

De Waterkering



Rick van Sambeek
Huub van Breugel
V1B Hogeschool Utrecht

02/02/2018

Inhoudsopgave

Inleiding	3
De opdracht	5
Analyse van eisen.....	6
Productontwerp.....	7
De subsystemen.....	8
Huub van Breugel	8
Rick van Sambeek.....	8
Conclusie	10
Overzicht diensten	10
Web App.....	10
Database.....	10
Raspberry Pi's met servo's	10
Eisen voldaan.....	10
Aanbevelingen	11
Bronnen:	11
Bijlage 1: Plan van Aanpak	12
Bijlage 3: Individueel aandeel	18
Huub van Breugel	18
Rick van Sambeek.....	18

Inleiding

Dit verslag wordt geschreven in verband met het IDP project van het eerste jaar HBO-ICT van de Hogeschool Utrecht. Dit project houdt in dat leerlingen uit 4 verschillende rollen konden kiezen en deze leerlingen werden weer in groepjes ingedeeld met de vier rollen in één groepje. Daarna konden de groepjes uit verschillende projecten kiezen: Domotica, Waterkering, Feedbacklamp, OpwegmetTomsBroer, Sportschool. Na enig overleg hebben wij als groep unaniem voor de Waterkering gekozen. We zullen dus in dit verslag uitleggen wat het project precies inhoudt en hoe we het hebben aangepakt en verder. Als eerste zullen we de opdracht zelf toelichten: wat zijn de eisen die er zijn gesteld en wat wil de klant dus graag. Daarna zullen we ons productontwerp toelichten in het algemeen, wat het kan en wat daar voor zorgt. Ook zal elk teamlid apart dieper in zijn aandeel ingaan om een specifiekere omschrijving aan te geven. Tenslotte zullen we aangeven wat het uiteindelijke project is geworden en we geven onze eigen mening over het project. In de bijlage is ons Plan van Aanpak ook te vinden.

De opdracht

De waterkering in de Nieuwe Waterweg, of Maeslantkering, is het laatste onderdeel van de Delta Werken die Nederland beschermen tegen hoog water in het algemeen en de zee in het bijzonder. De Maeslantkering bestaat uit twee grote sluisdeuren die bij extreem weer worden gesloten en zo de nieuwe waterweg afsluiten van de Noord Zee. De sluis wordt bestuurd door een informatiesysteem dat weersgegevens verzamelt en zo een afweging maakt of de sluis gesloten moet worden. Als deze beslissing eenmaal genomen is, kan dit niet meer door een persoon worden teruggedraaid. Eenmaal gesloten, analyseert het systeem de weersgegevens en beoordeelt of en zo ja wanneer de sluisdeuren weer open kunnen. Binnen het te maken Proof of Concept wordt uitgegaan van de waterstanden en niet van de overige weersgegevens. In de echte situatie bepaalt een complexer weermodel of de sluisdeuren moeten worden gesloten en weer geopend. Voor de aansturing van de sluis zijn verschillende disciplines uit de informatica nodig. Zo moet de informatie via sensoren worden verzameld en de sluis uiteindelijk gesloten via een hydraulisch systeem dat aangestuurd wordt door een chip. De informatie die nodig is om de afweging te maken, wordt op verschillende locaties verzameld, om een zo goed mogelijk beeld te krijgen van de weersomstandigheden. Ook is het commandocentrum, waar de uiteindelijke beslissing over het sluiten van de sluis wordt genomen, redundant uitgevoerd. Er is een locatie in de buurt van Hoek van Holland en een locatie meer naar het oosten. Het informatiesysteem zelf verwerkt de gegevens die worden geleverd en bepaald via een weermodel of de sluis moet worden gesloten. Tenslotte zijn er beheer procedures en scenario's die aangeven hoe verschillende instanties (politie, brandweer, ambulance) reageren. Het is aan ons om hiervan een proof of concept te maken om aan te tonen dat het kan werken.

