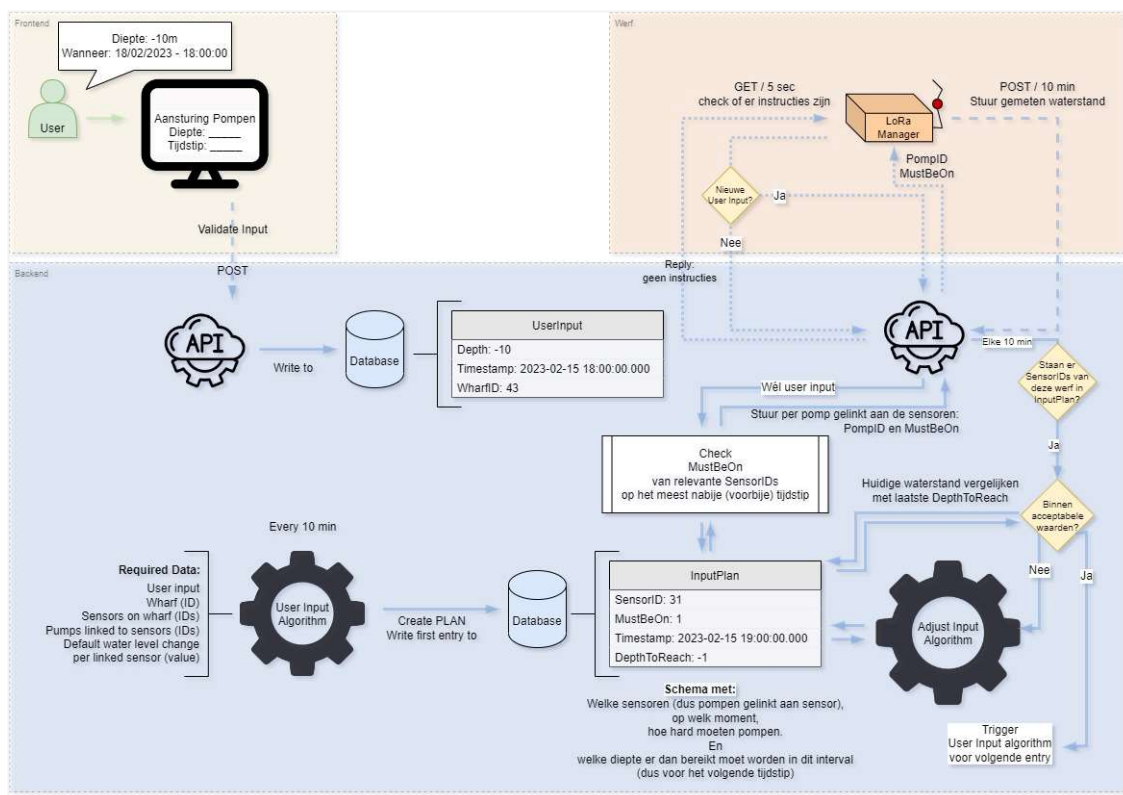
Automatisch aansturen van de pompen

Wanneer de gebruiker een gewenste diepte en periode ingeeft, wordt dit opgeslagen in de database via het API in de server. Elke tien minuten controleert de server of er een nieuwe user input aanwezig is in de database. Zo ja, dan gaat het deze input gebruiken om een plan op te stellen.

In het geval van een frequentiepomp gaat de server via het **User Input Algorithm** de optimale frequentie zoeken om de gewenste diepte over de gewenste periode te bereiken. In het geval dat de pomp enkel op of af kan staan, gaat de server een plan opstellen om de pomp op regelmatige tijdstippen op of af te zetten om de meer precieze frequentie-gestuurde pomp te benaderen.

Op regelmatige tijdstippen gaat de LoRa-manager op de werf de database raadplegen om na te gaan of er nieuwe instructies zijn. Indien het User Input Algorithm deze heeft gegenereerd, worden deze gebruikt om de pompen aan te sturen.

Indien het systeem detecteert dat de gewenste diepte en de gemeten diepte te ver van elkaar afwijken, gaat het **Adjust Input Algorithm** de opgestelde planning aanpassen.



Bijlagen/Plan/Backend/Process flow-User Input.jpg

Een overzicht van de werking van het User Input Algorithm en Adjust Input Algorithm is te vinden in "Bijlagen/Plan/Backend/Process flow-User Input Algorithm.jpg" en "Bijlagen/Plan/Backend/Process flow-Adjust Input Algorithm.jpg".

