

Young Innovators

Eindrapport Conceptfase

Project 4.0

Team C4 - Young Innovators

Brent De Vos
Matthew Seels
Louis Van Baelen
David van Dongen
Aiman Abdulsalam
Stef Vleugels

2020 – 2021

INHOUDSTAFEL

INHOUDSTAFEL	2
INLEIDING	3
EISENANALYSE.....	4
1.1 Usecase webapp	4
1.1.1 Functionele eisen	5
1.1.2 Niet functionele eisen	7
1.2 Usecase Kiosk	8
1.2.1 Functionele eisen	8
PROOF OF CONCEPT	9
2 DEELRESULTAAT 1	10
2.1 Webhosting	10
2.2 Webhosting security	11
2.3 Sensoren en microcontroller	13
2.3.1 Specificaties	13
2.3.2 Werking en situering in deelresultaat 1	14
2.4 Database setup	15
2.5 Statische webapp	16
2.6 Statische custom app voor de kiosk	17
3 DEELRESULTAAT 2	18
3.1 Connectiviteit.....	18
3.2 Dynamische functionaliteiten webapp	18
3.3 Enquête app kiosk.....	19
3.4 Visualisatie	19
4 DEELRESULTAAT 3	20
4.1 Meerdere sensoren	20
4.2 Voorspellingen	20
4.3 Extra webapp functionaliteiten	20
CONCLUSIE PROOF OF CONCEPT	21
REAL WORLD SCENARIO	23
5 PRIVATE LORA NETWORK.....	23
5.1 Enkele technische specificaties	23
5.2 Uiteindelijke implementatie	25
6 KIOSK OP MAAT	26
6.1 Indoor kiosk.....	26
6.2 Outdoor kiosk	26
SLOTWOORD	27
BIJLAGEN	28
BRONNEN	28

INLEIDING

In dit rapport van de conceptfase willen we verdere toelichting geven omtrent ons project, wat we in de conceptfase uitgedacht hebben, wat we ervan zouden maken indien we onbeperkte tijd & vrijheid hadden om het project tot een mooi einde te brengen en zullen we onze proof of concept beschrijven.

We kregen de opdracht om van Herentals een smart city te maken. Dit houdt in bepaalde dingen te automatiseren en/of te digitaliseren om processen zo efficiënt mogelijk te laten verlopen, specifiek om de stad beter te laten functioneren als bedrijf, als dienstverlener, als democratische ruimte en als fysieke plaats. In dit rapport bespreken we hoe we deze vier aspecten van de stad functioneel zouden verbeteren.

Als bedrijf en als fysieke ruimte willen we de stad sterker maken door middel van efficiëntere afvalophaling dankzij sensoren. Deze sensoren zullen geplaatst worden doorheen de stad om te meten hoe vol vuilbakken zijn. Aan de hand van die data kan gemeentepersoneel op tijd ingrijpen en kunnen er voorspellingen worden gemaakt voor in de toekomst, met behulp van de webapp die we zullen maken.

Met behulp van deze webapp wordt het ook gemakkelijk voor de werknemers om data omtrent Herentals te visualiseren.

De fysieke ruimte van de stad wordt ook verbreed dankzij de constructie van 'informatie-kiosken', waar de burgers simpelweg de Herentals-applicatie kunnen raadplegen op openbare plaatsen. Verder blijven de straten extra proper dankzij de efficiënte afvalophaling wat voor de inwoners aangenaam is.

De stad als dienstverlener en als democratische ruimte zal efficiënter worden dankzij de implementatie van enquêtes die de kiosk-gebruikers kunnen invullen.

Op maandag 14 december 2020 zullen wij u het eindresultaat van de conceptfase toelichten aan de hand van een presentatie. In deze presentatie zal er extra toelichting worden gegeven welke functie elke technologie heeft in ons project en hoe we dit zouden realiseren. We willen alle puzzelstukken samenleggen en het project als 1 geheel aan u presenteren. Ook wordt de proof of concept die in ontwikkeling gaat op 18 januari besproken.

Indien er nog vragen of opmerkingen zijn kan u onze projectleider contacteren:

Louis Van Baelen – Projectleider fase 2
r0695989@student.thomasmore.be

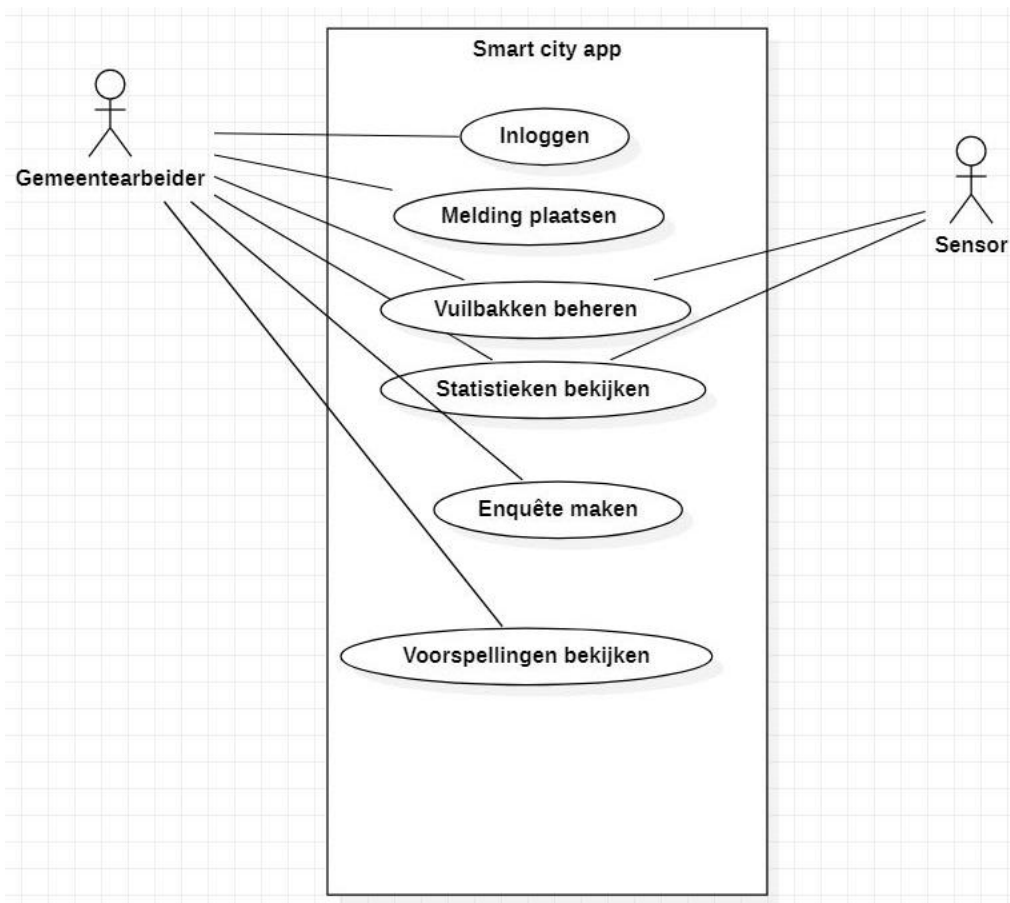
EISENANALYSE

Vooraleer in we in de technische aspecten gedoken zijn hebben we een eisenanalyse uitgevoerd die de functionele en niet functionele eisen van ons systeem diende te bepalen. We hebben gebruik gemaakt van usecase diagrammen om het systeem beter te begrijpen.

De proof of concept zal afwijken van bepaalde eisen.
Voor het real world scenario zouden we al deze eisen natuurlijk wel naleven.

We bieden zowel een webapp voor de gemeentearbeiders aan als een kiosk voor de burgers. Hiervoor zijn 2 usecase diagrammen nodig. Hieronder worden de 2 diagrammen weergegeven en worden de functionele eisen uitgelegd.

1.1 Usecase webapp



1.1.1 Functionele eisen

Inloggen

Functionaliteit: Als gemeentearbeider kan ik inloggen.

Voorwaarde: /

Normaal verloop: Het systeem toont een welkomspagina. De gemeentearbeider klikt op inloggen. Het systeem vraagt voor zijn credentials. De gemeentearbeider vult zijn credentials in en klikt op inloggen. Het systeem verwijst de persoon door naar het dashboard.

Melding plaatsen

Functionaliteit: Als gemeentearbeider kan ik een melding plaatsen.

Voorwaarde: Je bent ingelogd als gemeentearbeider.

Normaal verloop: Het systeem vraagt voor een titel en inhoud. De gemeentearbeider vult dit in en drukt op "plaatsen". Het systeem redirect de persoon naar het meldingsoverzicht met de nieuw geplaatste melding.

Vuilbakken beheren

Functionaliteit: Als gemeentearbeider kan ik vuilbakken beheren

Voorwaarde: Je bent ingelogd als gemeentearbeider.

Normaal verloop: Het systeem toont een overzicht van alle vuilbakken. De gemeentearbeider krijgt verschillende opties om vuilbakken te beheren.

Statistieken bekijken

Functionaliteit: Als gemeentearbeider kan ik statistieken bekijken.

Voorwaarde: Je bent ingelogd als gemeentearbeider.

Normaal verloop: Het systeem toont statistieken van verschillende onderwerpen (vuilbakken, drukke momenten aan kiosken etc.). De gemeentearbeider bekijkt de statistieken.

Enquête maken

Functionaliteit: Als gemeentearbeider kan ik een enquête maken.

Voorwaarde: Je bent ingelogd als gemeentearbeider.

Normaal verloop: Het systeem vraagt om input (titel, beschrijving, vragen). De gemeentearbeider vult dit in en drukt op "enquête maken". Het systeem slaagt de enquête op en redirect de arbeider naar het dashboard.

Voorspellingen bekijken

Functionaliteit: Als gemeentearbeider kan ik voorspellingen bekijken.

Voorwaarde: Je bent ingelogd als gemeentearbeider.

Normaal verloop: Het systeem toont verschillende voorspellingen. De gemeentearbeider kan deze voorspellingen bekijken.

Gebruikers beheren

Functionaliteit: Als admin kan ik gebruikersaccounts beheren.

Voorwaarde: Je bent ingelogd met een admin account.

Normaal verloop: Het systeem toont alle gebruikers met hun rol (admin, gemeentearbeider). Als admin kan je de gebruiker verwijderen, een nieuw account aanmaken of de rol van een gebruiker wijzigen.