Analyse van eisen

De klant heeft ons als groep verschillende eisen opgelegd waar het systeem aan moet voldoen en wat ze dus willen zien. We beginnen allereerst met het presenteren van het plan van aanpak voor de klant, hier laten we zien hoe we het project aan gaan pakken. Deze presentatie zullen we ondersteunen met een Powerpoint. Moet minimaal één sensor worden ontworpen en geïmplementeerd worden. Deze sensoren moet in staat zijn om het weer, de waterhoogte, de windsterkte en meer te meten en daarna kunnen beslissen of de sluisdeuren open of dicht moeten. Daarnaast is er ook een infrastructuur nodig voor de werking van de Maeslantkering om deze sluisdeuren met de informatie van de sensoren daadwerkelijk open en dicht te laten gaan. Twee motoren moeten voor de beweging van de sluisdeuren zorgen. Deze zijn gekoppeld aan de stations. Er zijn minimaal twee redundante stations waar de beslissingen van het weer en de beslissing om de sluis open en dicht te doen wordt vernomen. Redundant betekent dat het ene station de taken van de ander overneemt, als deze errors heeft of defect is. Dit redundant systeem zorgt ervoor dat de waterkering altijd werkt, als er maar één van de stations volledig werkt. Dit geeft veiligheid in het systeem en tijd om de stations te controleren en te repareren, of te vervangen. Deze stations moeten dus allebei de sluisdeuren kunnen openen en sluiten, ook moeten ze toegang hebben tot de informatie die de sensoren ophalen, zodat ze weten wanneer de sluisdeuren moeten sluiten. Ten slotte moeten ze informatie van elkaar af kunnen lezen, hierdoor kunnen ze achterhalen of één van de stations niet goed of helemaal niet functioneert, om zijn taken daarna over te nemen. Er wordt ook verwacht dat we een overzicht van de betrokken instanties, dus bijvoorbeeld de politie en rijkswaterstaat, en wat hun procedures zijn, wanneer de waterkering open of dicht gaat. Dit alles moet een Proof of Concept opleveren die op 1 februari moet worden gepresenteerd aan de klanten. Deze presentatie zullen we ondersteunen doormiddel van een poster met alle hoofdpunten van het Proof of Concept erop. Op 2 februari moet het geheel ingeleverd worden bij de klant.

Productontwerp

Het Proof of Concept werkt als volgt:

De opstelling bestaat uit twee raspberry pi's. Deze zijn beide aangesloten op de servo's en op de ultrasone sensor. De waterkering wordt allereerst aangezet. Hij start de code op waardoor hij connectie maakt met de andere Raspberry pi en redundant gaat werken. De servo's en de ultrasone sensor worden in dezelfde code ook aangezet. De ultrasone sensor werkt als volgt: de Ultrasone sensor zendt een signaal naar voren toe, daarna gaat hij luisteren en wacht tot dit signaal terug echot. Na een bepaalde tijd vangt de ultrasone sensor het signaal weer op. Deze tijd gebruikt hij om de afstand te berekenen omdat hij weet hoe snel dit signaal gaat. Als je een ultrasone sensor dus boven water hangt en het waterniveau stijgt, duurt het minder lang voordat het signaal terug is en weet hij dus dat het water is gestegen. De raspberry pi leest deze informatie van de sensor af en zet dit in een online database. Vervolgens wordt aan de waarde die de ultrasone sensor meet een 1 of een 0 aangehangen: de 1 staat voor waterkering moet dicht, de 2(blijven) staat voor waterkering moet open(blijven). De reden dat het niet direct op de pi blijft staan is dat er gemonitord kan worden wat er op de raspberry pi wordt gedaan, de Flask web app kan ook in deze database komen en kan zo de standen van het water en de stand van de waterkering laten zien. Vervolgens zal de raspberry pi met een ander stuk code deze database weer uitlezen. Deze code gaat over het aansturen van de servos. Hij leest een 1 of een 0. En op basis van dat cijfer zal de raspberry pi beslissen of de servos open of dicht moeten gaan. Als er iets niet lukt op een raspberry pi dan merkt de andere raspberry pi dit op. Hij zal vervolgens deze taak overnemen van degene die het niet meer kan. Dit overnemen wordt ook weer naar de database verstuurd en deze data komt weer op de Flask web app te staan. Zo kunnen onderhoudsteams ook zien wat er mis is en eventueel het station maken of vervangen als het nodig is.

