

2016•2017  
FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN  
*master in de industriële wetenschappen: energie*

## Masterproef

Automatisch uitlezen van vloeistofniveau bij sanitaire containers

Promotor :  
dr. ir. Johan BAETEN

Promotor :  
Dhr. PIETER-JAN DANIELS

Stan Antheunis , Robin Cardinaels

*Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: energie*

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven

2016•2017  
Faculteit Industriële  
ingenieurswetenschappen  
*master in de industriële wetenschappen: energie*

## Masterproef

Automatisch uitlezen van vloeistofniveau bij sanitaire  
containers

Promotor :  
dr. ir. Johan BAETEN

Promotor :  
Dhr. PIETER-JAN DANIELS

Stan Antheunis , Robin Cardinaels

*Scriptie ingediend tot het behalen van de graad van master in de industriële  
wetenschappen: energie*

## Woord vooraf

De masterproef vormt het sluitstuk van de opleiding tot master in de industriële wetenschappen energie focus automatisering.

In dit woord vooraf zouden wij graag even de tijd nemen om onze dank te richten aan alle personen die een bijdrage geleverd hebben in de goede afloop van deze masterproef.

Allereerst willen wij onze interne promotor Dr. Ir. Johan Baeten en externe promotor Pieter-Jan Daniels, bedanken voor hun uitstekende begeleiding tijdens onze masterproef.

Daarnaast willen wij ook ing. Luc Coenegracht bedanken. Met behulp van zijn inzicht en deskundige kennis van microcontrollers en databases hebben wij deze masterproef tot een goed einde kunnen brengen.

Ten slotte willen wij ook nog onze ouders bedanken om ons de opportuniteit te schenken om verder te studeren.

Veel leesplezier,  
Stan en Robin



<b>Inhoudsopgave</b>	
<b>Woord vooraf</b> .....	<b>1</b>
<b>Lijst van figuren</b> .....	<b>5</b>
<b>Abstract</b> .....	<b>7</b>
<b>Abstract in English</b> .....	<b>9</b>
<b>1 Inleiding</b> .....	<b>11</b>
1.1 Situering .....	11
1.2 Probleemstelling .....	11
1.3 Doelstelling .....	12
1.4 Materiaal en methode .....	12
1.4.1 Meting.....	12
1.4.2 Dataverwerking.....	13
1.4.3 De Human machine interface.....	13
<b>2 Literatuurstudie</b> .....	<b>15</b>
2.1 Inleiding .....	15
2.2 Meetmethode.....	15
2.2.1 Ultrasoon meting.....	15
2.2.2 Radarmeting .....	16
2.3 Communicatie en controller .....	17
2.3.1 Arduino .....	17
2.3.2 Dragonfly .....	18
2.4 WAMP .....	18
2.4.1 MYSQL-database .....	18
2.4.2 PHP-Website .....	19
2.5 Besluit.....	19
<b>3 Softwareontwikkeling</b> .....	<b>21</b>
3.1 Inleiding .....	21
3.2 Database ontwerp.....	22
3.3 Programmatie van de service .....	23
3.3.1 Communicatie.....	23
3.3.2 Databaseconnectie .....	25
3.4 Ontwikkeling van de website.....	27
3.4.1 Home .....	27
3.4.2 Container .....	27
3.4.3 Installatie .....	28
3.4.4 Klant .....	28

3.4.5	Log .....	28
3.4.6	Sensor .....	29
3.4.7	Verantwoordelijke .....	29
3.4.8	Toevoegen.....	30
3.4.9	Aanpassen.....	30
3.4.10	Codering van de website .....	31
3.5	Programmatie van de module .....	31
3.6	Besluit.....	32
<b>4</b>	<b>Hardware ontwerp.....</b>	<b>35</b>
4.1	Waterdichte behuizing .....	35
4.2	Connectors .....	35
4.3	Batterij .....	36
4.4	Besluit.....	36
<b>5</b>	<b>Besluit.....</b>	<b>37</b>
5.1	Overzicht .....	37
5.2	Bijdragen .....	37
5.3	Toekomstig werk.....	38
<b>Bijlage</b>	<b>.....</b>	<b>41</b>
	Mbed code .....	41
	Service code.....	47
	Websitecode.....	57
	Index.....	57
	Connection.....	57
	Routes .....	58
	Layout.....	59
	Begin pagina.....	59
	Container .....	60
	Nieuwe container .....	62
	Container aanpassen.....	64

## Lijst van figuren

Figuur 1: Gebruikte containers .....	11
Figuur 2: Systeem lediging.....	11
Figuur 3: Werking van de module.....	12
Figuur 4 Werkingsprincipe Ultrasoon sensor [13].....	15
Figuur 5: TDR-sensor [1].....	17
Figuur 6: Arduino 3G/GPRS shield [2].....	17
Figuur 7: Multitech dragonfly [3].....	18
Figuur 8: Principiële werking .....	21
Figuur 9: ER schema .....	22
Figuur 10: Voorbeeld: tabel aanmaken .....	22
Figuur 11: Voorbeeld: tabel aanmaken met phpMyAdmin .....	23
Figuur 12: Connectie van de service .....	23
Figuur 13: Basis authenticatie [4] .....	24
Figuur 14: HTTP-commando's .....	24
Figuur 15: Databaseconnectie visual studio .....	25
Figuur 16: Ophalen ID van de module.....	26
Figuur 17: Website pagina: beginscherm .....	27
Figuur 18: Website pagina: containers .....	27
Figuur 19: Website pagina: installatie.....	28
Figuur 20: Website pagina: klant.....	28
Figuur 21: Website pagina: log.....	28
Figuur 22: Website pagina: sensor.....	29
Figuur 23: Website pagina: verantwoordelijke.....	29
Figuur 24: Website pagina: klant toevoegen .....	30
Figuur 25: Website pagina: klant aanpassen.....	30
Figuur 26: Sequentie van de module.....	31
Figuur 27: Spelsberg waterdichte behuizing [15].....	35
Figuur 28: Weipu connector [5] .....	35





## Abstract

Deze masterproef wordt uitgevoerd bij industriële reinigingen DSSV te Kermt. Een van de activiteiten van DSSV is het plaatsen en ledigen van sanitaire containers op evenementen. Momenteel gebeurt de controle van het vulniveau visueel door een van de werknemers van DSSV. Het inschatten van het niveau is vaak een moeilijke opdracht, zeker bij de avond en/of nachtdienst. Ook kan het niveau op de spitsuren verbazend snel stijgen.

Het doel van de masterproef is een module te ontwikkelen die het niveau van de containers te allen tijde en overal raadpleegbaar maakt. Deze module moet ook een voorspelling maken op welk tijdstip de container vol is.

Het ontwikkelde prototype is een module die *stand alone* op elk type container geplaatst kan worden. De module meet de hoogte van het niveau in containers met een ultrasoon sensor en verzendt de meetdata via 3G naar een service. De service zorgt ervoor dat de ontvangen data wordt bijgehouden in een database. Een medewerker van DSSV kan via een webbrowser de data vanuit de database opvragen. Een achterliggende webservice zorgt voor een logische en gestructureerde weergave van de data.

Door gebruik te maken van een service, database en webservice is het eenvoudig om in een later stadium de toepassing verder uit te breiden naar meerdere modules. De module biedt nieuwe mogelijkheden voor het bedrijf. De module stelt ten allen tijde het niveau ter beschikking.. De lediging van de containers kan nu tijdsefficiënter. Hierdoor wordt de taak van de medewerkers verlicht.



## Abstract in English

This master thesis is accomplished at industrial cleanings, DSSV located in Kermt. One of the bigger branches in industrial cleaning is providing and emptying sanitary containers at events. Currently the fill level of every container will be manually checked by an employee of DSSV. This is a hard task, especially in the evening or night. The fill level in the containers can rise surprisingly quick.

The goal of this master thesis is to develop a prototype module that measures the fill level of the sanitary container. The fill level must be visible always and everywhere. The module must be capable to predict the moment when the container will be completely full.

The developed prototype is a *stand alone* module which can be placed on every type of container. The module measures the height of the fill level in a container with an ultrasonic sensor. The measured data will be send over 3G to a service that saves the data in a database. A DSSV employee can check the data with a simple webbrowser, collecting the data from the database. The underlying webservice provides a logical and structured data interface.

The use of a service, database and website allows an easy expansion of the application to a bigger amount of modules. The module offers new possibilities for the company. By using the module the fill level of the container will be known constantly. This will enlighten the task of the employees. Also the emptying of the containers can be handled more time efficient.



# 1 Inleiding

## 1.1 Situering

Deze masterproef wordt uitgevoerd bij DSSV te Kermt. DSSV is een familiebedrijf dat beschikt over 3 verschillende specialisaties. Deze specialisaties zijn bouw en renovatie van woningen, het transport van goederen en tot slot industriële reinigingen. Onze masterproef staat in functie van de laatste specialisatie. Eén van de grotere takken binnen de industriële reinigingswerken is het plaatsen en ledigen van sanitaire containers op grote evenementen zoals festivals. Figuur 1 geeft de gebruikte containers weer.



Figuur 1: Gebruikte containers

Bij grote evenementen zijn er verscheidene plaatsen, verspreid over het hele terrein, waar de containers geplaatst worden. De containers zijn aangesloten op de dichtstbijzijnde toiletgroep. Als de container vol is tankt een vrachtwagen de vloeistof over zoals figuur 2 schematisch weergeeft.



Figuur 2: Systeem lediging

## 1.2 Probleemstelling

Momenteel gebeurt de controle van het vulniveau manueel, dit wil zeggen dat één van de werknemers van DSSV constant op elke locatie gaat kijken. De snelheid waarmee de containers gevuld worden is sterk afhankelijk van het tijdstip en de plaats ervan. Hierdoor is het voor de werknemer moeilijk om een goede inschatting te maken van de vulsnelheid en het actuele niveau. Op de drukste momenten kan een container van 80 m<sup>3</sup> namelijk op 1,5 uur al vol zijn. De containers staan daarnaast ook nog eens verspreid over het hele festivalterrein wat mogelijk een extra obstakel is voor de werknemers om tijdig bij de containers te arriveren om deze te ledigen.

Een bijkomend probleem is het ontstaan van schuim aan de oppervlakte ten gevolge van de chemische reactie bij het rondpompen van het medium. Het schuim kan een verkeerde visuele inschatting van het niveau veroorzaken.

De containers worden op verschillende soorten van evenementen gebruikt. Op de meeste evenementen is er geen (elektrische) voeding voorzien.

### 1.3 Doelstelling

Om bovenstaande problemen te voorkomen of op te lossen moet er een mobiele meetopstelling ontwikkeld worden om het niveau uit te lezen en weer te geven. Deze module moet volgens de opdrachtgever aan verschillende eisen voldoen.

Het niveau van de containers moet ten allen tijde en overal raadpleegbaar zijn. Ook een voorspelling wanneer de container vol is moet weergegeven worden. Tevens moet de werkgever het volume, de afmetingen, de plaats enzovoort kunnen aanpassen aan de hand van welke module op welke container staat. De werknemers daarentegen moeten de informatie enkel kunnen raadplegen. Daarnaast moet één of meerdere werknemers een extra persoonlijke melding ontvangen als een container dreigt over te lopen of als het vloeistofniveau te snel stijgt.

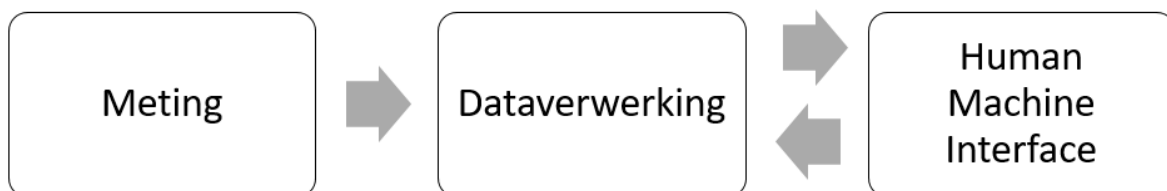
Tevens moet de module volledig stand alone kunnen werken. De aangewezen voedingsspanning is dan 24VDC. Maar in sommige gevallen zal er, door de organisatie van het evenement, een 230VAC voeding voorzien worden. Wanneer beide oplossingen ter beschikking zijn moet de verantwoordelijke een keuze kunnen maken tussen netspanning of stand alone.

Ook moet de meetopstelling makkelijk uit te wisselen zijn op verschillende containers met zowel verschillende formaten alsook verschillende vormen. In 99% van de gevallen staat de container buiten, in open lucht. De module, sensoren, en connectoren moeten dus minimaal spatwaterdicht zijn.

Tenslotte mag de maximale prijs om de te vervaardigen module slechts €1250 bedragen.

### 1.4 Materiaal en methode

Figuur 3 geeft een weergave van de belangrijkste processtappen voor het te ontwikkelen systeem. Een sensor meet het vulniveau van de container. De door de sensor verzamelde gegevens worden verwerkt door een controller en worden vervolgens zichtbaar gemaakt aan de gebruiker.



Figuur 3: Werking van de module

#### 1.4.1 Meting

De meting van het vloeistofniveau gebeurt met behulp van een sensor. Om het uitwisselen van de module eenvoudig en hygiënisch te laten verlopen is de keuze van de sensoren eerder beperkt. De sensoren mogen in geen geval contact maken met het medium. De keuze zal afhangen van enkele initiële testen. De resultaten van de testen bepalen welke sensor het meest geschikt is voor deze toepassing. Gezien schuimvorming een belangrijke factor is in de keuze van de sensor, is een ultrasoon sensor of een radar meetprincipe aangewezen. De meetsystemen hebben elk hun prijsklasse en specifieke voor- en nadelen.

### 1.4.2 Dataverwerking

De door de sensor opgemeten waarde wordt door een microcontroller verwerkt en verzonden naar een database. Voor de keuze van de microcontroller zijn er verschillende mogelijkheden. In deze keuze speelt vooral de energiezuinigheid een grote rol. Daardoor wordt de keuze van een laptop met labview en/of een PLC geëlimineerd. De microcontroller zal dus van het type Arduino, Raspberry Pi, Mbed, Dragonfly enz. zijn. De Dragonfly heeft in de standaard uitvoering al een communicatiemodule die via 3G en sms communiceert. Bij de andere basissystemen moet er een extra module aangekocht worden om de datacommunicatie mogelijk te maken. Bij Mbed en Dragonfly zijn er allerhande voorgeprogrammeerde bibliotheken ter beschikking. Bij Raspberry Pi en Arduino zijn er dan weer meer voorbeeldprogramma's voorhanden.