1.1.2 Niet functionele eisen

Performantie

Laravel biedt toegang tot een uitzonderlijk interessante hoeveelheid commando's die zeer nuttig zijn bij het stimuleren van de prestaties.

Daarnaast maakt Laravel het gemakkelijker om met meerdere databases te werken en gemeenschappelijke databasebewerkingen uit te voeren door Eloquent object relationele mapper (ORM) te leveren.

Eloquent heeft een positieve invloed op de prestaties van Laravel-applicaties door het concept van 'lazy loading' te hanteren. Hierbij wordt de initialisatie van een object uitgesteld tot op het punt waar het nodig blijkt.

Kwaliteitseisen

Er zijn bepaalde eisen waaraan we moeten voldoen om de beveiliging van het systeem te garanderen.

Deze kwaliteitseisen houden in:

- Informatie beschermen tegen ongeautoriseerde toegang
- Informatie beschermen tegen ongeautoriseerde wijzigingen
- Geautoriseerde gebruikers toegang geven tot de data en resources die ze nodig hebben

Dankzij het gebruik van Laravel zijn deze kwaliteitseisen gemakkelijk te bereiken. Het framework biedt ingebouwde authenticatie- en autorisatiediensten. Zo biedt Laravel een eenvoudig autorisatiemechanisme met twee primaire onderdelen, namelijk gates en policies.

Gates worden namelijk gebruikt om te bepalen of een gebruiker geautoriseerd is om een bepaalde actie uit te voeren.

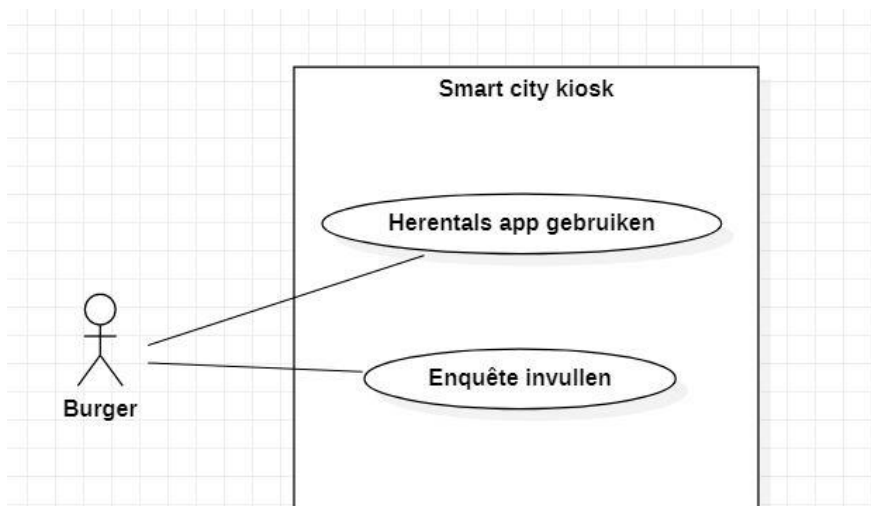
Policies worden binnen een array gedeclareerd en worden gebruikt binnen klassen en methoden die gebruik maken van autorisatiemechanismen.

Zo zal het overduidelijk geen probleem zijn om aan onze autorisatie-eisen te voldoen.

Gebruiksvriendelijkheid

Ook zullen we de toegankelijkheid van de inhoud van onze webapplicatie verhogen door middel van een ingebouwde vorm van gebruiksondersteuning, realiseerbaar door een simpel hulpmenu in de applicatie te voorzien voor gebruikers die onbekend zijn met de verschillende functionaliteiten.

1.2 Usecase Kiosk



1.2.1 Functionele eisen

Herentals app gebruiken

Functionaliteit: als burger kan ik de Herentals app gebruiken.

Voorwaarde: /

Normaal verloop: Het systeem toont de startpagina. De burger klikt op “Naar Herentals app”. Het systeem opent de Herentals app.

Enquête invullen

Functionaliteit: als burger kan ik een enquête invullen.

Voorwaarde: Er zijn enquêtes om in te vullen.

Normaal verloop: Het systeem toont de startpagina. De burger klikt op enquête. Het systeem toont een lijst van enquêtes. De burger kiest een enquête. Het systeem toont de enquête vraagt om input van de burger. De burger geeft zijn input en drukt op “verzenden”. Het systeem redirect de burger terug naar het startscherm.

PROOF OF CONCEPT

Op maandag 18 januari 2021 gaat onze proof of concept in ontwikkeling. Met deze proof of concept willen we de werking van ons systeem bewijzen.

Deze poc zal ontwikkeld worden met beperkte middelen. Het resultaat zal dus afwijken van het real world scenario.

De ontwikkeling van deze proof of concept zal 3 tot 4 weken in beslag nemen. We splitsen de ontwikkeling op in 3 deelresultaten. In elk deelresultaat zijn er nieuwe uitgewerkte onderdelen van ons systeem die telkens een functionaliteit hebben. Welke deelresultaten dit zullen zijn en wat deze inhouden leggen we u graag uit in dit hoofdstuk.