De subsystemen

Huub van Breugel

Ik heb me tijdens dit project voornamelijk gericht op de flask web app. Voordat ik hiermee begon had ik eigenlijk weinig ervaring met sites maken of HTML, dus ik heb alle informatie tijdens dit project geleerd. Voornamelijk tutorials van Youtube hebben ervoor gezorgd dat het gelukt mij gelukt is om de site te bouwen. Het begon met het maken van een site zonder variabelen. Dit wordt gedaan door middel van het importeren van het importeren van Flask in Python. Na de HTML aangepast te hebben kwam er een layout tevoorschijn die redelijk overzichtelijk was en dus goed genoeg voor een Proof of Concept. Toen werd het tijd om variabelen uit de database te fetchen en in de html te krijgen. Het fetchen van deze data gebeurt door in de python code de webserver, gebruikersnaam, wachtwoord en database in te vullen waardoor de code toegang krijgt tot de database. Dit inloggen staat in een while True loop zodat we er zeker van zijn dat we niet de connectie met de database verliezen, wat ervoor zou zorgen dat de site een error gaf. Daarna wordt het juiste gedeelte van de database geïmporteerd door Python. Door een fetchall wordt de data bruikbaar. Door een kleine aanpassing in de HTML kon de waterstand makkelijk gebruikt worden. De waterkeringstand(open, dicht) was een stukje lastiger: deze kwam als een tuple binnen met de datum en overige informatie binnen waardoor deze eerst gestript moest worden van deze informatie, ook de error codes komen zo binnen.

Naast de flask web app heb ik het overzicht van de instanties gemaakt. De informatie is vooral gebaseerd op de informatie die ik kreeg toen ik Rijkswaterstaat belde op haar informatienummer. De persoon aan de lijn legde mij uit dat er verschillende instanties zoals Veilige Regio's betrokken zijn bij het protocol van rampsituaties. Via de sites van deze instanties heb ik de meeste informatie kunnen bemachtigen die in het overzicht staan.

Rick van Sambeek

Ik heb me vooral ontfermd over de hardware en software van de pi's. Allereerst ben ik begonnen met de servos. Ik heb ervoor gezorgd dat de hardware draaiden via de software. Toen ben ik begonnen met docker. Docker is een module die helpt met het maken van een redundant systeem. In docker maak je een swarm aan voor het besturen van de servos en de informatie. Je kan verschillende raspberry pi's in de swarm stoppen en dan zijn ze met elkaar geconnect. Elke code krijgt een label in de swarm, bijvoorbeeld het bewegen van de servo's en het lezen van de database. Degene die het label heeft, draait de code. Als een raspberry pi zijn werk niet goed doet, kunnen de andere raspberry pi's dit

opmerken. Gebeurt dit, dan wordt het label van die raspberry pi afgepakt en op een andere raspberry pi gezet; hierdoor blijft het systeem alsnog werken ondanks dat er bij 1 raspberry pi niet alles meer werkt. Daarnaast heb ik via een site een SQL database opgezet om de informatie in te stoppen en om sensoren te kunnen simuleren. Ook heb ik ervoor gezorgd dat de pi's de database kunnen lezen en informatie in kunnen schrijven. De sensor is uiteindelijk niet gelukt aan te sluiten maar de theorie is er, dus met een kleine aanpassing zou dit ook moeten kunnen werken.

Conclusie

Na het hele project doorlopen te hebben met enig tegenstoot hebben we het volgende opkunnan leveren:

Overzicht diensten

We hebben een overzicht van de betrokken instanties gemaakt met daarin de procedures die deze instanties zouden moeten volgen wanneer de waterkering open en dicht zal gaan.

Web App

Met Flask hebben we een web app gemaakt waar de informatie die de sensoren moeten halen wordt kan worden afgelezen, ook de errors in de Raspberry pi's zijn hierop te zien.

Database

In de database wordt de informatie gezet die de sensoren zullen meten. Uit deze database halen de Raspberry pi's de gegevens die zo nodig hebben om de waterkering dicht of open te laten gaan. Ook haalt de Web App hier zijn informatie uit.