Realisatie

Van 16 januari 2023 tot en met 3 februari 2023 is dit project gerealiseerd. In deze periode is het gelukt om de gegevens van de pompen via de webapp te tonen. Na een gesprek met de klant is besloten om het preventief afsluiten van de pompen een lagere prioriteit te geven, en de focus te leggen op het automatisch aansturen.

In dit hoofdstuk worden de realisaties van het project toegelicht, alsook verwijzingen gemaakt naar de relevante bestanden in de bijlagen. Vele van deze bijlagen zijn verzamelingen van de geschreven code uit Git repositories. Aangezien deze informatie minder relevant is voor de klant, wordt de precieze werking van deze code niet beschreven in dit document. Om meer informatie over deze code te verkrijgen, verwijzen wij u door naar het README.md¹¹ bestand in elke gezipte repository, alsook de commentaar in de bestanden zelf (inline¹²).

Webapp

Het doel van de webapp is het volgen van de automatische bemaling, wat leidt tot een beter inzicht in de status van de werf en dus een meer efficiënte bemaling. Het is ook mogelijk om de pompen op de werf op afstand aan te sturen. Zo kan men dringende interventies vermijden, waar iemand ter plaatse zou moeten komen.

Voor meer informatie over het (lokaal) hosten van deze webapp, raadpleeg het document README.md in de gezipte map “**Rising-IT – Repository – Frontend.zip**” onder de folder “**Bijlagen**”.

Specificaties

In dit onderdeel worden de specificaties van de webapp, diens werking en mogelijke use cases beschreven. Het doel van dit onderdeel is het verduidelijken van de technologieën, functionaliteiten en hoe deze samenwerken om de doelen van de applicatie te realiseren.

De webapp is een *Angular*-applicatie. Angular is een front-end framework dat gebruik maakt van html, css en typescript. Het is open-source, met een focus op schaalvergroting, veiligheid en toegankelijkheid. De opmaak van de webpagina's is ontworpen met behulp van *Tailwind* css, een open-source framework gericht op het streamlinen van de lay-out.

De communicatie met de server gebeurt via HTTP-requests naar het API van de server. Meer informatie over deze API vindt u in het hoofdstuk “**Server**”.

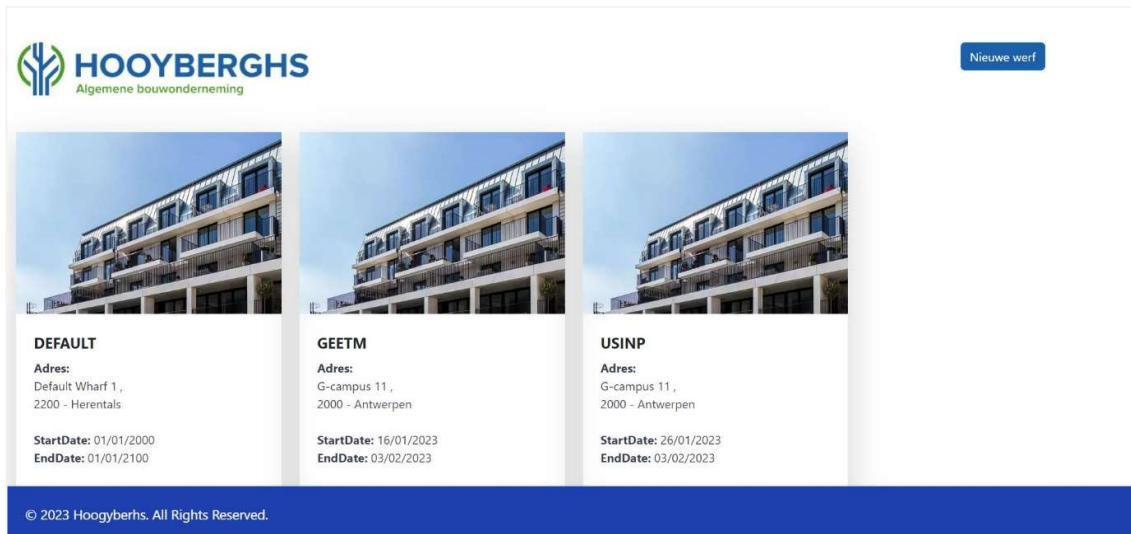
¹¹ <https://whatismarkdown.com/how-to-read-markdown-in-browser-2/>