Gemaakte keuzes voor Proof of Concept & Real world scenario

Alle technologieën en hardware die we uiteindelijk gekozen hebben zijn onderworpen aan uitgebreid onderzoek en analyse. Heel dit proces is uitgebreid gedocumenteerd. Omdat dit te veel informatie is om bij in dit rapport te voegen verwijzen we u graag door naar volgende Google Drive map waar u alle onderzoeken, analyses & weighted ranking methods kan vinden voor de technologieën.

<https://drive.google.com/drive/folders/1wJmCbCRYtJbLNOF2rlzfB3P0pigsIDkA?usp=sharing>

2 DEELRESULTAAT 1

Het eerste deelresultaat zal zich afspelen op het einde van de eerste werkweek. Na deze week zullen we alle essentiële basiscomponenten volledig afgewerkt hebben om hierop de komende weken kunnen verder te bouwen. Dit deel van het document beschrijft de geplande stand van zaken

2.1 Webhosting

Criteria	Weight	Requirement Score		
		Zelf hosten	Combella	Versio
Database type	10%	80	85	85
snellheid	15%	65	75	70
prijs	5%	5	90	80
Opslagruimte	15%	75	85	65
Security	15%	50	95	70
Backup	10%	50	80	80
onderhoud tijd	15%	5	95	95
gemakkelijkheid	15%	10	95	90
Weighted Scores	100%	44	87,75	79

Hierboven een visuele representatie van de verschillende variabelen waardoor onze keuze voor de webhosting service beïnvloed werd.



Om onze webapp samen met de benodigde databases en bijkomende infrastructuur te hosten doen wij beroep op de Belgische Hostingprovider Combella.

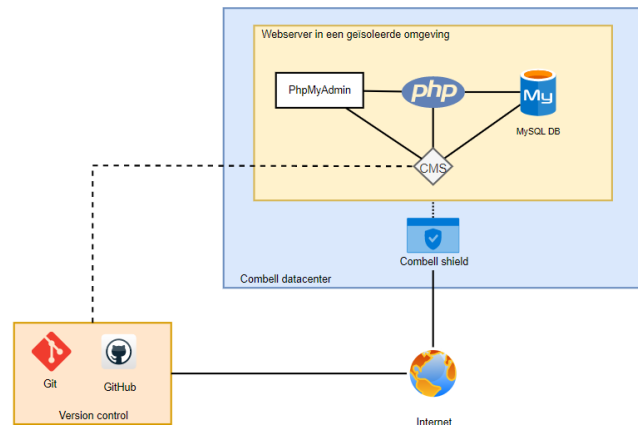
Combella is een webhosting & cloud serviceprovider in België dat als primaire partner gekozen werd door verschillende grote ondernemingen.

Als student hebben wij de mogelijkheid om gebruik te maken van een studenten-licentie. Zo krijgen we de opportuniteit om hun business pakket 1 jaar volledig gratis te gebruiken.

We activeren tijdens deze week ons pakket en maken de omgeving klaar voor ontwikkeling. Dankzij de ingebouwde CMS met GIT plugin kunnen we gemakkelijk telkens nieuwe versies rechtstreeks deployen op de webserver om ze verder te testen.

Op het einde van de eerste werkdag van week 1 zal deze omgeving klaarstaan voor de resterende tijd van werkweek 1.

De topologie zal er als volgt uitzien:



2.2 Webhosting security

We moeten voldoende aandacht besteden aan cyber security. We willen niet dat malafide gebruikers toegang krijgen tot onze data of malware installeren op onze infrastructuur. We overlopen welke maatregelen er allemaal zullen genomen worden om de IT infrastructuur & data zo goed mogelijk te beschermen.

De webserver en databases zullen gehost worden bij hostingprovider Combelle. Zij zullen de security van deze omgeving voor ons verzorgen. Combelle is ISO 9001 en ISO 27001 gecertificeerd.



Combelle noemt hun beveiliging laag de 'Combelle shield'. Combelle shield zorgt voor volgende bescherming:

Applicatie:

- Webfilter: *malafide code of veel gebruikte infecties worden automatisch geblokkeerd. Beveiliging tegen SQL-injection, rootkits en bruteforce attacks zijn eveneens voorzien*
- Scanner: *De scanner scant automatisch de hele website en gaat opzoek naar mogelijke beveiligingslekken en risico's. Er is ook een optie om de gevonden risico's automatisch te 'patchen'.*
- Jailing: *Onze applicatie bevindt zich in een geïsoleerde omgeving. Dit betekent dat onze omgeving afgescheiden is van andere klanten die voor mogelijke beveiliging risico's zouden kunnen zorgen.*
- Logging & IP blocking: *alle trafiek wordt gelogd en geanalyseerd. verzoeken van onbetrouwbare IP's & botnets worden geblokkeerd.*

Netwerk:

- DDoS beveiliging: 24/7 bescherming tegen DDoS attacks.
- Fysieke firewalls: *de beste bescherming wordt gegarandeerd door de meerdere Juniper SRX3400 firewalls. Deze bieden bescherming tegen attacks als SYN en UDP floods*

Operating system:

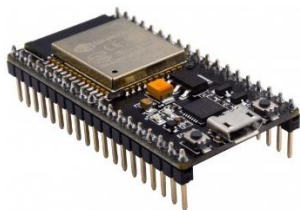
- Firewall op operating system niveau van de servers
- Automatische patching van onze server OS om beveiligingsrisico's door verouderde versies in te dijken