Raspberry Pi's met servo's

De Raspberry Pi's kunnen de gegevens van de Database verwerken en deze gebruiken om de servo's aan te sturen

Eisen voldaan

We hebben aan de meeste eisen kunnen voldoen, op een enkele na. De redundantie van het systeem hebben we in de praktijk kunnen gebruiken, maar deze is helaas gestopt met werken. In theorie moet het werken. Het zal met de juiste spullen een kleine moeite zijn om deze weer te laten werken.

Het is ook helaas niet gelukt om een sensor aan te sluiten en deze metingen te laten doen. De theorie is er, maar helaas hadden we niet de benodigdheden en tijd om deze te laten werken. Hier zal het dus ook met de juiste benodigdheden een kleine moeite zijn om het aan de praat te krijgen.

Aanbevelingen

We hebben er veel tijd ingestoken en hebben dus ook een aardig idee hoe de Waterkering er uiteindelijk uit moet zien, daarom hebben we een paar aanbevelingen.

- Het redundante systeem bij een waterkering is enorm belangrijk, er moet gezorgd worden dat er meerdere stations zijn die de waterkering aanstuurt zodat de kans van falen nihil is, een waterkering die niet goed werkt kan doden en veel schade met zich meebrengen.
- De motoren van de sluisdeuren moeten goed kunnen sluiten, want een werkend systeem zonder werkende motoren is nutteloos
- Een online database is heel handig, dit zorgt er niet alleen voor dat de klant de informatie kan zien over eventuele errors, maar ook derde partijen zoals instanties.

Bronnen:

-
-
-
-
-
-

Bijlage 1: Plan van Aanpak

Plan van Aanpak			Versie: 1.0	
			Doc. nummer: 1.0	
			Datum: 15-01-2018	
Naam project		Waterkering		
Docent		J. van Reenen		
Klas		V1B		
Naam projectgroep		De Waterratjes		
Projectleider/contact		Huub	Huub.vanbreugel@student.hu.nl	+31633157234
Versiebeheer		1.0		

Creatief Probleem Oplossen:		Welke kwestie gaan we oplossen en hoe gaan we het doen?
Context	Uitgezocht t.o.v. PV	<i>Er moet een Proof of Concept ontwikkelt worden van een waterkering</i>
Kwestie	Afgekaderd t.o.v. PV	<i>Doormiddel van een Proof of Concept kan er aangetoond worden dat een waterkering in de praktijk werkend kan zijn.</i>
Bedoeling	Doelstelling	<i>Een werkbaar redundant systeem maken die op basis van de informatie van sensoren een keuze maakt om poorten open of dicht te doen</i>
	Resultaten	<ul style="list-style-type: none"> -een overzicht van betrokken instanties - informatiesysteem - redundant infrastructuur -implementeren sensoren en servos <p>Specifiek <i>We leveren een Proof of Concept in. Dit gebeurt over een periode van 3 weken. We willen aan te kunnen tonen dat het een veilig product is die in theorie ook in kan worden gezet.</i></p> <p>Meetbaar <i>Het Proof of Concept wordt voor het weerhouden van wateroverlast gemaakt. De volgende punten komen aan bod:</i></p>