### 1.4.3 De Human machine interface

Zodra de microcontroller de meetgegevens verwerkt heeft, moeten deze raadpleegbaar zijn voor zowel de werknemers alsook voor de werkgever. De werkgever moet gemachtigd zijn om systeemdata zoals het totale volume, soort container en de plaats van de container aan te passen. Het personeel daarentegen zal enkel het vulniveau en de containerinformatie kunnen raadplegen. Deze gegevens moeten ten alle tijden en van op afstand voorhanden zijn. Een simpel touchscreen voldoet dus niet aan de eisen van de opdrachtgever. Een mogelijke oplossing is gebruik te maken van een sms-centrale. De centrale zal bij belangrijke gebeurtenissen een sms verzenden naar het gekozen gsm-nummer. Een andere, meer uitbreidbare oplossing, is het gebruik van een webserver en een website. Via de website is het ook eenvoudiger om een onderscheid te maken tussen werkgevers en werknemers.





## 2 Literatuurstudie

### 2.1 Inleiding

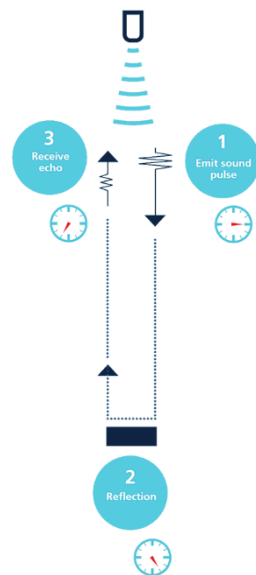
Een eerste en belangrijke stap is dat de ontwikkelaar inzicht krijgt in de methoden die voor handen zijn. Sectie 2.2 gaat dieper in op de verschillende meetmethoden die relevant zijn voor deze toepassing. De communicatie en de controller zijn de belangrijkste onderdelen van de te ontwikkelen module. Sectie 2.3 beschrijft daarom kort de werking van de gekozen communicatie en controller. Sectie 2.4 licht de interfacing tussen database en werknemer verder toe. WAMP vormt de basis voor het ontwerp van de database en de ontwikkeling van de webservice. De webservice is ontwikkeld om de gebruiker een eenvoudige bediening toe te staan. Sectie 2.5 geeft tot slot het besluit van de literatuurstudie. Hierin worden de belangrijkste aandachtspunten van hoofdstuk 2 herhaald.

### 2.2 Meetmethode

Gezien het ontstaan van schuim bij het rondpompen is de keuze in meetstelsel beperkt. De meetprincipes die in aanmerking komen voor de module zijn ultrasoon meting en radarmeting. Deze meetprincipes zullen verder uitvoerig worden besproken. De laatste paragraaf 2.2.4 besluit deze sectie met de voorkeur in meest geschikte sensor voor de module.

#### 2.2.1 Ultrasoon meting

Een ultrasone sensor zendt cyclusgewijs een korte, hoogfrequente geluidsimpuls (18 kHz tot 200 kHz) uit. Figuur 4 geeft de principiële werking van de ultrasone sensor weer.



*Figuur 4 Werkingsprincipe  
Ultrasoon sensor [13]*

De uitgezonden impuls plant zich voort met de geluidssnelheid in lucht. Wanneer de impuls op een ander medium dan lucht botst wordt het signaal weerkaatst. Het weerkaatste signaal zal de ultrasone sensor opnieuw bereiken. Het tijdsinterval tussen verzonden en ontvangen impuls wordt de looptijd genoemd en is een maat voor de afstand tot het medium. Een eerste (mogelijk) nadeel van de ultrasoon sensor is dat de meting temperatuurafhankelijk is. Aangezien de snelheid van het geluid in lucht verandert met de temperatuur. De ultrasoon sensor zal dus een

kleine afwijking geven bij extreme temperaturen. Deze is in de toepassing echter te verwaarlozen.

Een bijkomend (mogelijk) nadeel van de ultrasoon sensor is dat de sensor altijd een 'dode zone' heeft. Dit wil zeggen dat de ultrasoon sensor de eerste x aantal centimeters vanaf de sensor geen (betrouwbare) meting weergeeft. Typisch voor industriële ultrasoon sensoren is de dode zone ongeveer 20cm. Dit vormt echter geen probleem bij toepassing op de containers aangezien het medium nooit in de dode zone komt. De overloop van de containers is voorzien op 30cm vanaf de bovenkant van de containers.

Een derde nadeel van de ultrasoon sensor is de mogelijkheid van schuimdetectie. De ultrasoon sensor zal in sommige gevallen het schuim detecteren en niet het effectieve vloeistofniveau. Dit zal, bij de sanitaire toepassing, niet het geval zijn. Dit zal enkel voorkomen wanneer de containers gebruikt worden bij vloeistoffen met extreem veel schuimvorming, bijvoorbeeld bij het opruimen van bluswater.

Ultrasone sensoren hebben als voordeel dat alle objecten die geluid weerkaatsen gedetecteerd kunnen worden, ongeacht hun kleur en belichting. Een ultrasone meting heeft dus geen extra belichting nodig in het donker.

Sommige systemen bieden ook een high performance ultrasoon sensor aan. Bij deze sensor is het mogelijk een keuze te maken in gereflecteerde signalen. Bijvoorbeeld bij schuimvorming zal het eerste gereflecteerde signaal (van het schuim) een kleinere amplitude hebben dan het gereflecteerde signaal van de het fluïdum. Er kan dan gekozen worden voor het eerste gereflecteerde signaal of het best gereflecteerde signaal (dat van het effectieve fluïdum).

### **2.2.2 Radarmeting**

Een radar sensor werkt gelijkaardig aan een ultrasone sensor. Met het grote verschil dat radarmeettechniek geen geluidstrillingen maar elektromagnetische golven gebruikt en een frequentie heeft in het gigahertzbereik. De looptijd wordt gemeten aan de hand van de uitgezonde en door het mediumoppervlak gereflecteerde signaal. De looptijd is bij radarmeting in tegenstelling tot ultrasoon meting niet afhankelijk van druk, temperatuur, enz.. De nauwkeurigheid van dit meetprincipe is groter dan bij ultrasoon meting.

Figuur 5 toont een tweede vaak voorkomend radarmeting. Dit meetprincipe is de geleide radar of TDR-sensor.



Figuur 5: TDR-sensor [1]

Het grote voordeel van TDR-sensoren is dat deze geheel ongevoelig zijn voor schuimvorming. Maar deze staan in contact met het medium. Dit is in de voorliggende toepassing niet toegestaan. Toch is een meting waarbij de sensor in contact staat met het medium de enige oplossing om een meting te bekomen waarbij een fout door schuimvorming geheel te voorkomen. Bijgevolg komt de radarmeting niet in aanmerking.

### 2.3 Communicatie en controller

Omdat de module *on site* komt te staan kan de communicatie tussen module en server niet via normale TCP-IP communicatie. Er moet dus een oplossing gezocht worden waarmee de module draadloos kan communiceren met de server. De module moet bovendien in staat zijn om een sms te verzenden bij belangrijke gebeurtenissen. Om dit te doen bestaan er al enkele basissystemen met elk hun eigen voor- en nadelen.

#### 2.3.1 Arduino

Arduino is een (groep van) microcontroller(s) met een opensource programmeerplatform gebaseerd op de ATmega328P. De basisuitvoering (arduino uno) heeft 14 in-/uitgangen waarvan er 6 als pwm uitgang gebruikt kunnen worden. Ook heeft de arduino uno 6 analoge ingangen. Sommige van deze ingangen kunnen gebruikt worden voor seriële communicatie. Hiermee is het eenvoudig om sensoren op groter afstanden aan te sluiten. Voor arduino is er een gsm-shield ter beschikking, weergegeven in figuur 6. Deze module verbindt de arduino met het internet.



Figuur 6: Arduino 3G/GPRS shield [2]

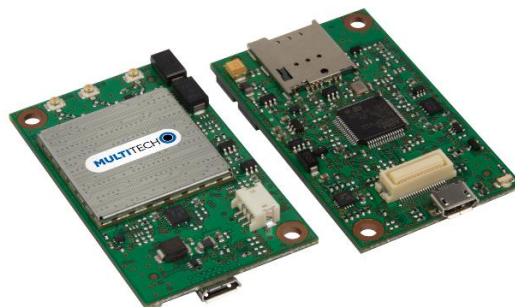
Door het gebruik van een arduino gsm-shield samen met de arduino worden de functionaliteiten van de arduino verruimd. Met deze module is de arduino in staat om verbinding te maken met het 3G netwerk alsook om sms berichten te verzenden.

Voordelen van de arduino met gsm-shield zijn dat deze eenvoudig te verkrijgen zijn, dat deze makkelijk programmeerbaar zijn met behulp van de arduino software en dat er veel voorbeeldprogramma's en tutorials beschikbaar zijn op internet.

Het nadeel van de arduino met gsm-shield is dat deze niet voorzien zijn om te gebruiken in een industriële toepassing. De connectoren voor arduino zijn breadboard connectoren. Hierdoor is de connectie niet optimaal.

### 2.3.2 Dragonfly

De Multitech Dragonfly, weergegeven op figuur 7, is een ARM gebaseerde microcontroller met geïntegreerde telit *cellular radio*. Door de geïntegreerde cellular radio kan de Dragonfly alle basis functionaliteiten van een gsm uitvoeren zonder extra hardware. Om stroomconsumptie te beperken is de cellular radio niet geconnecteerd totdat de verbinding is geïnitieerd is. De Dragonfly wordt met behulp van de online compiler van Mbed geprogrammeerd.



Figuur 7: Multitech dragonfly [3]

De Multitech Dragonfly heeft als voordeel dat de controller en de communicatiemodule in één behuizing vervat zijn. De Dragonfly is geproduceerd voor gebruik in industriële toepassingen. De in-/uitgangen worden door een 40pin board to board connector ter beschikking gesteld. Deze kunnen dus op een aparte printplaat op maat van de gebruiker geïmplementeerd worden.

## 2.4 WAMP

WAMP is een programma voor Windows om webapplicaties te maken. Dit programma ondersteunt Apache, MySQL database en PHP. Vandaar de afkorting WAMP: Windows, Apache, MySQL en PHP. Zo heeft WAMP ook een *localhost* ter realisatie van de website in combinatie met de database. Wamp is de uitgeleze keuze omdat dit de programmatie van een website en een database in één pakket omvat. Ook is dit pakket volledig gratis.

### 2.4.1 MYSQL-database

MySQL is de taal waarmee de database wordt geschreven. PhpMyAdmin is een onderdeel van het programma WAMP dat de MySQL ondersteunt. Het aanmaken van een database in phpMyAdmin vergemakkelijkt de connectie tussen de database en de website. Het programma is ook zeer gebruiksvriendelijk. Het toevoegen van tabellen en *records* kan op een gemakkelijke visuele manier gebeuren door een SQL code in te voegen. Het exporteren en importeren van databases is zeer gemakkelijk.

## 2.4.2 PHP-Website

De website is gemaakt met PHP. Dit is ook een onderdeel van het WAMP-programma. Door PHP-bestanden te maken via NetBeans of NotePad++ kan er een website geprogrammeerd worden. Elke pagina van de website komt overeen met een PHP bestand. Zo is het gemakkelijk om deze aan te passen of fouten op te sporen. Door de website op een localhost te draaien van WAMP kan de programmeercode getest worden. Het gebruik van PHP laat toe elementen uit de database aan te passen en of in te voegen. Ook kan het bestanden openen, lezen, schrijven, verwijderen, sluiten en nog veel meer. Omdat de taal zoveel mogelijkheden biedt is voor deze programmeertaal gekozen.

## 2.5 Besluit

Sectie 2.1 gaf een overzicht van de keuze in verschillende sensoren. Hierin werd duidelijk dat de radarsensoren een nauwkeuriger resultaat geven dan de ultrasoon sensor, zeker de TDR-sensor die de beste nauwkeurigheid heeft. Maar deze kan voor de toepassing niet gebruikt worden omdat voor de opdrachtgever de uitwisselbaarheid zeer belangrijk is. De TDR-sensor valt dus af in de keuze voor sensor. De ultrasoon sensor is minder nauwkeurig dan de radar maar beide voldoen aan de eisen van de opdrachtgever. De doorslaggevende factor in de keuze van de sensor is de kostprijs. Een radarsensor kost gemiddeld 3à4 keer meer dan een ultrasoon sensor. De keuze is daarom uitgegaan naar de ultrasoon sensor.

Sectie 2.3 besprak de manier van communiceren en de keuze van de controller. Zoals sectie 1.3 aangaf, was de keuze al snel gevallen op een microcontroller vanwege de energiezuinigheid. Maar ook binnen het gamma van microcontrollers is er nog een ruime keuze aan controllers die al dan niet een communicatiesysteem aan boord hebben. De multitech Dragonfly controller met ingebouwde cellular radio is uiteindelijk de gekozen controller in dit project.