Voor meer info: <https://www.combell.com/nl/hosting/combell-shield>

2.3 Sensoren en microcontroller

Criterium	Gewicht	Requirement Score		
		Arduino Nano 33 IoT	ESP32	LoRaPy4
Energie-efficiëntie	35%	30	90	90
Aansluitingen	15%	90	75	90
Standaard communicatie	10%	70	70	90
Uitbreidbaarheid communicatie	25%	95	95	95
Prijs	15%	85	95	40
Gewogen scores	100%	67,5	87,75	83,75

Met behulp van bovenstaande WRM-analyse kozen we voor het gebruik van een ultrasoon sensoren die gekoppeld worden aan een ESP32 microcontroller. In onze proof of concept zullen we meerdere afstandssensoren gebruiken om de werking van het systeem zo goed mogelijk te illustreren. Omdat we echter enkel toegang hebben tot één ESP32 zullen we ook een sensor met een Raspberry Pi verbinden.



ESP32 microcontroller



Ultrason sensor



Raspberry Pi

2.3.1 Specificaties

Ultrasone afstandssensoren meten de afstand tot een bepaald object door gebruik te maken van de looptijd van een ultrasone geluidspuls. Dit type sensor kan in zowel gewone vuilnisbakken als glasbollen gebruikt worden aangezien er gebruik gemaakt wordt van geluid en kenmerken zoals doorzichtigheid van het afval geen invloed hebben op de meetresultaten.

Stroomverbruik

Het verbruik bedraagt gemiddeld 10 mW (2 mA bij 5 V). Opnieuw is ultrasone afstandsmeting efficiënter dan andere technieken. Dit is bovendien een zeer belangrijk aspect omdat de autonomie van het systeem natuurlijk zo lang mogelijk moet zijn.

2.3.2 Werking en situering in deelresultaat 1

We moeten een script schrijven om de ultrasoon sensor te laten communiceren met de microcontroller (ESP32 en/of Raspberry Pi). In de eerste week willen we de sensoren verbonden hebben aan hun microcontroller en deze uitlezen door een C++ script te schrijven om meetresultaten te verkrijgen. Indien we tijdig klaar zijn zullen we verdere voorbereidingen treffen om meetgegevens in onze database te krijgen.

Connectiviteit

Voor de proof of concept houden we het voor de connectiviteit bij Wi-Fi of ethernet. We zullen in de eerste week ook de microcontrollers zo configureren dat ze aan het internet verbonden zijn. Dit is cruciaal voor de volgende fase om de meetgegevens in onze databases te krijgen.

2.4 Database setup

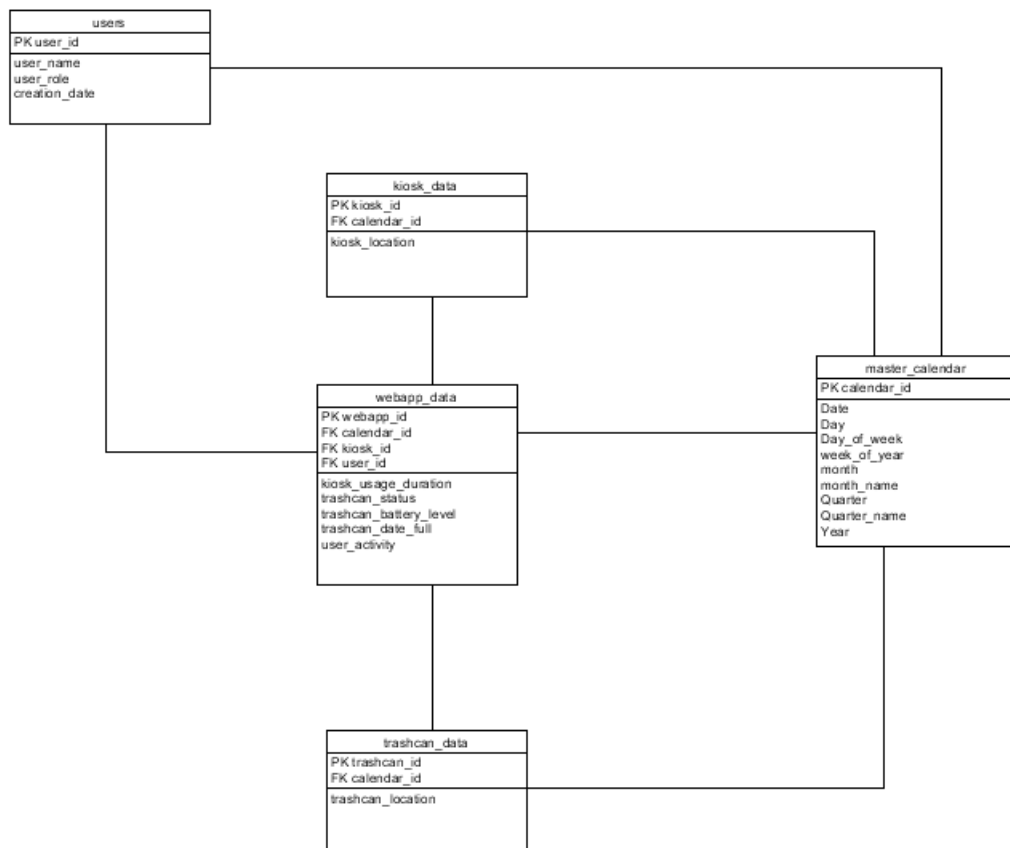


Databases met de benodigde relationele tabellen zullen worden geconfigureerd. We hebben 2 databases waarmee we onze proof of concept willen uitbouwen.