¹² <https://devopedia.org/code-comments>

Overzicht

Het eindresultaat van de webapplicatie ziet er als volgt uit:

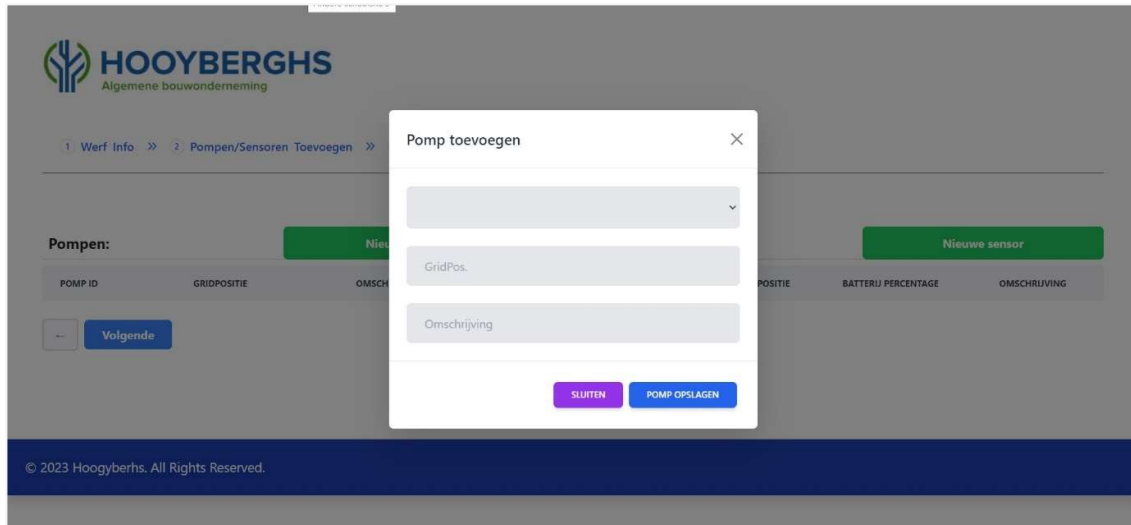
Bij het aanmelden krijgt de gebruiker een overzicht van alle werven in het systeem.



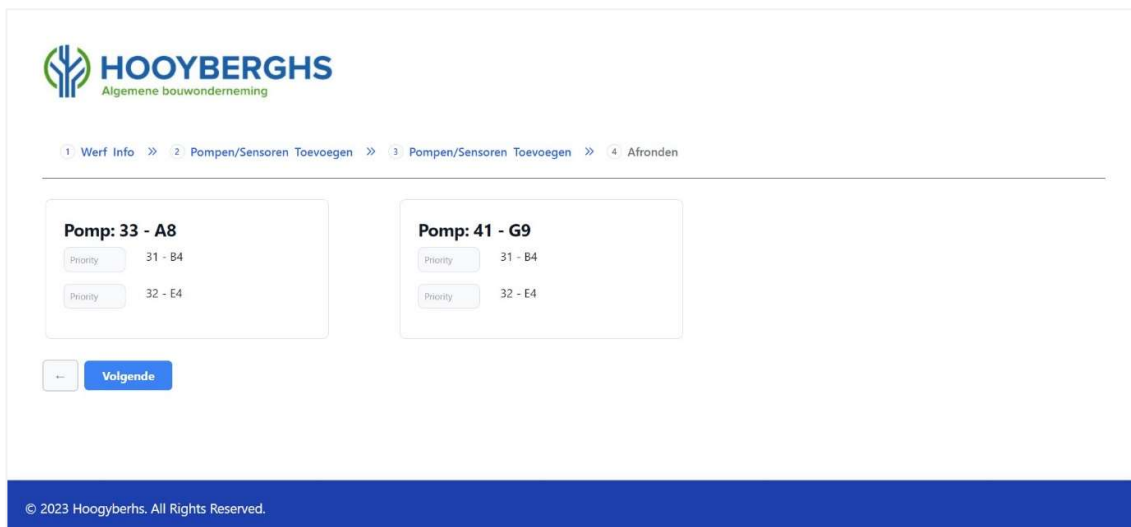
Bijlagen/Webapp/Overzicht werven.jpg

Deze heeft ook de optie om een nieuwe werf aan te maken.