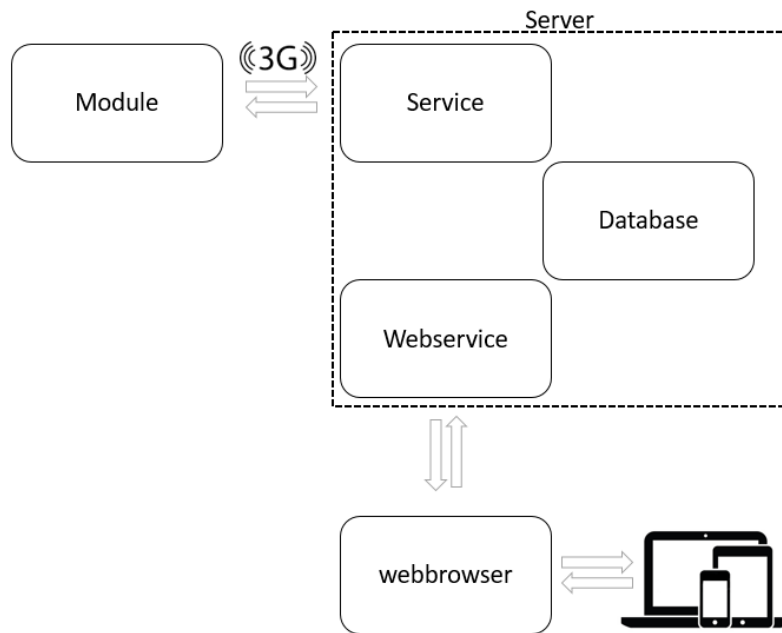
In tegenstelling tot de vorige secties gaf sectie 2.4 geen keuze weer maar een bespreking van het gebruikte programma om de database en de website te ontwikkelen. WAMP is een programma voor Windows om webapplicaties te maken. Dit programma ondersteunt Apache, MySQL database en PHP. Het was eerst en vooral zaak om notie te krijgen van de werking van de WAMP software en vooral het ontwerpen van de PHP website. Dit is in de opleiding tot industrieel ingenieur automatisering nooit aan bod gekomen in het lessenpakket.



### 3 Softwareontwikkeling

#### 3.1 Inleiding

Hoofdstuk 3 spitst zich vooral toe op het softwareontwerp van de module. Figuur 8 geeft de belangrijkste componenten van de softwareontwikkeling weer. Deze onderdelen komen terug in de secties van hoofdstuk 3.

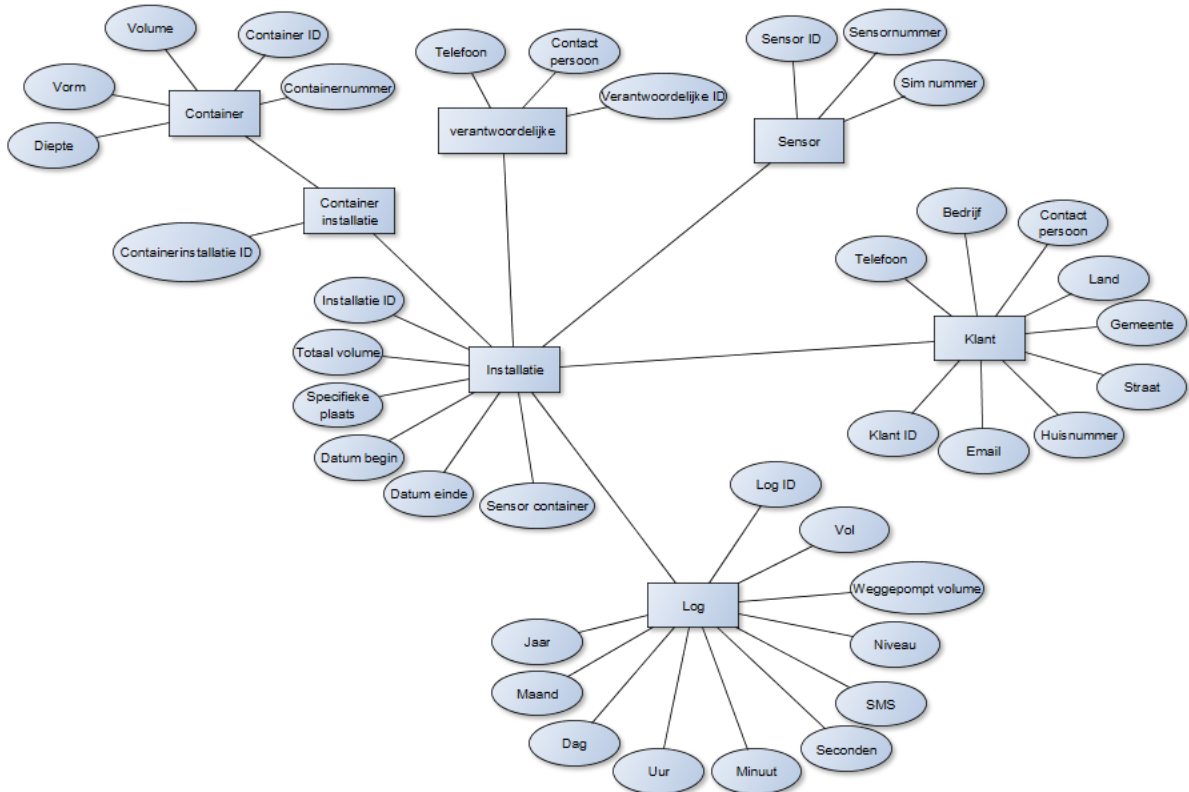


*Figuur 8: Principiële werking*

Sectie 3.2 gaat dieper in op het ontwerp en de structuur van de database. De achterliggende database vormt de basis van het geheel. Sectie 3.3 omvat de programmatie van de service. Dit is de belangrijkste component in de communicatie tussen module en database. De programmatie van de webservice wordt besproken in sectie 3.4. De webservice vormt de basis voor de visualisatie van de database. Voorgaande punten zijn enkel softwarematige oplossingen. Deze zullen uitgevoerd worden op een lokale server bij DSSV. De module daarentegen is een mobiele eenheid die op de verschillende containers geplaatst wordt. De programmatie van deze module wordt besproken in sectie 3.5. Tot slot bevat sectie 3.6 een besluit van de softwareprogrammatie waarin de belangrijkste aandachtspunten nog even terug aan bod komen.

### 3.2 Database ontwerp

Als eerste is er een ER schema ontworpen om alle onderdelen van de database in kaart te brengen. Dit geeft een mooie overzichtelijke basis. Figuur 9 geeft het ER schema weer.



Figuur 9: ER schema

Na het opstellen van het ER schema is dit geïmplementeerd in de database. Deze database is opgesteld met het programma phpMyAdmin dat ondersteunt wordt door WAMP. Het implementeren gaat als volgt. Er kunnen tabellen en *records* worden toegevoegd door een SQL-query in te voegen. Figuur 10 is een voorbeeld van hoe een query een tabel met topics kan maken.



Figuur 10: Voorbeeld: tabel aanmaken



Maar dit kan ook door het gebruiksvriendelijke programma, phpMyAdmin. De ingave van een tabelnaam, *records* en bijhorende opties resulteert in een nieuwe tabel met *records*. Figuur 11 is een voorbeeld van hoe er met phpMyAdmin een tabel kan aangemaakt worden.

Figuur 11: Voorbeeld: tabel aanmaken met phpMyAdmin

### 3.3 Programmatie van de service

De service is de achterliggende 'intelligentie' van de module. De module bevat geen intelligentie, deze vraagt/stuurt enkel sequentieel data van/naar de service. De serviceapplicatie is geschreven in visual studio en is geconnecteerd met de database. De service zorgt voor de dataverwerking en haalt/schrijft de data op de juiste plaats in de database.

#### 3.3.1 Communicatie

Figuur 12 geeft de code weer die ervoor zorgt dat de service te bereiken is vanaf een ander apparaat via wifi, 3g en dergelijke.

```
class Program
{
    static void Main(string[] args)
    {
        WebServiceHost host = new WebServiceHost(typeof(Service), new Uri("http://testdssv.hopto.org:8011"));
        ServiceEndpoint ep = host.AddServiceEndpoint(typeof(IService), new WebHttpBinding(), "");
        ServiceDebugBehavior sdb = host.Description.Behaviors.Find<ServiceDebugBehavior>();
        sdb.HttpHelpPageEnabled = false;

        host.Open();
        Console.WriteLine("Service is running");
        Console.WriteLine("Press enter to quit...");
        Console.ReadLine();
        host.Close();
    }
}
```

Figuur 12: Connectie van de service

De webservice zorgt ervoor dat de service vanaf een ander toestel bereikt kan worden. De host moet het IP-adres meekrijgen van de PC of server waarop de service draait alsook de TCP-poort waarover de communicatie plaatsvindt. Op figuur 12 is te zien dat de service opgeroepen wordt van op <http://testdssv.hopto.org> over poort 8011. Er is gekozen voor een virtueel vastgelegde hostname omdat verbindingen op IP-adres geen stabiele verbinding garandeert. Het publieke IP-adres verandert namelijk maandelijks.

Iedereen die het IP-adres of de hostname van de PC of server kent, kan nu data verzenden naar de service. De verbinding moet dus nog beveiligd worden. Figuur 13 stelt een gedeelte van het programma, wat de beveiliging van de service behandelt, voor.

```
private bool Authenticate(string realm)
{
    string[] credentials = GetCredentials(WebOperationContext.Current.IncomingRequest.Headers);

    if (credentials != null && credentials.Length == 2)
    {
        // do auth here
        var username = credentials[0];
        var password = credentials[1];
        if (username == "username" && password == "paswoord")
            // validate the username and password against whatever auth logic you have

            return true; // if successful
    }

    WebOperationContext.Current.OutgoingResponse.Headers["WWW-Authenticate"] = string.Format("Basic realm=\"{0}\"", realm);
    WebOperationContext.Current.OutgoingResponse.StatusCode = HttpStatusCode.Unauthorized;
    return false;
}
}
```

Figuur 13: Basis authenticatie [4]

Voor de beveiliging van de service is gekozen voor een basis authenticatie. Deze zal er voor zorgen dat enkel de toestellen met de gebruikersnaam en wachtwoord data kunnen verzenden of uitlezen. Wanneer dit bericht echter onderweg onderschept kan worden, heeft de persoon die dit kon onderscheppen, ook de gebruikersnaam en het wachtwoord.

Nu de service bereikt kan worden van op afstand moet er een manier van communicatie tussen de cliënt, de module en de server gedefinieerd worden. Het gebruikte communicatieprotocol is HTTP hyper tekst transfer protocol. De Dragonfly heeft een standaard bibliotheek die, mits aanpassingen en uitbreiding, gebruikt kan worden. Om dit te kunnen doen moeten de methodes gemaakt worden in de service. Figuur 14 geeft de HTTP-commando's weer.

```
[OperationContract]
[BasicAuthenticationInvoker]
[HttpGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetID/{SimNr}")]
string getID(string SimNr);

[OperationContract]
[BasicAuthenticationInvoker]
[HttpGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetTelNrVerantwoordelijke/{MyID}")]
string getTelefoonnummer(string MyID);

[OperationContract]
[BasicAuthenticationInvoker]
[HttpGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/PostNiveau/{id}/{meting}")]
void Logging(string id, string meting);

[OperationContract]
[BasicAuthenticationInvoker]
[HttpGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetSMSStatus/{MyID}")]
string getSMSStatus(string MyID);

[OperationContract]
[BasicAuthenticationInvoker]
[HttpGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetLocatie/{MyID}")]
string getLocatie(string MyID);
```

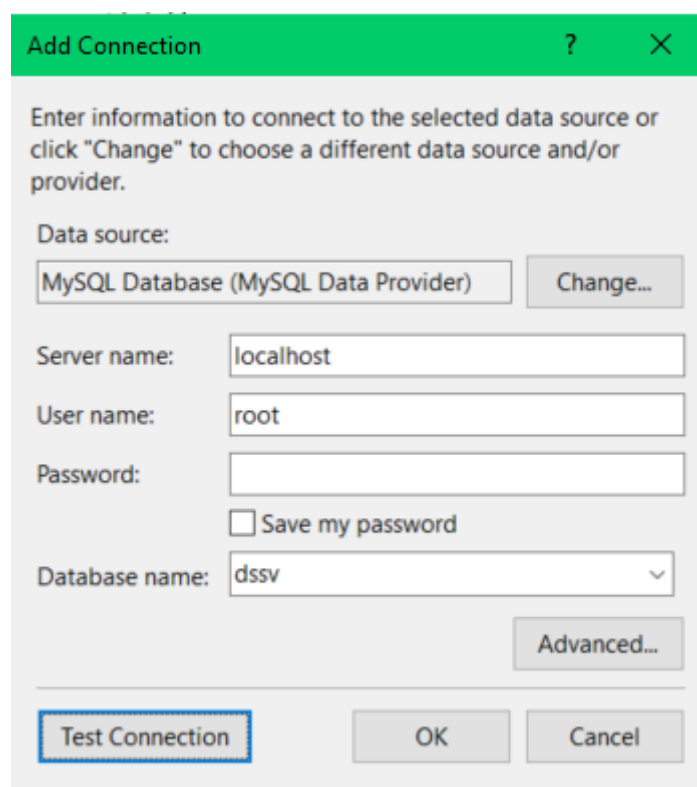
Figuur 14: HTTP-commando's

Het verzenden/ophalen van data kan nu op een eenvoudige manier. De get commando's krijgen hier altijd een variabele mee. Deze is gedefinieerd binnen de accolades. Bij elke operatie wordt

er een klasse uitgevoerd bijvoorbeeld `string getID(string TelNr)`: deze klasse verwacht het telefoonnummer te krijgen van de cliënt en geeft de ID van de module terug. Deze wordt dan via een `get` commando terug naar de cliënt gestuurd. In de klasse `getID` wordt de verbinding met de database tot stand gebracht. Dit wordt in sectie 3.2.2 databaseconnectie uitvoerig besproken. Voor elk connectie wordt de klasse “BasicAuthenticationInvoke” opgeroepen. Deze klasse gaat na of de aanvraag voor het `get` commando wel de bevoegdheid heeft om dit uit te voeren. Dit wil zeggen dat gebruikersnaam en wachtwoord overeen moeten komen.

### 3.3.2 Databaseconnectie

Telkens wanneer de module data verzendt of opvraagt moet deze informatie op een of andere manier met de database gekoppeld worden. Daarvoor moet de database aan het visual studio programma gekoppeld worden. Figuur 15 toont het tabblad om een databaseconnectie toe te voegen aan de applicatie.



The image shows a dialog box titled "Add Connection" with a green header bar. Below the header, there is a text box with the instruction: "Enter information to connect to the selected data source or click 'Change' to choose a different data source and/or provider." The dialog contains several input fields: "Data source:" with a dropdown menu showing "MySQL Database (MySQL Data Provider)" and a "Change..." button; "Server name:" with a text box containing "localhost"; "User name:" with a text box containing "root"; "Password:" with an empty text box and a checkbox labeled "Save my password"; and "Database name:" with a dropdown menu showing "dssv". At the bottom right, there is an "Advanced..." button. At the bottom, there are three buttons: "Test Connection" (highlighted with a blue border), "OK", and "Cancel".