We zullen zowel een Azure database als de database van Combell in gebruik nemen. Hoofdzakelijk zullen we in onze proof of concept gebruik maken van de Combell database maar we zien deze PoC als ideale mogelijkheid om de Azure SQL database ook uit te proberen.

We hebben reeds gecommuniceerd met de klantenservice van Combell. Volgens hen zou hun database ook perfect moeten werken voor deze doeleinden. De data blijft in dit geval kort bij de webserver, wat voor de minimale delay zal zorgen.

Hieronder toegevoegd al een basisschets van hoe de database met relationele tabellen eruit zou zien.



We bevoorraden tabellen voor de sensoren, de kiosken, de verschillende users in het systeem en een algemene kalender tabel. Alle meetbare waardes die getoond zullen worden in de visualisaties in de webapp worden gecentraliseerd in een 'fact' table (webapp_data).

2.5 Statische webapp



Voor de eerste week gaan we concentreren op een goede start. Het opzetten van een github repository voor version control en gemakkelijke samenwerking aan de webapplicatie is een must.

Aangezien we Laravel als backend en React voor de frontend samen willen laten werken gaan we zorgvuldig de basis van het project opstellen met alle benodigde libraries en dependencies. Hiermee kunnen we veel tijd besparen in het voorkomen van problemen in de verdere weken wanneer we met de database gaan werken.

Een van onze eerste doelen voor de webapp is een duidelijke hiërarchie van prioriteit opstellen tegenover al de geplande functionaliteiten en vanuit daar verder te werken. We verdelen en groeperen al de te maken functionaliteiten van de web app alvast op basis van prioriteit.

De verschillende categoriën zullen betreffen; wat moet een niet ingelogde gebruiker kunnen, wat moet een ingelogde gebruiker kunnen en wat moet een admin kunnen.

Hierna kunnen we overgaan op het samenstellen van een huisstijl. Deze zullen we dan over de hele applicatie kunnen toepassen. Deze stijl zal leunen naar de stijl van de Herentals site.

Wat willen we aan het einde van week 1 gedaan hebben.

- Huisstijl is gekozen
- Authenticatie en autorisatie is geïmplementeerd
- Basis project is aangemaakt
- Datamodel aangemaakt
- Testgegevens toegevoegd
- Basis navigatie werkt
- De user roles zijn beslist
- Basis opmaak en layout is gemaakt
- Integratie React en PHP Laravel is gelukt

2.6 Statische custom app voor de kiosk



We wensen het mogelijk te maken voor de kiosk-gebruikers om informatie omtrent de stad te raadplegen doorheen de reeds bestaande 'Herentals' app.

We willen ook echter de functionaliteit aanbieden aan het gemeentepersoneel om enquêtes op te stellen en deze te uploaden op de kiosken die verspreid zullen worden rondom Herentals.

Indien we dit mogelijk willen maken zullen we een customized applicatie maken die onmiddellijk aanspringt bij het opstarten van het apparaat.

Belangrijk is dat deze custom app de enigste applicatie is die gebruikt kan worden op de OS die zal opereren op de kiosk, en ook automatisch opstart eens de kiosk aangezet wordt, of bijvoorbeeld wanneer deze moet rebooten.

Deze kiosk app zouden we programmeren in Flutter. Dankzij het gebruiken van dit mobile framework hebben we op het einde van de productie een app die gereed is voor distributie op zowel IOS als Androidsystemen.

Dit zal handig uitkomen indien de app voor de kiosken later ook nog uitgebreid dient te worden naar een applicatie beschikbaar voor individuen op de App Store of de Play Store.

3 DEELRESULTAAT 2

In de 2de werkweek bouwen we verder op de fundamenteën die we in deelresultaat 1 hebben gerealiseerd. In dit hoofdstuk beschrijven we welke bijkomende functies en bouwstenen we zullen bijbouwen aan de reeds bestaande delen van ons project.

3.1 Connectiviteit

We zorgen ervoor dat onze sensoren hun meetgegevens naar onze database versturen. Dit zal gebeuren door een C++ of python script te schrijven dat connectie maakt met onze database. Doordat de gegevens van de sensoren nu in de databank terecht komen kunnen we onze webapp dynamisch gaan maken.

3.2 Dynamische functionaliteiten webapp

Vanaf de tweede week starten we aan alle dynamische functionaliteiten van de webapp.

Om te beginnen brengen we de initiële opstelling van de verbinding tussen de database en de webapp in orde om zo met een voorbereide backend te beginnen aan de functionaliteiten van de webapp.

We starten met het ontwikkelen van de login-functie, gevolgd door de verdere functionaliteiten. De verwerking van de functionaliteiten dient te gebeuren op basis van de priorisering gemaakt gedurende de eerste week.

In andere woorden starten we bij de administratieve functies en gaan zo geleidelijk aan de functies af, totdat we alle functies per verschillend type gebruiker gereed hebben. Dit houdt dus in: ten eerste de functies van de administratoren, vervolgens de ingelogde gebruiker en ten slotte ook de functionaliteiten van de niet ingelogde gebruiker.