Bijlagen/Webapp/Werf aanmaken (1).jpg

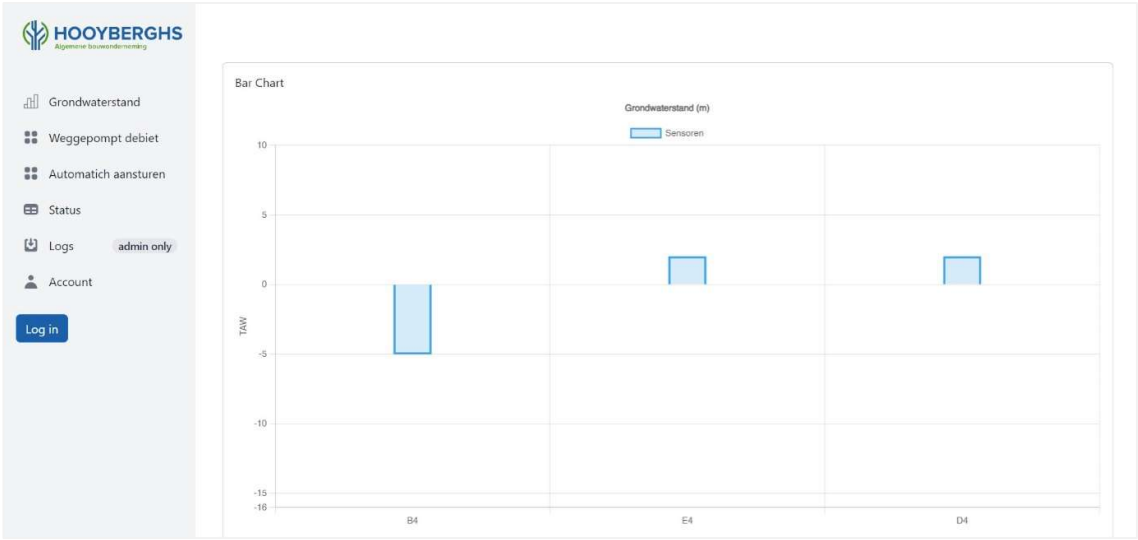


Bijlagen/Webapp/Werf aanmaken (3).jpg

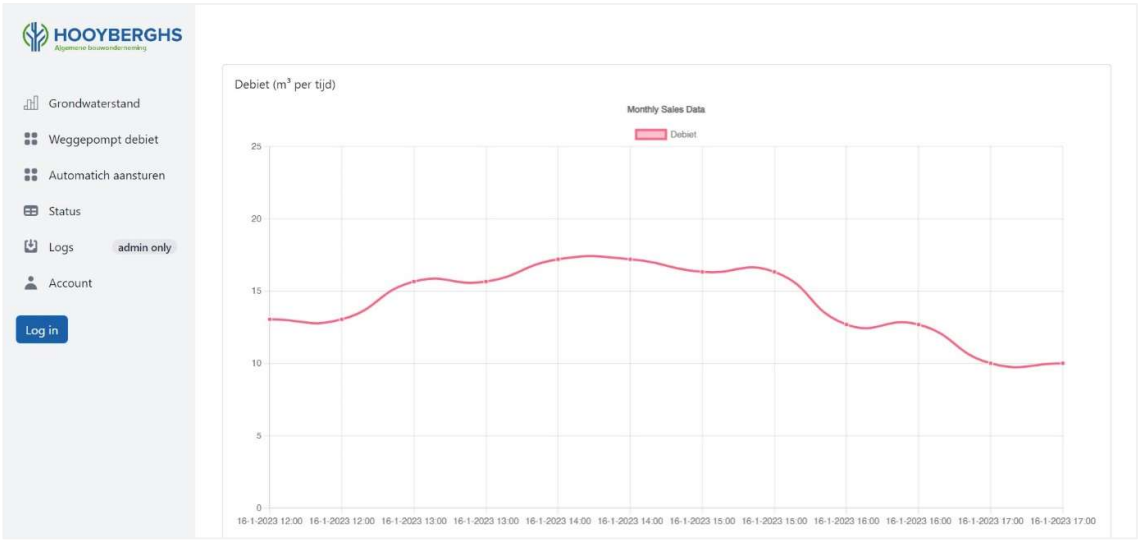


Bijlagen/Webapp/Werf aanmaken (5).jpg

Door een specifieke werf te selecteren, krijgt de gebruiker meer informatie over de grondwaterstand, het debiet, informatie over de pompen en sensoren, ...