Figuur 15: Databaseconnectie visual studio

De *datasource* is een MySQL database. Dit kan, na installatie van de MySQL *add-ons* voor visual studio geselecteerd worden in de lijst. De database kan lokaal op de pc/ server staan of op een externe server. In het geval van een externe databaseopslag dient bij *server name* de hostname of het IP-adres van de server ingegeven te worden. Bij lokale opslag is dit localhost. *User name* en wachtwoord kunnen worden ingesteld als extra beveiliging. Standaard voor MySQL is er geen wachtwoord en is de user name root.

Nu de database verbonden is met de applicatie kan de data in de desbetreffende tabellen geschreven worden met behulp van query's. Figuur 16 toont een stuk uit de code waarmee de ID van de module uit de database gehaald wordt aan de hand van de simkaartnummer die door de module meegegeven wordt.

```
public string getID(string SimNr)
{
    string Error = "Error";
    string MyID = "";
    MySqlCommand comm = new MySqlCommand();
    comm.Connection = new MySqlConnection(
        Properties.Settings.Default.ConnectionString);

    string mysql = @"SELECT Sensornummer from sensor where Simnummer=@SimNr";
    comm.CommandText = mysql;
    comm.CommandTimeout = 10000;

    comm.Parameters.Add("@SimNr", MySqlDbType.VarChar, 50);
    comm.Parameters["@SimNr"].Value = SimNr;
    comm.Parameters["@SimNr"].IsNullable = false;

    comm.Connection.Open();

    MySqlDataReader cursor = comm.ExecuteReader();
    while (cursor.Read())
    {
        MyID = cursor["Sensornummer"].ToString();
    }
    comm.Connection.Close();
    if (MyID != "")
        return MyID;
    else
        return Error;
}
```

*Figuur 16: Ophalen ID van de module*

Elke keer wanneer de er een commando wordt verzonden vanuit de module moet de databaseconnectie geopend worden en de data uitgelezen. In elke klasse wordt de database connectie dus geopend en ook weer afgesloten. De query die meegegeven wordt haalt de data uit de database en stopt alle verzamelde data in een cursor. In bovenstaand voorbeeld wordt er maar 1 record uit de database gehaald. Wanneer dit niet het geval is bevat de cursor een array met de data uit de database.

### 3.4 Ontwikkeling van de website

Als eerste is de *interface* opgesteld. Deze bestaat uit volgende *topics*: Home, Container, Installatie, Klant, Log, Sensor en Verantwoordelijke. Daarna werden alle *topics* uitgewerkt.

#### 3.4.1 Home

Figuur 17 toont de home-pagina van de webinterface. Deze interface is met opzet zeer eenvoudig gehouden. Zo is het bruikbaar voor elke werknemer.



Figuur 17: Website pagina: beginscherm

#### 3.4.2 Container

Figuur 17 geeft een voorbeeld van de container-pagina. Deze pagina geeft alle containers weer die in de database zitten. Onderaan deze pagina kunnen er container(s) worden toegevoegd en rechts achter elke *topic* kunnen de containerparameters worden aangepast.

<a href="#">Home</a>	<a href="#">Container</a>	<a href="#">Installatie</a>	<a href="#">Klant</a>	<a href="#">Log</a>	<a href="#">Sensor</a>	<a href="#">Verantwoordelijke</a>
Container ID	Container nummer	Vorm	Volume	Diepte		
1	12	Vierkant	20	5.5	<a href="#">Pas aan</a>	
3	10	Rond	5	3.5	<a href="#">Pas aan</a>	
7	200	Vierkant	40	5	<a href="#">Pas aan</a>	
8	25	Rond	25	2	<a href="#">Pas aan</a>	

[Voeg nieuwe container toe](#)

Figuur 18: Website pagina: containers

### 3.4.3 Installatie

Klikken op 'installatie' geeft toegang tot een overzicht van installatiegegevens. Hier wordt weergegeven welke klant de installatie in gebruik heeft, waar de installatie zich bevindt, welke containers de installatie bevat, het totale volume van de installatie, het nummer van de sensor die zich op de installatie bevindt, op welke container de sensor staat, wie de verantwoordelijke is voor de installatie en de begin- en einddatum van het in gebruik nemen van de installatie door de klant. Ook hier kunnen er installaties worden aangepast en toegevoegd. Figuur 19 toont het uitzicht van de pagina 'installatie'.

[Home Container Installatie Klant Log Sensor Verantwoordelijke](#)

Installatie ID	Klant	Plaats	Containernummer	Totaal volume	Sensor nummer	Sensor Container	Verantwoordelijke	Begin datum	Eind datum	
1	Castelli internationals NV	Pukkelpop	12,10	25	101	12	Pieter-Jan Daniels	2016-12-11	2016-12-12	<a href="#">Pas aan</a>
3	MetaTwist	Diepenbeek	25	25	35	25	Pieter-Jan Daniels	2016-12-07	2016-12-15	<a href="#">Pas aan</a>
33	Castelli internationals NV	Stevoort	200	40	69	200	Pieter-Jan Daniels	2017-01-30	2017-02-03	<a href="#">Pas aan</a>

[Voeg nieuwe installatie toe](#)

Figuur 19: Website pagina: installatie

### 3.4.4 Klant

Figuur 20 geeft een voorbeeld van de klant-pagina. Deze pagina stelt de gegevens van alle klanten ter beschikking, zoals het bedrijf van de klant, de contactpersoon van dat bedrijf, het telefoonnummer van de contactpersoon, het e-mailadres van de contactpersoon, de straat, het huisnummer, de gemeente en het land waar het bedrijf zich bevindt. Onderaan deze pagina kunnen er klanten worden toegevoegd en rechts achter elke *topic* kunnen de klantgegevens worden aangepast.

[Home Container Installatie Klant Log Sensor Verantwoordelijke](#)

Klant ID	Bedrijf	Contact persoon	Telefoon	Email	Straat	Huisnummer	Gemeente	Land	
1	Castelli internationals NV	Luca	042345678	Castelli@international.com	Heiligeboontjesstraat	102	Rome	Italie	<a href="#">Pas aan</a>
2	D.S.S.V. n.v.	Pieter-Jan Daniels	011313229	Pieter-jan@dssv.be	Lummense Kiezel	21	Kermt	Belgie	<a href="#">Pas aan</a>
3	Total	Hilde	0479543978	Hilde@total.be	Stevoortweg	78	Stevoort	Belgie	<a href="#">Pas aan</a>
4	MetaCoat	Frank	0479134976	Franky@meta.com	Europabaan	8	Lummen	Frankrijk	<a href="#">Pas aan</a>
11	MetaTwist	Stan	0478931764	Stan@metatwist.com	Industrielaan	118	Sint-Truiden	Belgie	<a href="#">Pas aan</a>

[Voeg nieuwe klant toe](#)

Figuur 20: Website pagina: klant

### 3.4.5 Log

Figuur 21 toont de log-pagina. De log-pagina geeft alle meetwaarden van de sensoren weer. Deze data geeft weer wanneer welke container een bepaald niveau heeft. Dit kan niet aangepast of toegevoegd worden omdat de sensor(s) op vaste tijdstippen zelf een nieuwe waarde toevoegt aan de database.

[Home Container Installatie Klant Log Sensor Verantwoordelijke](#)

Log ID	Timestamp	Niveau	Weggepomptvolume	Vol	SMS
1	2016-11-23 12:00:46	2	0	1	1

Figuur 21: Website pagina: log

### 3.4.6 Sensor

Figuur 22 geeft de sensor-pagina van de webinterface weer. Deze pagina geeft de gegevens van de sensoren met de bijhorende module weer. De voornaamste gegevens zijn het unieke nummer en het telefoonnummer van de desbetreffende module. Onderaan deze pagina kunnen er sensor(en) worden toegevoegd en rechts achter elke *topic* kunnen de gegevens worden aangepast.

[Home](#) [Container](#) [Installatie](#) [Klant](#) [Log](#) [Sensor](#) [Verantwoordelijke](#)

SensorID	Sensor nummer	Telefoon	
1	101	0479284657	<a href="#">Pas aan</a>
2	35	0478316497	<a href="#">Pas aan</a>
4	69	0479172834	<a href="#">Pas aan</a>
5	12	0475985432	<a href="#">Pas aan</a>

[Voeg nieuwe sensor toe](#)

Figuur 22: Website pagina: sensor

### 3.4.7 Verantwoordelijke

Figuur 23 toont een voorbeeld van de verantwoordelijke-pagina. Deze pagina stelt de gegevens van alle verantwoordelijken ter beschikking. Zo is er weergegeven wie de verantwoordelijke kan zijn en wat het telefoonnummer is van de desbetreffende verantwoordelijke. Verantwoordelijken toevoegen kan met de knop onderaan de pagina. Achter elke verantwoordelijke kunnen de gegevens worden aangepast.

[Home](#) [Container](#) [Installatie](#) [Klant](#) [Log](#) [Sensor](#) [Verantwoordelijke](#)

Verantwoordelijke ID	Contactpersoon	Telefoon	
1	Pieter-Jan Daniels	04798462	<a href="#">Pas aan</a>

[Voeg nieuwe verantwoordelijke toe](#)

Figuur 23: Website pagina: verantwoordelijke



### 3.4.8 Toevoegen

Bijna bij elke tabel kan een topic worden toegevoegd. Dit geldt voor volgende tabellen: container, installatie, klant, sensor en verantwoordelijke. Figuur 24 geeft een voorbeeld van de pagina van het toevoegen van een klant.

**Bedrijf: \***

**Contact persoon: \***

**Telefoon: \***

**Email: \***

**Straat: \***

**Huisnummer: \***

**Gemeente: \***

**Land: \***

\* required

Figuur 24: Website pagina: klant toevoegen

### 3.4.9 Aanpassen

Bijna bij elke tabel kan een topic worden aangepast. Dit geldt voor de tabellen container, installatie, klant, sensor en verantwoordelijke. Figuur 25 toont een voorbeeld van de aanpaspagina van een klant.

**ID: 2**

**Bedrijf: \***

**Contact persoon: \***

**Telefoon: \***

**Email: \***

**Straat: \***

**Huisnummer: \***

**Gemeente: \***

**Land: \***

\* Required

Figuur 25: Website pagina: klant aanpassen



### 3.4.10 Codering van de website

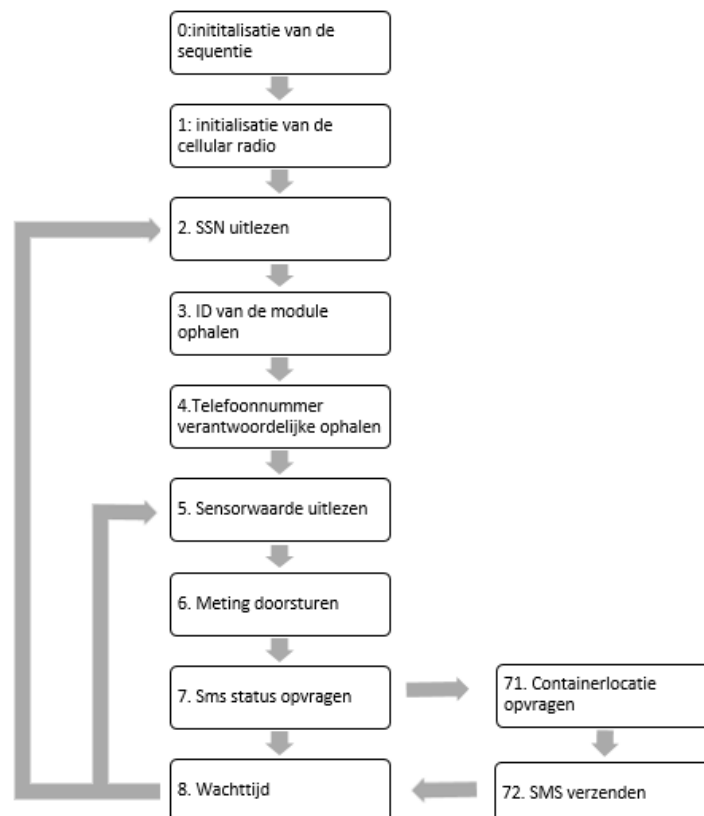
Zoals vermeld in sectie 2.4.2. is de website geschreven in PHP. Query's worden gebruikt voor het uitlezen, invoegen en aanpassen van records en tabellen van de database vanuit de PHP code. Hier enkele voorbeelden van query's die gebruikt zijn in het programmeren van de website:

- "SELECT\* FROM `log`" is een query waar heel de tabel van log wordt getoond.
- "INSERT into container (Container, Vorm, Volume, Diepte) VALUES ('\$container', '\$vorm', '\$volume', '\$diepte');" is een query waar de records van een container wordt toegevoegd.
- "UPDATE klant SET Telefoon='\$telefoon', Bedrijf='\$bedrijf', Contactpersoon='\$contact', Email='\$email', Straat='\$straat', Huisnummer='\$huis', Gemeente='\$gemeente', Land='\$land' WHERE Klantid='\$id'" is een query waar de records van de tabel klant wordt aangepast.

Bij de topic websitecode onder bijlage bevindt zich de rest van de code. Deze code is overgenomen van de bestaande code maar aangepast en verder uitgewerkt.

### 3.5 Programmatie van de module

De module wordt geprogrammeerd in de online Mbed-compiler. De module wordt geprogrammeerd in C# programmeertaal. De module bevat een minimum aan intelligentie. De module zal sequentieel commando's uitvoeren. Figuur 26 geeft de sequentie weer.