Kort opgesomd wat we willen hebben bereikt aan het einde van week twee:

- Opzetten database (lokaal)
- Admin en gemeentearbeider: Inloggen
- Admin: gebruikers beheren
- Gemeentearbeider: vuilbakken beheren
- Gemeentearbeider: statistieken bekijken
- Gemeentearbeider: enquête maken
- Gemeentearbeider: voorspellingen bekijken

3.3 Enquête app kiosk

Eens we een custom app hebben voor de kiosk, is het de bedoeling om in de kiosk-software te kunnen kiezen tussen het opstarten van de reeds bestaande Herentals-app, en een nieuwe app, namelijk bedoeld om enquêtes te beantwoorden.

Bij het openen van de custom boot app willen we de mogelijkheid bieden waarbij de gebruiker kan kiezen tussen de Herentals app te gebruiken of een enquête in te vullen (die in het real world scenario door de gemeente uitgestuurd kan worden).

Deze applicatie voor enquêtes te tonen en te beantwoorden zal in deelresultaat 2 af zijn en zal natuurlijk ook geprogrammeerd worden in Flutter, net zoals de custom boot app.

3.4 Visualisatie



Als BI-tool voor datavisualisatie kozen we voor Qlik.

Qlik is een modern en volledig webbased data-analyse platform, gemaakt om iedereen binnen een organisatie de mogelijkheid te geven om data gedreven te werken.

Qlik biedt een end-to-end platform met data-integratie, gebruikersgestuurde business intelligence en conversational analytics.

We zullen Qlik gebruiken om de data die gegenereerd wordt door de sensoren en de kiosken te visualiseren.

We zorgen ervoor dat data opgehaald wordt uit onze database en dat hiermee dashboards gemaakt kunnen worden doorheen de webapp.

Deze dashboards zullen embedded kunnen worden op webpagina's om gemeentearbeiders een makkelijk overzicht te geven.

Aangezien we geen functionele kiosk maken in onze proof of concept, zullen we de data hiervan simuleren om zo aan te tonen dat we het gebruik van de kiosken kunnen meten en visualiseren.

4 DEELRESULTAAT 3

In deze laatste fase van de proof of concept zullen we ons focussen op de afwerking van ons PoC en voegen we nog één extra functionaliteit toe.

4.1 Meerdere sensoren

We willen meerdere sensoren kunnen uitlezen en de status en meetgegevens van deze extra sensoren op onze webapplicatie kunnen weergeven.

We zullen deze functionaliteit in de proof of concept aantonen a.d.h.v. Raspberry Pi's, potmeters en/of ultrasoon sensoren om meetgegevens te simuleren.

4.2 Voorspellingen

Eens we na de tweede week de data gevisualiseerd krijgen, gaan we de functionaliteit voorzien voor de admins en de gemeente-medewerkers om met deze data voorspellingen maken.

Bijvoorbeeld kunnen er voorspellingen gemaakt worden over drukke momenten aan kiosken, hoe snel bepaalde vuilbakken vol zitten, ...

Indien we tijdens de voorgaande weken niet genoeg data hebben ontvangen van onze sensoren zullen we genoeg data simuleren zodat de AI voldoende info heeft om mee te werken.

4.3 Extra webapp functionaliteiten

Aan het einde van week drie willen we dat de functionaliteiten van de web app volledig af zijn. Verder willen we ook dat de applicatie getest is en gehost kan worden. We willen ook de security op frontend en backend werkend hebben. Dit gaat dan over validaties op front -en backend maar ook over beveiliging van routes en inloggen.

Wat willen we aan het eind van de derde week af hebben:

- Frontend validaties
- Backend validaties
- Testing frontend
- Hosting applicatie
- Alle functionaliteiten

Onze webapplicatie zal in gereedheid gebracht worden voor de Demo. Het belangrijkste aspect is dat de meetresultaten van de 'vuilbakken' op de webapp te zien zullen zijn

CONCLUSIE PROOF OF CONCEPT

Voor het maken van onze proof of concept wouden we zo kort mogelijk bij een realistische situatie blijven, echter hebben we toch wat beperkingen.

We kozen voor moderne technologieën die hun steentje zeker zullen bijdragen in onze concepten bedoeld om van Herentals een smart city te maken.

Zo hebben we voor 1 van de beste hosting providers gekozen in heel Europa en gebruiken we gerenommeerde en efficiënte programmeertalen zoals PHP en React. We namen ook de keuze om een type sensor te gebruiken die binnen een real world scenario niet zou onderdoen in vergelijking met andere sensoren.

Nog een zeer belangrijk punt voor ons was om onze visualisatie-mogelijkheden uit te werken in onze proof of concept. Het hoofddoel van visualisatie in onze opstelling is dat de meetgegevens van de sensoren en het gebruik van de kiosken visualiseerbaar is.