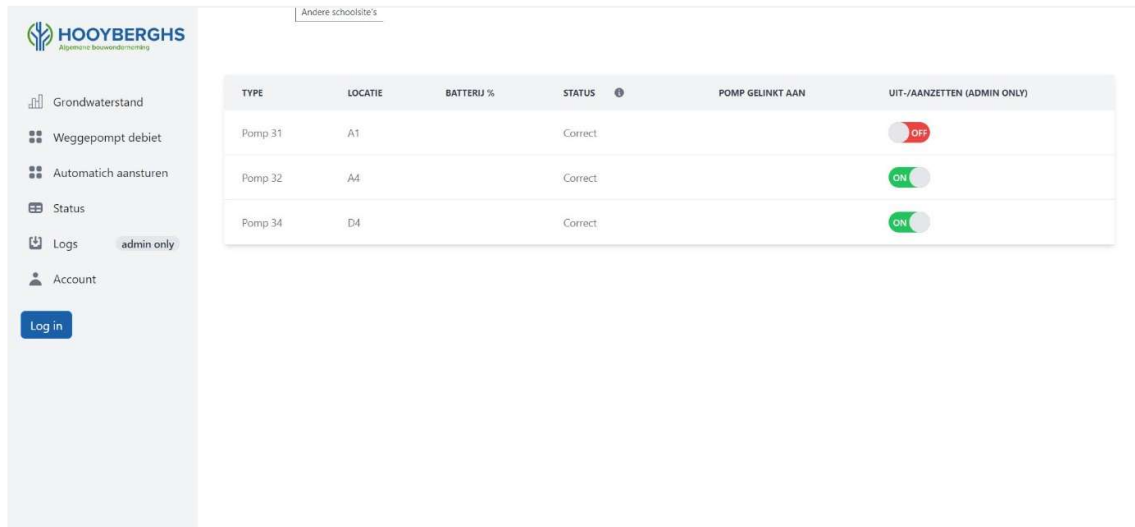


Bijlagen/Webapp/Werf - Grondwaterstand.jpg



Bijlagen/Webapp/Werf - Debiet.jpg

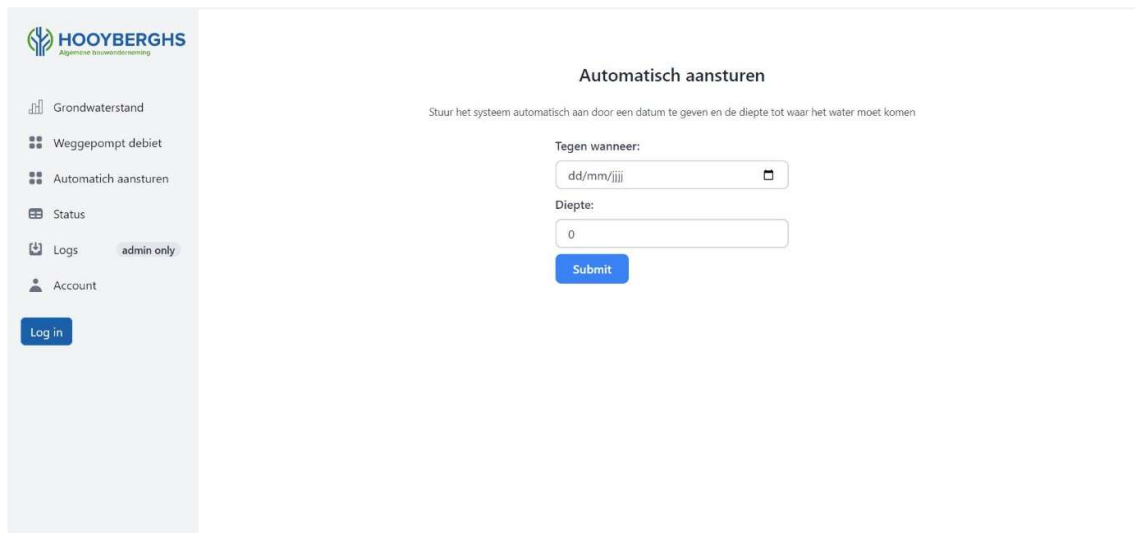
Ook kan de gebruiker hier pompen manueel en automatisch aansturen.