Figuur 26: Sequentie van de module

In stap 0 van de sequentie gebeurt er niets. Deze stap dient enkel om de sequentie te kunnen resetten. In stap 1 wordt de verbinding met de cellular radio tot stand gebracht. Deze stap moet slechts 1x uitgevoerd worden. De cellular radio blijft geconnecteerd met het mobiele netwerk zolang de module onder spanning staat. In stap 2 wordt de serienummer van de simkaart, afgekort SSN, uitgelezen. De SSN is het unieke nummer dat te vinden is op de simkaart. Dit nummer is ook terug te vinden in de database. Aan de hand van dit nummer wordt via een HTTP commando de ID van de module opgehaald in stap 3. De ID wordt gebruikt om de module te identificeren in het geval dat er meerdere modules te gelijktijd data zouden verzenden. Nadat de ID gekend is wordt het telefoonnummer van de verantwoordelijke via een HTTP commando opgevraagd uit de database. Dit is het nummer dat in noodgevallen een sms krijgt vanuit de module. Vanaf stap 5 wordt pas de effectieve logging van het niveau gedaan. In stap 5 wordt de waarde van de sensor ingelezen en omgezet van spanning (0 tot 3 volt) in een hoogte in meter. De gemeten waarde wordt samen met de ID doorgestuurd via een HTTP commando naar de service die deze dan opslaat in de database als percentage. Wanneer de meting verzonden is zal in stap 7 opgevraagd worden of er een noodgeval is. Hier zijn 2 mogelijkheden: indien het antwoord van de service *true* is wordt de specifieke plaats en het niveau van de container opgevraagd in stap 71. Deze informatie wordt geconverteerd tot een string die in stap 72 als sms verzonden wordt naar de verantwoordelijke. Wanneer de sms verzonden is gaat de sequentie verder naar stap 8. In het andere geval: wanneer het antwoord van de service *false* is, worden stap 71 en 72 overgeslaan en gaat de sequentie meteen naar stap 8. In stap 8 wordt er een tijd van 3 minuten gewacht. Wanneer de wachttijd verstreken is zijn er ook hier 2 mogelijkheden: elke tiende iteratie wordt alle informatie opnieuw opgevraagd om zeker te zijn dat er geen wijzigingen gebeurd zijn in de hardware of database. Dit wil zeggen dat de sequentie terug naar stap 2 gaat. In alle andere gevallen wordt het programma vanaf de meting (stap 5) opnieuw doorlopen.

### 3.6 Besluit

De eerste opdracht in het programmeren van het geheel was het ontwerp van de database. De database is het hart van het gehele project. Deze bevat alle gegevens van elke module.

Vermits alle gegevens in de database zitten, bevat de module zelf geen intelligentie. Het grote voordeel hiervan is dat het programma van de module wanneer dit één maal geschreven is toepasbaar is op elke module. Indien er wijzigingen dienen te gebeuren kan dit door de database en de service aan te passen. Om wijzigingen aan te brengen moet er dus nooit iemand *on site* aanpassingen gaan doen. Dit gebeurt allemaal op de server bij DSSV. De service voorziet de modules van informatie zoals ID, telefoonnummer van de verantwoordelijke en dergelijke, aan de hand van de unieke simkaartnummer. De module stuurt dan op zijn beurt het gemeten niveau door samen met zijn ID. Op deze manier is elk bericht tussen een module en service uniek en kunnen meerder modules tegelijkertijd communiceren met de server.

De website wordt gebruikt om een eenvoudige interface te leveren tussen database en personeel. Het personeel kan de gegevens van elke installatie bekijken en weet dus op elk ogenblik het vulniveau van de containers in de installatie. Aan de hand van de vulsnelheid kan er een schatting gemaakt worden wanneer de container vol is. De tijdstippen dat de verschillende containers vol zijn kunnen dus worden ingeschat en aan de hand hiervan kan een planning gemaakt worden om de containers te ledigen.

De module zelf voorziet samen met de service ook nog een extra deel van de communicatie. Wanneer de container voor 80% gevuld is wordt er vanuit de service een commando doorgegeven naar de module om een bericht te verzenden naar de verantwoordelijke. Dit is een extra beveiliging voor wanneer het personeel het niveau van de container even uit het oog verloren is.



## 4 Hardware ontwerp

Hoofdstuk 4 richt zich tot het hardware-ontwerp van de mobiele unit. Sectie 4.1, 4.2 en 4.3 lichten de gekozen materialen verder toe. Sectie 4.4 vormt tot slot een besluit met de belangrijkste aandachtspunten van hoofdstuk 4.

### 4.1 Waterdichte behuizing

Er is gekozen voor een waterdichte behuizing van Spelsberg. De behuizingen van Spelsberg zijn gemaakt van van glasvezel versterkt polyester dat weersbestendig is. Ook hebben ze verscheidende standaardmaten in hun gamma. Een ander voordeel is dat de behuizing een beschermingsgraad heeft van IP 66. Wanneer de connectoren van een hogere beschermingsgraad zijn is de behuizing zelfs IP67. Dit betekent dat de inhoud van de behuizing beschermd is tegen stof en tijdelijke onderdompeling. Tevens zijn de dekselschroeven corrosievrij en niet-verliesbaar. Ook zijn er verschillende opties van de dekselschroefkoppen beschikbaar. Binnen in de behuizing zijn schroefdraadbussen beschikbaar in formaat M4 en M6.



*Figuur 27: Spelsberg waterdichte behuizing [15]*

### 4.2 Connectors

De gekozen connector is van het merk Weipu. Het model, SP13, heeft 5 aansluitingen waar een nominale stroom van 5A mag doorvloeien. Ook kan de connector een spanning aan van 180V AC. Deze keuze van connector is gemaakt omwille van zijn hoge beschermingsgraad, namelijk IP 68. Met deze beschermingsgraad is de connector stofvrij en waterdicht. Hierdoor blijft hij bruikbaar onder water.



*Figuur 28: Weipu connector [5]*

### **4.3 Batterij**

Bij de keuze van de batterij is voor een simpele loodaccumulator gekozen van 12V DC en 80Ah. Dit type batterijen is voor deze opstelling naar prijs/kwaliteit verhouding zeer geschikt. Het bedrijf is bovendien reeds in bezit van batterijladers voor loodaccumulators. Dit alles compenseert de volgende nadelen: de batterij is redelijk zwaar en is corrosiegevoelig (aan de positieve pool van de batterij).

### **4.4 Besluit**

De eerste opdracht van het verwezenlijken van de hardware was het kiezen van een robuuste en waterdichte behuizing. Daarom is een volledige glasvezel versterkt polyester behuizing gekozen.

De connector moest minstens dezelfde beschermingsgraad hebben als de behuizing. Daardoor is er voor een connector gekozen van IP 68.

Bij het kiezen van de batterij is voor de meest voor de hand liggende en reeds al beschikbare batterij gekozen. Doordat de batterij bevestigd is aan de container betekent dit, dat het nadeel dat de batterij zwaar is, deels geëlimineerd is.

De module heeft een beschermingsgraad van IP67. Bij het bepalen van de totale beschermingsgraad wordt er altijd gekeken naar het materiaal met de laagste beschermingsgraad. In dit geval is dat de waterdichte behuizing met een IP 67. Dit betekent dat de module stofvrij is en dompeldicht.

## 5 Besluit

### 5.1 Overzicht

Tijdens deze masterproef zijn er verschillende stappen gezet om een idee om te zetten in een bruikbare toepassing. Als eerste werd er een literatuurstudie uitgevoerd om met de nodige basiskennis over de producten aan de slag te kunnen gaan. Vervolgens werd er gewerkt aan de programmatie van de software. Hiervoor waren verschillende keuzes bepalend in het verloop van de masterproef. Tot slot werd er een waterdichte case gebouwd om alle componenten de nodige bescherming te bieden.

In de literatuurstudie werden belangrijkste componenten van de masterproef onderzocht. Zo werd er als eerste de meetmethodes die toepasbaar zijn op dit project besproken en de belangrijkste voor- en nadelen overlopen. Hieruit is het besluit gekomen dat de ultrasone sensor meer geschikt is dan een radar sensor. Daarnaast werd de communicatie tussen module en service besproken alsook de controle-eenheid van de module. Hieruit werd besloten dat de Dragonfly microcontroller de meest complete oplossing biedt voor het probleem. Tot slot werd WAMP in detail bekeken WAMP is de overkoepelende software om de MySQL-database en de PHP website te ontwerpen. Eerst als een lokale website via localhost, later bereikbaar van op de server van DSSV.

Vervolgens werd de programmatie van de software besproken. Dit is onderverdeeld in de verschillende programma's met elk hun eigen programmeertaal. Als eerste werd het ontwerp van de database besproken. De database is het belangrijkste onderdeel van het geheel. Hierin wordt alle informatie opgeslagen. Dit vormt de basis voor verdere programmatie van de verschillende componenten. Vanaf het ontwerp van de database finaal is konden de service en de webservice gelijktijdig geprogrammeerd worden. Vanuit de database wordt er verbinding gemaakt met de module. Dit gedeelte van het project wordt de service genoemd. Ook in de ander richting wordt er gecommuniceerd, niet enkel de service geeft informatie vanuit de database naar de module. De module geeft op zijn beurt ook de meetinformatie door.

De interfacing tussen database en gebruiker oftewel de webservice zorgt ervoor dat de werknemers van DSSV via een website de informatie kunnen uitlezen. Zij kunnen zo alle installaties controleren en een schatting maken wanneer de containers vol zijn. Hierdoor kan het ledigen van de containers makkelijker vooraf ingepland worden.

Tot slot werd de programmatie van de module en de functionaliteiten hiervan besproken. De module bevat geen enkele intelligentie. Deze gaat simpelweg de geprogrammeerde sequentie aflopen en de data opvragen of verzenden naar de service, waar de verwerking gebeurt.

Als laatste werd er stilgestaan bij het ontwerp en de ideeën van de hardware. De module moet makkelijk verplaatsbaar zijn, maar toch waterdicht. Daarom zijn alle, aan de omgeving blootgestelde, componenten van beschermingsgraad IP67 of hoger. Daarnaast werd er nog kort de keuze van de batterij besproken. Omdat de module *stand alone* moet kunnen werken wordt deze gevoed door een 12V DC lood accumulator, simpelweg omdat deze ter beschikking zijn bij DSSV.

### 5.2 Bijdragen

De module geeft nieuwe mogelijkheden voor het bedrijf. Wanneer deze toegepast worden op alle verhuurde installaties kan er bij DSSV, aan de hand van de inschatting van het tijdstip waarop de

containers vol zouden zijn, een planning gemaakt worden voor het ledigen van de containers. Het ledigen van de containers kan dus veel tijdsefficiënter gebeuren. Ook wordt de arbeid van de werknemers verlicht. Deze zijn geen uren meer in de weer met het controleren van het vloeistofniveau van de sanitaire containers die verspreid staan over het gehele festivalterrein. Het niveau kan nu simpelweg door elke werknemer afgelezen worden op een smartphone, tablet of laptop.

### **5.3 Toekomstig werk**

De opdracht van deze masterproef was het ontwerpen van een module om het niveau van de containers uit te lezen. Tijdens deze masterproef werd er een werkend prototype ontwikkeld. Dit kan op verscheidene aspecten verbeterd worden. Onderstaande opsomming geeft de belangrijkste verbeterpunten weer:

- beveiliging van de communicatie tussen service en module verbeteren
- beveiliging van de communicatie tussen database en webservice implementeren
- verfraaien van de webinterface
- meerdere modules implementeren

De beveiliging van de beide communicaties zijn momenteel minimaal, zoals besproken in hoofdstuk 3. Dit is zeker een aandachtspunt voor de toekomst.

In deze masterproef is er gekozen om een basic uitvoering te maken van de website. Deze kan in een verder stadium nog verfraaid worden. Eventueel kan de bediening gebruiksvriendelijker gemaakt worden door de ontwikkeling van een speciale smartphone applicatie.

Daarnaast is het ook nog mogelijk om de database te voorzien van een extra service. Die de hele facturering van de installaties bij de verscheidene klanten dient te vereenvoudigen. Het merendeel van de facturatie-informatie is immers al terug te vinden in de database (bijvoorbeeld klanten, aantal containers, weggepompt volume,...). De beoogde uitbreiding zou enkel een tabel met de prijzen per onderdeel vereisen.



## Literatuurlijst

- [1] VEGA Grieshaber KG, „VEGAFLEX 86,” VEGA, [Online]. Available: <https://www.vega.com/nl-NL/Producten/Product-catalog/Niveau/Geleide-radar/VEGAFLEX-86>. [Geopend 20 September 2016].
- [2] arduino.cc, „GSM Shield V2,” Arduino, [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoGSMShield>. [Geopend 21 September 2016].
- [3] multitech, „MultiConnect® Dragonfly™,” Multitech, [Online]. Available: <http://www.multitech.com/brands/multiconnect-dragonfly>. [Geopend 29 september 2016].
- [4] Dylan, „Stackoverflow.com,” stackoverflow, 10 januari 2013. [Online]. Available: <http://stackoverflow.com/questions/10799393/wcf-rest-basic-authentication-on-certain-methods>. [Geopend 24 oktober 2016].
- [5] conrad, „conrad,” weipu, [Online]. Available: <http://www.conrad.be/ce/nl/product/734570/IP68-connector-serie-SP13-Aantal-polen-5-Kabelsteker-5-A-SP1310-P-5-I-Weipu-1-stuks?promotionarea=01>. [Geopend 12 december 2016].
- [6] MBED, „MTS-Dragonfly,” MBED, [Online]. Available: <https://developer.mbed.org/platforms/MTS-Dragonfly/>. [Geopend oktober 2016].
- [7] cplusplus, „cplusplus.com,” cplusplus, [Online]. Available: <http://www.cplusplus.com/reference/string/string/erase/>. [Geopend 8 november 2016].
- [8] Cybermaxs, „stackoverflow,” stackoverflow, [Online]. Available: <http://stackoverflow.com/questions/17140259/how-to-call-self-host-wcf-service-from-asp-net-client/17142998>. [Geopend 18 oktober 2016].
- [9] killersites, „killersites forum,” killersites, [Online]. Available: <http://www.killersites.com/>. [Geopend 9 november 2016].
- [10] Codular, „Codular,” Codular, [Online]. Available: <http://codular.com/>. [Geopend 9 november 2016].
- [11] stackoverflow, „stackoverflow forum,” stackoverflow, [Online]. Available: <http://stackoverflow.com/>. [Geopend november 2016].
- [12] w3schools, „w3schools: tutorials, references and examples,” w3schools, [Online]. Available: <http://www.w3schools.com/>. [Geopend 9 november 2016].
- [13] Microsonic GmbH, „Support microsonic,” [Online]. Available: <http://www.microsonic.de/nl/support/ultrasone-technologie/werkingsprincip.htm>. [Geopend 20 September 2016].
- [14] Spelsberg, „spelsberg TK PS 1818,” Spelsberg, [Online]. Available: <https://www.spelsberg.nl/product/p/tk-ps-1818-11-m-lege-behuizing/an/10590701/cHash/b239e1b5835eb3e382e7bb366785b141/>. [Geopend 12 december 2016].
- [15] requiremind, „requiremind,” requiremind, [Online]. Available: <http://requiremind.com/>. [Geopend november 2016].