		<ul style="list-style-type: none"> - Een overzicht van betrokken instanties en de acties die deze instanties ondernemen. - Een redundant systeem dat ervoor zorgt dat de waterkering blijft werken , als er minimaal 1 pi volledig werkt. - Applicatie(Tkinter, Flask ,etc) waarin de aanwezige informatie gemonitord kan worden door de klant en door de betrokken instanties - Werkende sensoren en servos aan een pi worden gekoppeld die met de informatie van de sensoren de servos kan besturen. <p>Acceptabel De hele groep is gemotiveerd om dit project goed af te ronden en daarmee het doel te bereiken.</p> <p>Realistisch Het is naar ons idee mogelijk om het PoC in de aangegeven tijd goed neer te kunnen zetten en te presenteren.</p> <p>Tijdgebonden Het einddoel is gehaald als de opdrachtgever een goed beeld heeft van ons PoC en een beslissing kan nemen om dit PoC te aanvaarden of een ander plan te bedenken.</p>
Aanpak		
	Deel opdracht 1	<p><i>Doel: Voorbereidingen treffen om het Project zo goed mogelijk in te gaan en het zo goed mogelijk presenteren van het PvA.</i></p> <p>Specifiek <i>Door goed voorbereid te zijn is het makkelijk om te beginnen aan het project en zijn er afspraken over wat er moet gebeuren. Dit zal bijdragen aan de vloeiendheid van het project en ook uiteindelijk aan het PoC.</i></p> <p>Meetbaar <i>Wanneer de presentatie van het PvA met een voldoende is afgerond is onze doelstelling gehaald.</i></p> <p>Acceptabel <i>Ieder projectlid wil het plan van aanpak goed inrichten om het project goed te beginnen.</i></p> <p>Realistisch <i>Het is haalbaar om in de eerste week met de project groep de PvA op te zetten</i></p> <p>Tijdgebonden <i>Het plan van aanpak moet in de eerste week gemaakt worden en akkoord worden gegeven door de projectgroep en de klant.</i></p>
	Deel opdracht ...	SMART
	Deel opdracht n	SMART
Opdracht in één zin		<i>Een Redudante systeem te maken die door informatie van sensoren servos sluit.</i>

Onderzoek doen!	Op welke schouders kunnen we staan en wat weten we nog niet?
------------------------	--

Beschikbare kennis	Begrippen	<ul style="list-style-type: none"> - <i>Proof of Concept</i> - <i>Redundant</i> - <i>Python</i> - <i>Html</i> - <i>Tkinter</i> - <i>Socket</i> - <i>Rpi.GPIO</i> - <i>Packettracer</i>
	Modellen	<i>Netwerkmodel</i>
	Methoden	<p><i>Weber, J., & Jansen, R. (2017). Werkboek HU HBO ICT - Computersystemen en Netwerken 2017. Geraadpleegd van https://cursussen.sharepoint.hu.nl/fnt/51/TICT-V1CSN-15/Studiemateriaal/Werkboek/Werkboek%20HU%20B%20ICT%20-%20Computersystemen%20en%20Netwerken%202017%20v1.3.pdf</i></p>
	Oplossingen	-
Deel-vragen	Per deel-opdracht	<p><i>-BIM: Welke instanties zijn betrokken</i></p> <p><i>-SIE: Wat is de juiste methode om een overzicht te maken van de informatie</i></p> <p><i>-SNE: Met welke methode zetten we een redundant systeem op</i></p> <p><i>-TI: hoe komen we aan de sensoren en servos en hoe halen we er informatie vanaf</i></p>
	Methode	<i>Doormiddel van het opzoeken van informatie op internet,, maar ook door het proberen wat het effectiefs is</i>

Leiderschap, Samenwerken, Communicatie		Wie zijn er betrokken bij het project en hoe gaan we ervoor zorgen dat de samenwerking proactief is en de gewenste impact heeft?	
Docent begeleiders	Eventueel ander betrokken docenten: Jos van Reenen, Bart van Eijkelenburg		
Externe stakeholders			
Studenten	Naam	Functionele rol(len)	Belbin rol(len)
	Huib van Breugel	BIM, projectleider.	Bedrijfsman, Voorzitter
	Piotr Breedveld	TI	Plant
	Rick van Sambeek	TI	Plant
	Leon van Dijk	SNE	Plant
Verwachtingen	We gaan een goedwerkend systeem neerzetten die goed ontworpen is. We zullen naar elkaar luisteren en elkaar waar het kan zoveel mogelijk helpen zodat iedereen van elkaar kan leren. Als er conflicten ontstaan gaan we deze op een volwassen manier uitspreken en met zijn alle tot een beslissing komen. We gaan natuurlijk voor een zo hoog mogelijk cijfer en denken zeker dat deze ambitie haalbaar is		