The screenshot shows the HOOYBERGHS web application interface. On the left is a sidebar with navigation links: Grondwaterstand, Weggepompt debiet, Automatisch aansturen, Status, Logs (with an 'admin only' badge), and Account. A 'Log in' button is at the bottom of the sidebar. The main content area has a header 'Andere schoolsite's' and a table with the following columns: TYPE, LOCATIE, BATTERIJ %, STATUS, POMP GELINKT AAN, and UIT-/AANZETTEN (ADMIN ONLY). The table contains three rows of pump data.

TYPE	LOCATIE	BATTERIJ %	STATUS	POMP GELINKT AAN	UIT-/AANZETTEN (ADMIN ONLY)
Pomp 31	A1		Correct		<input type="checkbox"/>
Pomp 32	A4		Correct		<input checked="" type="checkbox"/>
Pomp 34	D4		Correct		<input checked="" type="checkbox"/>

Bijlagen/Webapp/Werf – Aansturen – manueel.jpg



The screenshot shows the 'Automatisch aansturen' (Automatic control) form in the HOOYBERGHS web application. The sidebar is identical to the previous screenshot. The main content area has a title 'Automatisch aansturen' and a subtitle 'Stuur het systeem automatisch aan door een datum te geven en de diepte tot waar het water moet komen'. Below this are two input fields: 'Tegen wanneer:' with a date picker showing 'dd/mm/jjjj' and a calendar icon, and 'Diepte:' with a text input field containing '0'. A blue 'Submit' button is at the bottom.

Bijlagen/Webapp/Werf – Aansturen – automatisch.jpg

Infrastructuur

Om de server en database te hosten, is er gebruik gemaakt van Microsoft Azure. Hierbij is telkens gekozen om de instellingen zo duurzaam en goedkoop mogelijk te houden. Indien nodig, kan men dit upscalen met Azure-functies. Deze zijn echter niet inbegrepen in dit project.

De PowerShell-scripts in “**Rising-IT – Repository – Infrastructuur.zip**” in de folder “**Bijlagen/**” kunnen gebruikt worden om deze infrastructuur aan te maken. Meer informatie over het gebruik van deze scripts is terug te vinden in de README.md file van deze folder. Voor het uitvoeren van de scripts moet men in het bezit zijn van een Azure-subscriptie.

Met behulp van deze scripts, kan men de volgende Azure-resources aanmaken:

- Een resource group, die nodig is om de andere resources in te huisen
- Een SQL-server, waarin de SQL-database zich bevindt
- Firewall-regels voor deze SQL-server
- Een storage account, waarin de function app (zie hieronder) huist
- Een function app, die gebruikt wordt om de automatische aansturing van de pompen te regelen

In deze repository zijn ook SQL-instructies terug te vinden. Deze kunnen gebruikt worden om de (vooralsnog lege) database de structuur van het eerder getoonde schema te geven. Ook hierover staat meer uitleg in het README.md-bestand.

Server

De server is het centraal punt tussen de werf, database en website. Deze bevat een groot aantal HTTP-functies om data op te vragen, aan te maken, te bewerken en te verwijderen. Het API is geschreven in .NET, Microsofts gratis open-source software framework, en wordt gehost in een Azure app service. Deze app service kan men opstellen met behulp van Visual Studio¹³.