## Bijlage

### Mbed code

Hieronder bevindt zich het Mbed programma van de multitech Dragonfly controller. Hiervoor zijn er enkele voorbeeldprogramma's uit de Mbed bibliotheek gebruikt. [6] [7]

```
#include "mbed.h"
#include "mthsas.h"
#include <iostream>
#include <string>
#include <vector>
#include <iterator>
#include <HTTPTson.h>
#include <math.h>
// Initialisatie van Stapsequentie
int step=0;
int i= 0;

// Initialisatie van parameters
bool prevSend = false;
static std::string WebAdr = "http://testdssv.hopto.org:8011/";
std::string strID = "";
std::string strSSN = "";
std::string MyID = "";
std::string Tel = "";
std::string SMSStatus = "";
std::string bericht = "";
float Niveau = 0;
AnalogIn in(A0);

// initialisatie van de radio
// This line controls the regulator's battery charger.
// BC_NCE = 0 enables the battery charger
// BC_NCE = 1 disables the battery charger
DigitalOut bc_nce(PB_2);

bool init_mthsas();
char* httpResToStr(HTTPResult res);
// The MTSSerialFlowControl object represents the physical serial link between the processor and the cellular radio.
mthsas::MTSSerialFlowControl* io;
// The Cellular object represents the cellular radio.
mthsas::Cellular* radio;

// An APN is required for GSM radios.
// web.be voor mobile vikings
// internet.proximus.be voor proximus
static const char apn[] = "internet.proximus.be";

bool radio_ok = false;

int main() {
// step naar nul als er gereset wordt
step=0;
while (true)
{
/*
sequentie:
0: initialisatie van de sequentie
1: initialisatie van de cellular radio
2: SSN uitlezen
3: ID van de module ophalen (http)
4: telefoonnummer van de verantwoordelijke ophalen (http)
5: sensorwaarde uitlezen (analoog)
6: meting doorsturen (http)
7: sms status opvragen
true:
```

```

71: opvragen containerlocatie
72: sms verzenden
8: wachttijd

*/
switch (step)
{
case 0: //initialisatie van de sequentie
{
step=1;
}
break;

case 1: // initialisatie van de cellular radio
{
logInfo("Step 1: Initialisation");
// Disable the battery charger unless a battery is attached.
bc_nce = 1;
// Change the baud rate of the debug port from the default 9600 to 115200.
Serial debug(USBTX, USBRX);
debug.baud(115200);
//Sets the log level to INFO, higher log levels produce more log output.
//Possible levels: NONE, FATAL, ERROR, WARNING, INFO, DEBUG, TRACE
mts::MTSLog::setLogLevel(mts::MTSLog::INFO_LEVEL);
logInfo("initializing cellular radio");
radio_ok = init_mtsas();
if (!radio_ok)
{
while (true)
{
logError("failed to initialize cellular radio");
wait(1);
}
}
logInfo("setting APN");
if (radio->setApn(apn) != MTS_SUCCESS)
{
logError("failed to set APN to \"%s\"", apn);
// wanneer het instellen van de APN mislukt opnieuw proberen
step=0;
}
else
{
logInfo(" step = 2");
step=2;
}
}
break;

case 2 : // SSN uitlezen
{
logInfo("Step 2 : get SSN from simcard");
std::string responsSSN = radio->getSSN();
logInfo("responsSSN=\"%s\"",responsSSN.c_str());
logInfo("responsSSN.length=\"%i\"",responsSSN.size());

strSSN=responsSSN;
// uitfilteren van de data
std::cout << strSSN << '\n';

strSSN.erase (36,44);
std::cout << strSSN << '\n';

strSSN.erase (0,23);
std::cout << strSSN << '\n';

logInfo("string na erase=\"%s\"", strSSN.c_str());
}
}

```

```

if (!responsSSN.empty())
{
    logInfo(" make step = 3");
    step=3;
}
}
break;

case 3: //ID van de module ophalen (http)
{
    std::string getMyID= WebAdr;
    getMyID.append("getID/");
    getMyID.append(strSSN.c_str());

    logInfo("getString=\"%s", getMyID.c_str());
    logInfo("bringing up the link");
    if (!radio->connect())
    {
        logError("failed to bring up the link");
    }
    else
    {
        // HTTPClient object used for HTTP requests.
        HTTPClient http;
        // HTTP GET example.
        char http_rx_buf[1024];
        HTTPResult res;
        // IHTTPDataIn object - will contain data received from server.
        HTTPText http_rx(http_rx_buf, sizeof(http_rx_buf));
        http.basicAuth("username", "password"); // encoder https://www.base64encode.org/
        logInfo("ik kom in de else");
        // Make a HTTP GET request to http://adress

        res = http.get(getMyID.c_str(), &http_rx);

        if (res != HTTP_OK)
            logError("HTTP GET failed [%d][%s]", res, httpResToStr(res));
        else
            logInfo("HTTP GET succeeded [%d]\r\n%s", http.getHTTPResponseCode(), http_rx_buf);
            MyID=http_rx_buf;
            logInfo("myID=\"%s", MyID.c_str());
            step=4;
    }
}
break;

case 4: //telefoonnummer van de verantwoordelijke ophalen (http)
{
    strID=MyID;
    std::cout << strID << '\n';

    strID.erase(0,1);
    std::cout << strID << '\n';

    strID.erase ((strID.size()-1),(strID.size()));
    std::cout << strID << '\n';

    logInfo("string na erase=\"%s\"", strID.c_str());

    std::string getTel= WebAdr;
    getTel.append("GetTelNrVerantwoordelijke/");
    getTel.append(strID.c_str());
    logInfo("getString=\"%s", getTel.c_str());
    logInfo("bringing up the link");
    if (!radio->connect())

```

```

{
logError("failed to bring up the link");
}
else
{
// HTTPClient object used for HTTP requests.
HTTPClient http;
// HTTP GET example.
char http_rx_buf[1024];
HTTPResult res;

// IHTTPDataIn object - will contain data received from server.
HTTPText http_rx(http_rx_buf, sizeof(http_rx_buf));
logInfo("ik kom in de else");
// Make a HTTP GET request to http:// final adress ???
res = http.get(getTel.c_str(), &http_rx);

if (res != HTTP_OK)
logError("HTTP GET failed [%d][%s]", res, httpResToStr(res));
else
logInfo("HTTP GET succeeded [%d]\r\n%s", http.getHTTPResponseCode(), http_rx_buf);
Tel=http_rx_buf;

logInfo("myID=\"%s", Tel.c_str());
step=5;
}
// logInfo("finished - bringing down link");

// radio->disconnect();

}
break;
case 5: //sensorwaarde uitlezen (analoog)
// lineariseren van 0-3 V tot een meting van 0-5 meter.
Niveau= 3.434457668*in.read()+0.195191759;
//afronden op 2 beduidende cijfers
Niveau= floor(100*Niveau)/100;

step=6;
break;
case 6://meting doorsturen (http)
{
char meting[20];
sprintf(meting,"%f",Niveau);
std::string postNiveau= WebAdr;
postNiveau.append("postNiveau/");
postNiveau.append(strID.c_str());
postNiveau.append(meting);

logInfo("postcommand = %s" , postNiveau);

if (!radio->connect())
{
logError("failed to bring up the link");
}
else
{
// HTTPClient object used for HTTP requests.
HTTPClient http;
// HTTP GET example.
char http_rx_buf[1024];
HTTPResult res;

// IHTTPDataIn object - will contain data received from server.
HTTPText http_rx(http_rx_buf, sizeof(http_rx_buf));
logInfo("ik kom in de else");
// Make a HTTP GET request to http://adress
res = http.get(postNiveau.c_str(), &http_rx);

```

```

        if (res != HTTP_OK)
            logError("HTTP GET failed [%d][%s]", res, httpResToStr(res));
        else
            logInfo("HTTP GET succeeded [%d]\r\n%s", http.getHTTPResponseCode(), http_rx_buf);
            step=7;
    }
}
// logging doorsturen
break;
case 7://sms status opvragen
{
    std::string getSMS= WebAdr;
    getSMS.append("GetSMSStatus/");
    getSMS.append(strID.c_str());

    logInfo("getString=\"%s", getSMS.c_str());
    logInfo("bringing up the link");
    if (!radio->connect())
    {
        logError("failed to bring up the link");
    }
    else
    {
        // HTTPClient object used for HTTP requests.
        HTTPClient http;
        // HTTP GET example.
        char http_rx_buf[1024];
        HTTPResult res;

        // IHTTPDataIn object - will contain data received from server.
        HTTPText http_rx(http_rx_buf, sizeof(http_rx_buf));
        res = http.get(getSMS.c_str(), &http_rx);
        if (res != HTTP_OK)
            logError("HTTP GET failed [%d][%s]", res, httpResToStr(res));
        else
            logInfo("HTTP GET succeeded [%d]\r\n%s", http.getHTTPResponseCode(), http_rx_buf);
            SMSStatus=http_rx_buf;

        std::string checkstring= "True" ;
        if (SMSStatus.find(checkstring)!= std::string::npos)
        {
            prevSend = false;
            logInfo( "SMS nog niet verzenden");
            step=8;
        }
        else
        {
            if (prevSend == false)
            {
                logInfo(" SMS verzenden");
                step=71;
                prevSend= true;
            }
            else
                step=8;
            logInfo("SMSStatus=\"%s", SMSStatus.c_str());
        }
    }

}

logInfo("finished - bringing down link");

radio->disconnect();
}
break;

case 71: // opvragen containerlocatie
{

```

```

std::string getLocatie= WebAdr;
getLocatie.append("getLocatie/");
getLocatie.append(strID.c_str());

logInfo("getString=\ "%s", getLocatie.c_str());
logInfo("bringing up the link");
if (!radio->connect())
{
logError("failed to bring up the link");
}
else
{

// HTTPClient object used for HTTP requests.
HTTPClient http;
// HTTP GET example.
char http_rx_buf[1024];
HTTPResult res;
// IHTTPDataIn object - will contain data received from server.
HTTPText http_rx(http_rx_buf, sizeof(http_rx_buf));
res = http.get(getLocatie.c_str(), &http_rx);
if (res != HTTP_OK)
logError("HTTP GET failed [%d]\ "%s)", res, httpResToStr(res));
else
logInfo("HTTP GET succeeded [%d]\r\n%s", http.getHTTPResponseCode(), http_rx_buf);
std::string Locatie=http_rx_buf;
bericht="container ";
bericht.append(Locatie);
bericht.append("is bijna vol");

step = 72;
}
}
break;

case 72: //sms verzenden
{
logInfo("sending SMS to %s", Tel.c_str());
mts::Cellular::Sms msg;
msg.phoneNumber = Tel;
msg.message = bericht;

if (radio->sendSMS(msg) != MTS_SUCCESS)
logError("sending SMS failed");
else
step = 8;
}
break;
case 8: //wachtijd
{
wait(1.80);
if (i<9)
{

step=5;
i++;
}
else
{
step= 2 ;
i=0;
}
}
break;
}
}
}

```



```

}
// initialisatie van de radio
bool init_mtsas() {
    io = new mts::MTSSerialFlowControl(RADIO_TX, RADIO_RX, RADIO_RTS, RADIO_CTS);
    if (! io)
        return false;

    // radio default baud rate is 115200
    io->baud(115200);
    radio = mts::CellularFactory::create(io);
    if (! radio)
        return false;

    // Transport must be set properly before any TCPSocketConnection or UDPSocket objects are created
    Transport::setTransport(radio);

    return true;
}
// http resultaat naar sting converteren.
char* httpResToStr(HTTPRequest res) {
    switch(res) {
        case HTTP_PROCESSING:
            return "HTTP_PROCESSING";
        case HTTP_PARSE:
            return "HTTP_PARSE";
        case HTTP_DNS:
            return "HTTP_DNS";
        case HTTP_PRTCL:
            return "HTTP_PRTCL";
        case HTTP_NOTFOUND:
            return "HTTP_NOTFOUND";
        case HTTP_REFUSED:
            return "HTTP_REFUSED";
        case HTTP_ERROR:
            return "HTTP_ERROR";
        case HTTP_TIMEOUT:
            return "HTTP_TIMEOUT";
        case HTTP_CONN:
            return "HTTP_CONN";
        case HTTP_CLOSED:
            return "HTTP_CLOSED";
        case HTTP_REDIRECT:
            return "HTTP_REDIRECT";
        case HTTP_OK:
            return "HTTP_OK";
        default:
            return "HTTP Result unknown";
    }
}
}

```

## Service code

Hieronder bevindt zich de code van het programma van de service, geschreven in visual studio. Hiervoor zijn er enkele stukken code van internet aangepast. [4], [8]

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.ServiceModel.Description;

namespace ConsoleRestSvc

```

```

{
class Program
{
static void Main(string[] args)
{
WebServiceHost host = new WebServiceHost(typeof(Service), new Uri("http://testdssv.hopto.org:8011"));
ServiceEndpoint ep = host.AddServiceEndpoint(typeof(IService), new WebHttpBinding(), "");
ServiceDebugBehavior sdb = host.Description.Behaviors.Find<ServiceDebugBehavior>();
sdb.HttpHelpPageEnabled = false;

host.Open();
Console.WriteLine("Service is running");
Console.WriteLine("Press enter to quit...");
Console.ReadLine();
host.Close();
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;
using System.ServiceModel.Web;
using System.Text;
using System.Threading.Tasks;

```

```

namespace ConsoleRestSvc
{
public class BasicAuthenticationInvoker : Attribute, IOperationBehavior, IOperationInvoker
{
#region Private Fields

private IOperationInvoker _invoker;