Communicatie plan	<p>'teamcontract': Noteer hier de afspraken die je met elkaar maakt in verband met de samenwerking en met betrekking tot hoe jullie met elkaar communiceren. Denk hierbij aan:</p> <ul style="list-style-type: none"> - Hoe organiseer je bijeenkomsten? Wie leidt die bijeenkomsten? - Op welke momenten kom je bij elkaar? Vaste momenten? Plan deze momenten onder 5 in. - Hoe leg je afspraken vast? - Waar bewaar je concept- en definitieve documenten? - Hoe bereik je elkaar en hoe wissel je informatie uit? (per mail, groepsapp, dropbox?) - Wat doe je als iemand afwezig is? Als iemand vaak afwezig is? Als iemand zonder melding afwezig is? - Wat doe je bij conflicten?
-------------------	--

Planning & Organisatie, Ethische Verantwoording		Hoe plannen & organiseren we tijd, geld en kwaliteit en verbinden we het project duurzaam met de organisatie?																	
GANTT planning		Week a						Week b						Week c					
Versie/datum:	uren	ma	di	wo	do	vr	zazo	ma	di	wo	do	vr	zazo	ma	di	wo	do	vr	zazo
..... (bijvoorbeeld)																			
Fase X: deelopdrachten																			
1 (kenmerkende titel)		bijvoorbeeld																	
2																			
3																			
Fase Y: deelopdrachten																			
4																			
5																			
6																			
7 Presentatie																			
Team leden		Student nr.			Mobiel			E-mail						Woonplaats					

Huub van Breugel			Huub.vanbreugel@student.hu.nl	
Rick van Sambeek			Rick.vansambeek@student.hu.nl	
Leon van Dijk			Leon.vandijk@student.hu.nl	
Piotr Breedveld			Piotr.breedveld@student.hu.nl	

Beschikbaarheid projectleden gedurende de week:

	Naam	Naam	Naam	Naam	Naam
Ma-o	B	B	B	B	B
Ma-m	B	B	B	B	B
Di-o	B	B	B	B	B
Di-m	B	B	B	B	B
Wo-o	NB	B	B	B	B
Wo-m	NB	NB	B	B	B
Do-o	B	B	B	B	B
Do-m	B	B	B	B	B
Vr-o	B	B	B	B	B
Vr-m	B	B	B	B	b

Geplande projectgroep bijeenkomsten:

Week nummer	Dag/datum/tijd	Vergadering of werkbijeenkomst?	Bij vergadering: voorzitter en notulist
a			
a			
b			

b				
c				
c				
Locatie voor vergaderingen: Skype/ daltonlaan 200 utrecht				
Locatie voor bijeenkomsten: Daltonlaan 200 Utrecht				
Projectarchief cq projectdossier Trello/Dropbox/...				
Hoe is het archiveren en versiebeheer geregeld van zaken als: De code wordt bijgehouden door een gedeelde git op github.com De documenten zullen worden bijgehouden in een gedeelde Onedrive •				
Afhankelijkheden				
Raspberry Pi 3 Servos Ultrasonic sensor				
Risico's				
Bedreiging	Tegenmaatregel	Kans	Effect	Risico
Project loopt uit	Bijeenkomen en bespreken of het nodig is om dingen minder aandacht te geven om belangrijkere onderdelen werkend te krijgen	middel	Groot	We hebben een onvoldoende PoC en we halen het project niet
Groeplid stopt met opleiding	Overleg met docent	Middel	Groot	Kans dat het project uitloopt

Apparatuur gaat kapot	Kijken of we nieuwe apparatuur kunnen regelen	Klein	Middel	We kunnen geen nieuwe apparatuur vinden en kunnen daardoor ons project niet voortzetten

Bijlage 3: Individueel aandeel

Huub van Breugel

Ik heb me vooral bezig gehouden met de web app, daarnaast heb ik het overzicht van instanties gemaakt maar er niet heel veel tijd aan kunnen besteden wegens de grootte van onze groep. Ook heb ik meegekeken met Rick als hij ergens een probleem mee had en we konden het redelijk vaak oplossen. Ik merkt wel dat door Ricks ervaring met programmeren etc. Het best vaak aan mij voorbij ging.

Rick van Sambeek

De werking van de pi's met de servos en het redundant systeem heb ik gedaan, ook heb ik de database opgezet. Daarnaast heb ik Huub bijgesprongen als hij er niet goed uitkwam.