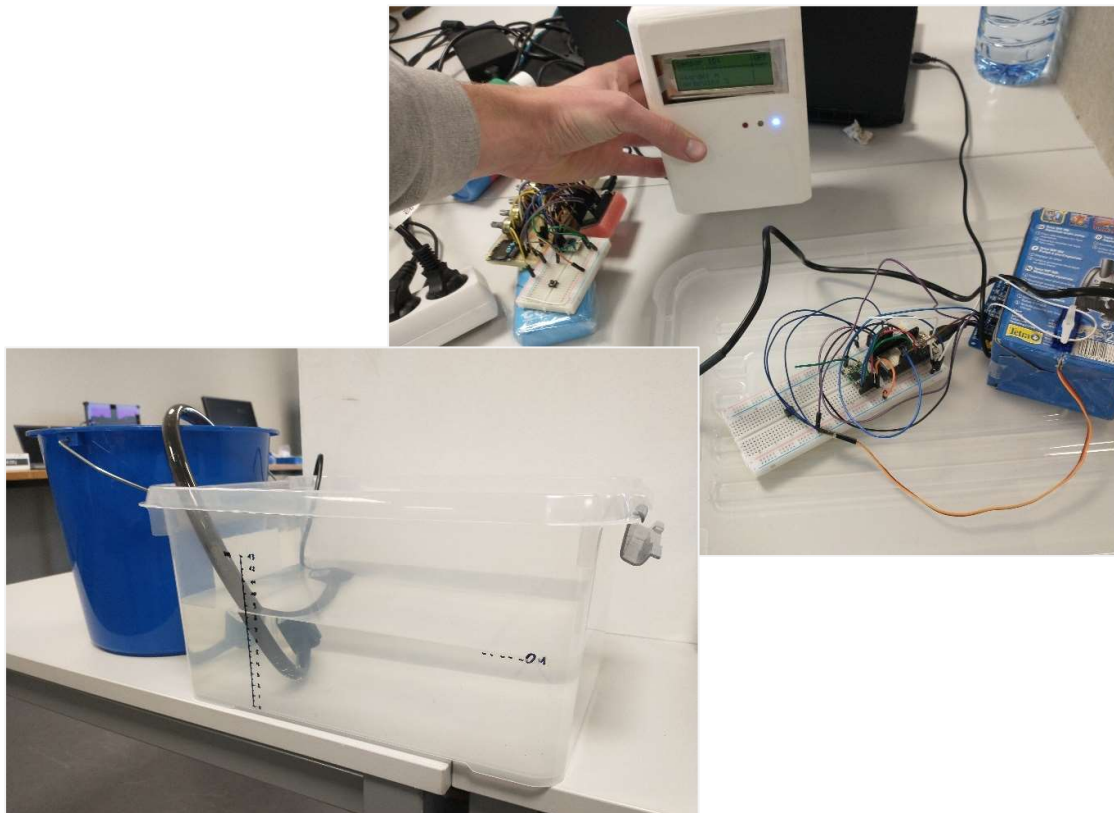
De code kan worden teruggevonden onder “**Bijlagen**”, in het bestand “**Rising-IT – Repository – Server.zip**”.

Om het automatisch pompen te faciliteren, is een Azure functionapp voorzien in de infrastructuur-scripts. Deze functie checkt periodiek of de waterstand de gewenste hoogte bereikt, en of deze moet worden bijgestuurd. De code van deze functie staat evenals onder “**Bijlagen**”, in het bestand “**Rising-IT – Repository – RepeatingFunction.zip**”.

Toonmodel

Tot slot is de werf zelf gesimuleerd met een toonmodel. Omdat de feedback van de pompen en sensoren via elektrisch signaal wordt doorgegeven, is deze gesimuleerd met een aantal potentiometers. Deze zijn aangesloten op een ESP32, een kleine computer, die via de LoRa-module (Long Range) het signaal doorstuurt naar de manager, ook een ESP32 met een LoRa-module. Deze verwerkt de signalen en communiceert met de hierboven genoemde server. Via HTTP-requests verstuurt de manager informatie en verkrijgt deze instructies.

De code van deze ESP32-modules is terug te vinden onder de folder “**Bijlagen/**” in het bestand “**Rising-IT – Repository – Toonmodel.zip**”.



¹³ <https://learn.microsoft.com/en-us/dotnet/core/tutorials/publishing-with-visual-studio?pivot=dotnet-7-0>

Temperatuurbestendigheid

Het warmteprobleem

De componenten in het omhulsel hebben een bepaalde minimum en maximum temperatuur tijdens gebruik welke beschreven zijn hieronder.

	Minimum	Maximum
ESP32	-40	80
RFM95	-20	70
Buck Converter	-40	85
batterij	-40	55

Er bevindt zich geen isolatie tussen het batterij compartiment en het elektronica compartiment waar het enkel relevant is om dit als het geheel te bezien.

De uiterste minimum en maximum temperaturen van het geheel worden genomen van de hoogste minimum en de laagste maximum temperatuur welke -20 graden Celsius en 55 graden Celsius bedraagt.