#endregion

#region IOperationBehavior Members

public void ApplyDispatchBehavior(OperationDescription operationDescription,
DispatchOperation dispatchOperation)
{
_invoker = dispatchOperation.Invoker;
dispatchOperation.Invoker = this;
}

public void ApplyClientBehavior(OperationDescription operationDescription,
ClientOperation clientOperation)
{
}

public void AddBindingParameters(OperationDescription operationDescription,
BindingParameterCollection bindingParameters)
{
}

public void Validate(OperationDescription operationDescription)
{
}

#endregion

#region IOperationInvoker Members

public object Invoke(object instance, object[] inputs, out object[] outputs)

```

```

{
    if (Authenticate("Client Name here"))
        return _invoker.Invoke(instance, inputs, out outputs);
    else
    {
        outputs = null;
        return null;
    }
}

public object[] AllocateInputs()
{
    return _invoker.AllocateInputs();
}

public IAsyncResult InvokeBegin(object instance, object[] inputs,
                                AsyncCallback callback, object state)
{
    throw new NotSupportedException();
}

public object InvokeEnd(object instance, out object[] outputs, IAsyncResult result)
{
    throw new NotSupportedException();
}

public bool IsSynchronous
{
    get
    {
        return true;
    }
}

#endregion

private bool Authenticate(string realm)
{
    string[] credentials = GetCredentials(WebOperationContext.Current.IncomingRequest.Headers);

    if (credentials != null && credentials.Length == 2)
    {
        // do auth here
        var username = credentials[0];
        var password = credentials[1];
        if (username == "username" && password == "password")
            // validate the username and password against whatever auth logic you have

            return true; // if successful
    }

    WebOperationContext.Current.OutgoingResponse.Headers["WWW-Authenticate"] = string.Format("Basic realm=\"{0}\"", realm);
    WebOperationContext.Current.OutgoingResponse.StatusCode = HttpStatusCode.Unauthorized;
    return false;
}

private string[] GetCredentials(WebHeaderCollection headers)
{
    string credentials = WebOperationContext.Current.IncomingRequest.Headers["Authorization"];
    if (credentials != null)
        credentials = credentials.Trim();

    if (!string.IsNullOrEmpty(credentials))
    {
        try
        {
            string[] credentialParts = credentials.Split(new[] { ' ' });

```

```

        if (credentialParts.Length == 2 && credentialParts[0].Equals("basic", StringComparison.OrdinalIgnoreCase))
        {
            credentials = Encoding.ASCII.GetString(Convert.FromBase64String(credentialParts[1]));
            credentialParts = credentials.Split(new[] { ':' });
            if (credentialParts.Length == 2)
                return credentialParts;
        }
    }
    catch
    {
    }
}

return null;
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.IO;

```

```

namespace ConsoleRestSvc
{

```

```

    [ServiceContract]
    public interface IService
    {
        // test echo's .
        [OperationContract]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/EchoWithGet/{s}")]
        string EchoWithGet(string s);

        [OperationContract]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/test")]
        string test();

        // data voor dragonfly

        [OperationContract]
        [BasicAuthenticationInvoker]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetID/{TelNr}")]
        string getID(string TelNr);

        [OperationContract]
        [BasicAuthenticationInvoker]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetTelNrVerantwoordelijke/{MyID}")]
        string getTelefoonnummer(string MyID);

        [OperationContract]
        [BasicAuthenticationInvoker]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/PostNiveau/{id}/{meting}")]
        void Logging(string id, string meting);

        [OperationContract]
        [BasicAuthenticationInvoker]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetSMSStatus/{MyID}")]
        string getSMSStatus(string MyID);

        [OperationContract]
        [BasicAuthenticationInvoker]
        [WebGet(ResponseFormat = WebMessageFormat.Json, UriTemplate = "/GetLocatie/{MyID}")]
        string getLocatie(string MyID);
    }

```

```

    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Data.Sql;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

using System.Data;
using System.IO;
using System.Runtime.Serialization.Json;
using System.Net.Http;
using System.Net;

namespace ConsoleRestSvc
{
    class Service : IService
    {
        public T Deserialize<T>(string json)
        {
            T obj = Activator.CreateInstance<T>();
            MemoryStream ms = new MemoryStream(Encoding.Unicode.GetBytes(json));

            DataContractJsonSerializer serializer = new DataContractJsonSerializer(obj.GetType());
            obj = (T)serializer.ReadObject(ms);
            ms.Close();
            return obj;
        }
        public string Serialize<T>(T obj)
        {
            DataContractJsonSerializer serializer = new DataContractJsonSerializer(obj.GetType());
            MemoryStream ms = new MemoryStream();
            serializer.WriteObject(ms, obj);
            string retVal = Encoding.UTF8.GetString(ms.ToArray());
            return retVal;
        }

        public string EchoWithGet(string s)
        {
            return "You said " + s;
        }
        public string test()
        {
            return "klommel";
        }

        public string getID(string TelNr)
        {
            string Error = "Error";
            string MyID = "";
            MySqlCommand comm = new MySqlCommand();
            comm.Connection = new MySqlConnection(
                Properties.Settings.Default.ConnectionString);

            string mysql = @"SELECT Sensornummer from sensor where Telefoon=@TelNr";
            comm.CommandText = mysql;
            comm.CommandTimeout = 10000;
        }
    }
}

```

```

comm.Parameters.Add("@TelNr", MySqlDbType.VarChar, 50);
comm.Parameters["@TelNr"].Value = TelNr;
comm.Parameters["@TelNr"].IsNullable = false;

comm.Connection.Open();

MySqlDataReader cursor = comm.ExecuteReader();
while (cursor.Read())
{
    MyID = cursor["Sensornummer"].ToString();
}
comm.Connection.Close();
if (MyID != "")
    return MyID;
else
    return Error;
}

public string getTelefoonnummer(string SensorNR)
{
    string Error = "Error";
    string TelNr = "";
    MySqlCommand comm = new MySqlCommand();
    comm.Connection = new MySqlConnection(
        Properties.Settings.Default.ConnectionString);

    string mysql = @"SELECT verantwoordelijke.Telefoon from verantwoordelijke INNER JOIN installatie ON
verantwoordelijke.verantwoordelijkheid= installatie.verantwoordelijkheid INNER JOIN Sensor on sensor.Sensorid = installatie.sensorid where
Sensornummer=@SensorNR";
    comm.CommandText = mysql;
    comm.CommandTimeout = 10000;

    comm.Parameters.Add("@SensorNR", MySqlDbType.VarChar, 50);
    comm.Parameters["@SensorNR"].Value = SensorNR;
    comm.Parameters["@SensorNR"].IsNullable = false;

    comm.Connection.Open();

    MySqlDataReader cursor = comm.ExecuteReader();
    while (cursor.Read())
    {
        TelNr = cursor["Telefoon"].ToString();
    }
    comm.Connection.Close();
    if (TelNr != "")
        return TelNr;
    else
        return Error;
}

public string getSMSStatus(string SensorNR)
{
    string Error = "Error";
    string TelNr = "";
    MySqlCommand comm = new MySqlCommand();
    comm.Connection = new MySqlConnection(
        Properties.Settings.Default.ConnectionString);

    string mysql = @"SELECT Log.SMS as SMSStatus from log INNER JOIN installatie ON log.installatieid= installatie.installatieid
inner join sensor on sensor.Sensorid=installatie.sensorid where sensor.Sensornummer=@SensorNR";
    comm.CommandText = mysql;
    comm.CommandTimeout = 10000;

    comm.Parameters.Add("@SensorNR", MySqlDbType.VarChar, 50);
    comm.Parameters["@SensorNR"].Value = SensorNR;
    comm.Parameters["@SensorNR"].IsNullable = false;

    comm.Connection.Open();

```

```

MySQLDataReader cursor = comm.ExecuteReader();
while (cursor.Read())
{
    TelNr = cursor["SMSStatus"].ToString();
}
comm.Connection.Close();
if (TelNr != "")
    return TelNr;
else
    return Error;
}

public void Logging(string id,string meting)
{
    int ID = int.Parse(id);
    double dMeting = double.Parse(meting);
    //StreamReader reader = new StreamReader(p);
    //string Result = reader.ReadToEnd();
    //reader.Close();
    //reader.Dispose();
    // deze data doorsturen: {"id":123,"niveau":805}

    MySQLCommand comm = new MySQLCommand();
    Container cont = getContainer(ID);
    Log prevLog = getPrevLog(cont.installatieID);
    Log logdata = new Log();
    logdata.installatieID = cont.installatieID;
    switch (cont.vorm)
    {
        case "Vierkant": // vierkant
            // geeft het huidige vulniveau in %
            logdata.niveau = Math.Round((((cont.hoogte - dMeting) / cont.hoogte) * 100), 2);
            break;
        case "Rond": // rond
            // berekening van het volume voor een ronde container (nog na te kijken)
            double d = cont.hoogte;
            double h = cont.hoogte-dMeting;
            double vulopp = (d * d * Math.Sin(2 * Math.Acos((2 * h - d) / d)) / 8 - (d * d * Math.Acos((2 * h - d) / d))) / 4 + d * d * Math.PI
/ 4;

            logdata.niveau = Math.Round(((vulopp / (Math.PI * cont.hoogte * cont.hoogte / 4)) * 100), 2);
            break;
    }
    // berekening van het weggepompte volume
    if (logdata.niveau < prevLog.niveau - 5)
        logdata.weggepompt = prevLog.weggepompt + (prevLog.niveau - logdata.niveau) * cont.TotaalVolume / 100;
    else
        logdata.weggepompt = prevLog.weggepompt;
    // checken of de container voor meer dan 85 % gevuld is
    if (logdata.niveau >= 85)
    {
        logdata.vol = true;
        logdata.SMSVerzonden = true;
    }
    else
        logdata.vol = prevLog.vol;
    Log sendLog = postNiveau(logdata);
}

public Log getPrevLog(int installatieID)
{
    MySQLCommand comm = new MySQLCommand();
    Log logData = new Log();
    comm.Connection = new MySqlConnection(
        Properties.Settings.Default.ConnectionString);
}

```

```

        string mysql = @"SELECT installatieid, Niveau, vol, Weggepomptvolume FROM `log` WHERE logid = (SELECT max(logid)
FROM `log` WHERE installatieid=@installatieID)";
        comm.CommandText = mysql;
        comm.CommandTimeout = 10000;

        comm.Parameters.Add("@installatieID", MySqlDbType.VarChar, 50);
        comm.Parameters["@installatieID"].Value = installatieID;
        comm.Parameters["@installatieID"].IsNullable = false;

        comm.Connection.Open();
        MySqlDataReader cursor = comm.ExecuteReader();
        while (cursor.Read())
        {
            logData.installatieID = (int)cursor["installatieid"];
            logData.niveau = (double)cursor["Niveau"];
            logData.vol = (bool)cursor["vol"];
            logData.weggepompt = (double)cursor["Weggepomptvolume"];
        }
        return logData;
    }
    public Container getContainer(int ID)
    {
        MySqlCommand comm = new MySqlCommand();
        Container cont = new Container();
        comm.Connection = new MySqlConnection(
            Properties.Settings.Default.ConnectionString);

        string mysql = @"SELECT container.ContainerID as containerid, container.Diepte as diepte, container.Vorm as vorm,
installatie.installatieid as instalID, installatie.TotaalVolume as totaalVolume FROM container inner join installatie on
container.ContainerID=installatie.sensorcontainer inner join sensor on sensor.Sensorid=installatie.sensorid WHERE
(sensor.Sensornummer=@SensorID)";
        comm.CommandText = mysql;
        comm.CommandTimeout = 10000;

        comm.Parameters.Add("@SensorID", MySqlDbType.VarChar, 50);
        comm.Parameters["@SensorID"].Value = ID;
        comm.Parameters["@SensorID"].IsNullable = false;

        comm.Connection.Open();

        MySqlDataReader cursor = comm.ExecuteReader();
        while (cursor.Read())
        {
            cont.id = (int)cursor["containerid"];
            cont.hoogte = (double)cursor["diepte"];
            cont.vorm = cursor["vorm"].ToString();
            cont.installatieID = (int)cursor["instalID"];
            cont.TotaalVolume = (double)cursor["totaalVolume"];
        }
        comm.Connection.Close();
        return cont;
    }

    public Log postNiveau(Log logdata)
    {
        int dummySMS = 0;
        int dummyVol = 0;
        MySqlCommand comm = new MySqlCommand();
        comm.Connection = new MySqlConnection(

```



```

Properties.Settings.Default.ConnectionString);
string mysql = "insert into log (Timestamp, installatieid ,SMS, Niveau, Weggepomptvolume, vol) Values (
CURRENT_TIMESTAMP, @MyID, @SMS, @niveau, @weg ,@Vol)";
comm.CommandText = mysql;
comm.CommandTimeout = 10000;

comm.Parameters.Add("@MyID", MySqlDbType.VarChar, 50);
comm.Parameters["@MyID"].Value = logdata.installatieID;
comm.Parameters["@MyID"].IsNullable = false;

if (logdata.SMSVerzonden == true)
    dummySMS = 1;

comm.Parameters.Add("@SMS", MySqlDbType.VarChar, 50);
comm.Parameters["@SMS"].Value = dummySMS;
comm.Parameters["@SMS"].IsNullable = false;

comm.Parameters.Add("@niveau", MySqlDbType.Double);
comm.Parameters["@niveau"].Value = logdata.niveau;
comm.Parameters["@niveau"].IsNullable = false;

comm.Parameters.Add("@weg", MySqlDbType.Double);
comm.Parameters["@weg"].Value = logdata.weggepompt;
comm.Parameters["@weg"].IsNullable = false;

if (logdata.vol == true)
    dummyVol = 1;

comm.Parameters.Add("@Vol", MySqlDbType.VarChar, 50);
comm.Parameters["@Vol"].Value = dummyVol; //dummyVol
comm.Parameters["@Vol"].IsNullable = false;

comm.Connection.Open();
try
{
    int result = comm.ExecuteNonQuery();
}
catch (Exception e)
{ }
comm.Connection.Close();

return logdata;
}
public string getLocatie(string MyID)
{
    string Error = "Error";
    string Locatie = "";
    string Bedrijf = "";
    string returnValue = "";
    MySqlCommand comm = new MySqlCommand();
    comm.Connection = new MySqlConnection(
    Properties.Settings.Default.ConnectionString);

    string mysql = @"SELECT installatie.SpecifiekePlaats as plaats, klant.Bedrijf as bedrijf from installatie INNER JOIN klant ON
installatie.klantid=klant.Klantid INNER JOIN Sensor on sensor.Sensorid = installatie.sensorid where Sensornummer=@SensorNR ";
    comm.CommandText = mysql;
    comm.CommandTimeout = 10000;

    comm.Parameters.Add("@SensorNR", MySqlDbType.VarChar, 50);
    comm.Parameters["@SensorNR"].Value = MyID;
    comm.Parameters["@SensorNR"].IsNullable = false;

    comm.Connection.Open();

    MySqlDataReader cursor = comm.ExecuteReader();

```

```

while (cursor.Read())
{
    Locatie = cursor["plaats"].ToString();
    Bedrijf = cursor["bedrijf"].ToString();
}
comm.Connection.Close();
if (Locatie != "")
{
    returnValue = string.Concat(Locatie, " bij ", Bedrijf);
    return returnValue;
}
else
    return Error;
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace ConsoleRestSvc
{

```

```

    public class RecLog
    {
        public int ID;
        public double meting;
    }
    public class Log
    {
        public int installatieID;
        public double niveau;
        public bool SMSVerzonden;
        public double weggepompt;
        public bool vol;
    }
    public class Container
    {
        public int id;
        public double hoogte;
        public string vorm;
        public int installatieID;
        public double TotaalVolume;
    }
}

```

## Websitecode

Hieronder bevindt zich het programma van de website. Hiervoor zijn er enkele code van het internet aangepast. [9], [10], [11], [12]

### Index

```
<?php
require_once('connection.php');

if (isset($_GET['controller']) && isset($_GET['action'])) {
    $controller = $_GET['controller'];
    $action = $_GET['action'];
} else {
    $controller = 'pages';
    $action = 'home';
}

require_once('views/layout.php');
?>
```

### Connection

```
<?php
class Db {
    private static $instance = NULL;

    private function __construct() {}

    private function __clone() {}

    public static function getInstance() {
        if (!isset(self::$instance)) {
            $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
            self::$instance = new PDO('mysql:host=localhost;dbname=dssv', 'root', '', $pdo_options);
        }
        return self::$instance;
    }
}
?>
```

## Routes

```
<?php
function call($controller, $action) {
    require_once('controllers/' . $controller . '_controller.php');

    switch($controller) {
        case 'pages':
            $controller = new PagesController();
            break;
        case 'klant':
            // we need the model to query the database later in the controller
            require_once('models/klant.php');
            $controller = new klantController();
            break;
    }

    $controller->{ $action }();
}

// we're adding an entry for the new controller and its actions
$controllers = array('pages' => ['home', 'klant', 'container',
'logging','installatie','sensor','verantwoordelijke', 'login', 'logout', 'error']);

if (array_key_exists($controller, $controllers)) {
    if (in_array($action, $controllers[$controller])) {
        call($controller, $action);
    } else {
        call('pages', 'error');
    }
} else {
    call('pages', 'error');
}
?>
```

## Layout

```
<DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <header>

    </header>

    <?php require_once('routes.php'); ?>

  <body>
</html>
```

## Begin pagina

```
<html>
  <head>
  </head>
  <body>
    <header>
      <a href='/dssv'><div>Home</div></a>
      <p> </p>
      <a href='?controller=pages&action=container'>Container</a>
      <p> </p>
      <a href='?controller=pages&action=installatie'>Installatie</a>
      <p> </p>
      <a href='?controller=pages&action=klant'>Klant</a>
      <p> </p>
      <a href='?controller=pages&action=logging'>Log</a>
      <p> </p>
      <a href='?controller=pages&action=sensor'>Sensor</a>
      <p> </p>
      <a href='?controller=pages&action=verantwoordelijke'>Verantwoordelijke</a>
      <p> </p>
    </header>

    <footer>

    <p> <font size="1">Copyright</font></p>
      <p> <font size="1">Stan Antheunis & Robin Cardinaels</font></p>

    </footer>
  <body>
</html>
```

## Container

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>D.S.S.V. nv</title>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<a href='/dssv'>Home</a>
```

```
<a href='/dssv/views/pages/view-container.php'>Container</a>
```

```
<a href='/dssv/views/pages/view-installatie.php'>Installatie</a>
```

```
<a href='/dssv/views/pages/view-klant.php'>Klant</a>
```

```
<a href='/dssv/views/pages/view-log.php'>Log</a>
```

```
<a href='/dssv/views/pages/view-sensor.php'>Sensor</a>
```

```
<a href='/dssv/views/pages/view-verantwoordelijke.php'>Verantwoordelijke</a>
```

```
</header>
```

```
<?php
```

```
/*
```

```
view-container.PHP
```

```
Displays all data from 'container' table
```

```
*/
```

```
// connect to the database
```

```
include('connect-db.php');
```

```
// get results from database
```

```
$container = <<<SQL
```

```
SELECT *
```

```
FROM `container`
```

```
SQL;
```

```
if (!$result = $db->query($container)) {
```

```
    die("There was an error running the query [' . $db->error . ']);
```

```
}
```

```

// display data in table

echo "<table border='1' cellpadding='10'>";

echo "<tr> <th>Container ID</th> <th>Container nummer</th> <th>Vorm</th>
<th>Volume</th> <th>Diepte</th> <th></th></tr>";

// loop through results of database query, displaying them in the table

while ($row = $result->fetch_assoc()) {

// echo out the contents of each row into a table

echo "<tr>";

echo '<td>' . $row['ContainerID'] . '</td>';

echo '<td>' . $row['ContainerNummer'] . '</td>';

echo '<td>' . $row['Vorm'] . '</td>';

echo '<td>' . $row['Volume'] . '</td>';

echo '<td>' . $row['Diepte'] . '</td>';

echo '<td><a href="/dssv/views/pages/pasaan-container.php?id=' . $row['ContainerID'] .
">Pas aan</a></td>';

echo "</tr>";
}

// close table>

echo "</table>";
?>
<p><a href="/dssv/views/pages/new-container.php">Voeg nieuwe container toe</a></p>

</body>

</html>

```

## Nieuwe container

```
<?php
/*
new-container.PHP
Allows user to create a new entry in the database
*/
// creates the new record form
// since this form is used multiple times in this file, I have made it a function that is easily
reusable
function renderForm($con, $vor, $vol, $die, $error)
{
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Nieuwe container</title>
</head>
<body>
<?php
// if there are any errors, display them
if ($error != "")
{
echo '<div style="padding:4px; border:1px solid red; color:red;">'.$error.'</div>';
}
?>
<form action="" method="post">
<div>
<strong>Container nummer: *</strong> <input type="text" name="ContainerNummer"
value="<?php echo $con; ?>" /><br/>
<strong>Vorm: *</strong> <!--<input type="text" name="Vorm" value="<?php echo $vor; ?>"
/><br/>-->

        <select name="Vorm">
            <?php
                                echo '<option value="Vierkant"> Vierkant
</option>';
                echo '<option value="Rond"> Rond </option>';
            ?>
                                </select><br>
<strong>Volume: *</strong> <input type="text" name="Volume" value="<?php echo $vol; ?>"
/><br/>
<strong>Diepte: *</strong> <input type="text" name="Diepte" value="<?php echo $die; ?>"
/><br/>
<p>* required</p>
<input type="submit" name="submit" value="Toevoegen">
</div>
</form>
```



```

<button onclick="history.go(-1);"> Terug </button>
</body>
</html>
<?php
}
// connect to the database
include('connect-db.php');
// check if the form has been submitted. If it has, start to process the form and save it to the
database
if (isset($_POST['submit']))
{
// get form data, making sure it is valid
$container = $db->real_escape_string(htmlspecialchars($_POST['ContainerNummer']));

$vorm = $db->escape_string(htmlspecialchars($_POST['Vorm']));

$volume = $db->escape_string(htmlspecialchars($_POST['Volume']));

$diepte = $db->escape_string(htmlspecialchars($_POST['Diepte']));
// check to make sure both fields are entered
if ($container == " || $vorm == " || $volume == " || $diepte == " )
{
// generate error message
$error = 'ERROR: Please fill in all required fields!';
// if either field is blank, display the form again
renderForm($container, $vorm, $volume, $diepte, $error);
}
else
{
// save the data to the database
/*
mysql_query("INSERT players SET Bedrijf='$bedrijf', Contactpersoon='$contact',
Telefoon='$telefoon', Email='$email', Straat='$straat', Huisnummer='$huis',
gemeente='$Gemeente', Land='$land'");
or die(mysql_error());
*/
$db->query("INSERT into container (ContainerNummer, Vorm, Volume, Diepte) VALUES
('$container', '$vorm', '$volume', '$diepte')");
// once saved, redirect back to the view page
header("Location: view-container.php");
}
}
else
// if the form hasn't been submitted, display the form
{
renderForm("", "", "", "");
}
?>

```

## Container aanpassen

```
<?php
/*

EDIT.PHP

Allows user to edit specific entry in database

*/

// creates the edit record form
// since this form is used multiple times in this file, I have made it a function that is easily
reusable

function renderForm($id, $con, $vor, $vol, $die, $error) {
    ?>

    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

    <html>

        <head>

            <title>Pas container aan</title>

        </head>

        <body>

            <?php
// if there are any errors, display them

            if ($error != "") {

                echo '<div style="padding:4px; border:1px solid red; color:red;">' . $error . '</div>';
            }
            ?>

            <form action="" method="post">

                <input type="hidden" name="ContainerID" value="<?php echo $id; ?>" />

                <div>

                    <p><strong>Container ID:</strong> <?php echo $id; ?></p>
```

```

        <strong>Container nummer: *</strong> <input type="text"
name="ContainerNummer" value="<?php echo $con; ?>" /><br/>

        <strong>Vorm: *</strong>
                <select name="Vorm">
        <?php
                echo '<option value="Vierkant"> Vierkant
</option>';
        echo '<option value="Rond"> Rond </option>';

        ?>
                </select><br>
        <strong>Volume: *</strong> <input type="text"
name="Volume" value="<?php echo $vol; ?>" /><br/>

        <strong>Diepte: *</strong> <input type="text"
name="Diepte" value="<?php echo $die; ?>" /><br/>

        <p>* Required</p>

        <input type="submit" name="submit" value="Pas aan">

</div>

</form>

</body>

</html>

<?php
}

// connect to the database

include('connect-db.php');

// check if the form has been submitted. If it has, process the form and save it to the database

if (isset($_POST['submit'])) {

// confirm that the 'id' value is a valid integer before getting the form data

if (is_numeric($_GET['id'])) {

```

```

// get form data, making sure it is valid
    //hier zei ge, haal de id uit uw form. Maar ge geeft geen nieuwe ID mee uit uw form. Die hebt
ge al. dus deze regel moet niet.
    $id = $_GET['id'];

    $container = $db->real_escape_string(htmlspecialchars($_POST['ContainerNummer']));

    $vorm = $db->escape_string(htmlspecialchars($_POST['Vorm']));

    $volume = $db->escape_string(htmlspecialchars($_POST['Volume']));

        $diepte = $db->escape_string(htmlspecialchars($_POST['Diepte']));

// check that firstname/lastname fields are both filled in

    if ($container == "" || $vorm == "" || $volume == "" || $diepte == "") {

// generate error message

        $error = 'ERROR: Please fill in all required fields!';

//error, display form

        renderForm($id, $container, $vorm, $volume, $diepte, $error);
    } else {

// save the data to the database
    /*
        mysql_query("UPDATE players SET SET Bedrijf='$bedrijf', Contactpersoon='$contact',
Telefoon='$telefoon', Email='$email', Straat='$straat', Huisnummer='$huis',
gemeente='$Gemeente', Land='$land' WHERE Klantid='$id'")

        or die(mysql_error());
    */
    //ge werkt met mysqli das de improved versie van mysql functies in php. Zijn verschillen
tussen die functies en wat die moeten hebben in de code. deed ne update op user_id=$id ipv
Klantid=$id.
        mysqli_query($db, "UPDATE container SET ContainerNummer='$container',
Vorm='$vorm', Volume='$volume', Diepte='$diepte' WHERE ContainerID='$id'")or
die(mysqli_error($db));

// once saved, redirect back to the view page

```

```

        header("Location: view-container.php");
    }
} else {

// if the 'id' isn't valid, display an error

    echo 'Error!';
}
} else {

// if the form hasn't been submitted, get the data from the db and display the form
// get the 'id' value from the URL (if it exists), making sure that it is valid (checking that it is
numeric/larger than 0)

    if (isset($_GET['id']) && is_numeric($_GET['id']) && $_GET['id'] > 0) {

// query db

        $id = $_GET['id'];

// $result = mysql_query("SELECT * FROM klant WHERE Klantid=$id")
        $result2 = $db->query("SELECT * FROM container WHERE ContainerID=$id") or
die(mysql_error()); //miste ; hier

        $row = mysqli_fetch_array($result2); //ge had mysql functie terwijl uw connectie mysqli is.
(mysqli_improved)
// check that the 'id' matches up with a row in the database

        if ($row) {

// get data from db
//
//dit hoeft niet ge hebt die al ingevuld door uw $id up te vragen na op pas aan te klikken.
// $id = $_POST['id'];

        $container = $row['ContainerNummer'];

        $vorm = $row['Vorm'];

        $volume = $row['Volume'];

        $diepte = $row['Diepte'];

```

```
// show form

        renderForm($id, $container, $vorm, $volume, $diepte, "");
    } else {

// if no match, display result

        echo "No results!";
    }
} else {

// if the 'id' in the URL isn't valid, or if there is no 'id' value, display an error

    echo 'Error!';
}
}
?>
```

# Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:  
**Automatisch uitlezen van vloeistofniveau bij sanitaire containers**

Richting: **master in de industriële wetenschappen: energie-automatisering**  
Jaar: **2017**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

**Antheunis, Stan**

**Cardinaels, Robin**

Datum: **18/01/2017**