Mogelijke oplossingen

Als mogelijke oplossingen om de temperatuur in het omhulsel tussen -20 en 55 graden Celsius te houden zijn de volgende mogelijke oplossingen:

- Het omhulsel van goed warmtegeleidend materiaal voorzien met de top als uitzondering. Hierdoor wordt de koude van de grond gebruikt als koeling en geleid de bovenzijde minder goed warmte indien deze zich in het zonlicht bevindt.
- Het omhulsel plaatsen in de schaduw indien mogelijk.

Totale kost

De kost van dit prototype is relatief laag, aangezien vele onderdelen en services door de studenten (gratis) konden worden verkregen. Toch zijn er nog enkele noodzakelijke aankopen gebeurd. In dit hoofdstuk worden alle kosten overlopen, zodat de klant een goed beeld heeft van de mogelijke kost van het in productie nemen van dit systeem.

Prijs	Verklaring	Betaald door
€ 38.40	Aangekocht voor toonmodel: 3 LoRa-modules	Rising-IT
€ 33.00	Aangekocht voor toonmodel: Pomp, waterbak, buis	Rising-IT
€ 70.00	Maandelijkse kost van de Azure App Service, waaronder de server en de function app vallen	Student credit van Thomas More
€ 10.00	Maandelijkse kost van de Azure SQL Databases	Student credit van Thomas More
€ 0.00	Kost van het hosten van de webapp, aangezien deze lokaal is gehost. Afhankelijk van de vereisten van het programma, kan dit hosten tussen de € 10 en € 100 per maand kosten.	/

Conclusie

Met dit prototype tonen wij aan dat de bemaling sneller en efficiënter kan verlopen met behulp van automatisering. Men krijgt een beter overzicht van de gegevens op de werf, en kan gericht pompen. Doch zijn er nog enkele obstakels;

Het prototype is opgebouwd met simpele elektronica-componenten om de werking van de werf te *simuleren*. Er is dus geen test uitgevoerd met echte componenten op een echte werf. Daarom is het vooralsnog moeilijk om te voorspellen welke complicaties hier kunnen optreden.

Eén van de doelen, namelijk het preventief afsluiten van pompen, is niet bereikt met dit prototype. Door tijdsgebrek is gekozen om dit onderdeel uit te stellen en volledig de focus te leggen op het automatisch pompen.

Het algoritme om een “pompschema” op te stellen (om de gewenste diepte over de gewenste periode *gestaag* te bereiken) is nog vrij ruw en geeft niet altijd goede resultaten. Voor dit in gebruik kan worden genomen, is een grondige herwerking nodig om dit efficiënter en correcter te doen werken. Een overzicht van de werking van dit algoritme is terug te vinden in “Bijlagen/Plan/Backend/Process flow-User Input Algorithm.jpg”. De code is terug te vinden in “Bijlagen/Rising IT – Repository – Server.zip” en “Bijlagen/Rising IT – Repository – RepeatingFunction.zip”.

In de huidige implementatie geeft het automatisch pompen ook vaak een conflict met het manueel aan- of uitzetten van pompen. Er is namelijk geen mechanisme om te zorgen dat de manuele aanpassing niet wordt overschreven door de volgende automatische aanpassing, waardoor manueel aansturen dus niet nuttig is wanneer men al een input heeft gegeven voor de automatische aansturing.

Het toonmodel is niet erg robuust en is heel gevoelig. Kleine wijzigingen in positie kunnen ervoor zorgen dat de connectie onderbroken is of dat bepaalde componenten niet meer werken. Dit is grotendeels te wijten aan de goedkope onderdelen gebruikt voor dit prototype en kan vermeden worden door betere materialen te gebruiken.

Wij geloven dat al deze obstakels op te lossen zijn, mits voldoende tijd en geld hierin wordt geïnvesteerd. Dit zou echter buiten de scope van dit project vallen. Wij hopen dat dit prototype drempelverlagend kan werken om meer te investeren in het automatiseren van de bemaling. Ook geloven wij dat de werkwijze en code gebruikt kan worden door onze opvolgers als opstap bij het ontwerpen van de volgende versie van de automatische grondbemaling voor Hooyberghs NV.

Bijlagen

Alle code, afbeeldingen en overige relevante bestanden zijn terug te vinden in de folder **"Bijlagen\"**.