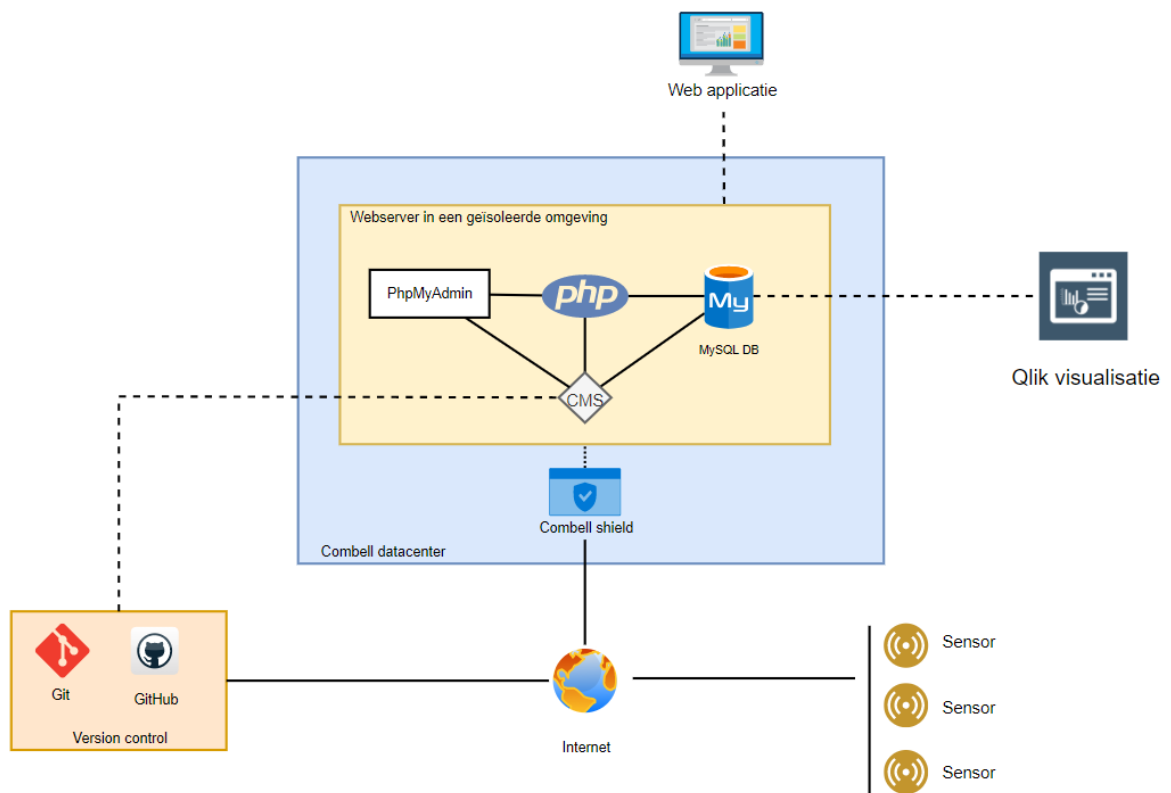
In een real world scenario zouden we in de toekomst nog meer IoT-oplossingen hebben die data versturen, deze data zouden we dan kunnen visualiseren en gebruiken om voorspellingen te maken waar nodig of waar het bruikbaar is.

Daarom vonden we het zeer belangrijk om de werking rondom visualisatie van data afkomstig uit IoT-sensoren te bewijzen in onze proof of concept.

Ook de vier aspecten (Herentals als...) wouden we graag aan deze PoC koppelen. 'Stad als: dienstverlener, fysieke plaats, bedrijf en democratische ruimte' komen in deze PoC aan bod.

Dankzij het slimme vuilbakken systeem blijft de stad als fysieke plaats proper en hebben inwoners steeds voldoende ruimte om hun afval kwijt te kunnen. De stad als bedrijf kan zijn arbeiders efficiënt inzetten dankzij een aangepaste planning voor afvalophaling en door de mogelijkheden van de kiosk te onderzoeken leggen we de eerste hand om de stad als democratische ruimte te gebruiken (enquêtes invullen aan een kiosk). Ook op vlak van dienstverlening biedt de kiosk veel voordelen.

In de volgende figuur wordt de totaalopstelling van onze proof of concept die we zullen ontwikkelen weergegeven.

Proof of concept infrastructuur blueprint:

REAL WORLD SCENARIO

Bij onze proof of concept willen we zo kort mogelijk bij de realiteit blijven. Toch is dit niet altijd mogelijk. Indien we onbeperkt veel tijd, geld en middelen hadden zouden we enkele dingen anders aanpakken. We leggen in dit hoofdstuk uit welke onderdelen we dan zouden willen realiseren.

5 PRIVATE LORA NETWORK



Criterium	Gewicht	Requirement Score		
		Sigfox	LoRa	Wi-Fi
<i>Data rate</i>	10%	50	60	95
<i>Airtime</i>	25%	50	50	100
<i>Energie-efficiëntie</i>	25%	95	95	50
<i>Prijs</i>	20%	90	90	30
<i>Security</i>	20%	80	80	80
Gewogen scores	100%	75,25	76,25	69

Voor de connectiviteit zouden we een private LoRa netwerk aanleggen voor de stad Herentals, dit bleek de beste oplossing te zijn met behulp van de bovenstaande WRM-analyse.

Steden zoals Oostende of grote industriële gebieden zoals de haven van Antwerpen connecteren hun IoT-oplossingen ook via hun eigen LoRa netwerk.

Enkele specificaties van LoRa en hoe we het zouden implementeren vindt u hieronder.

5.1 Enkele technische specificaties

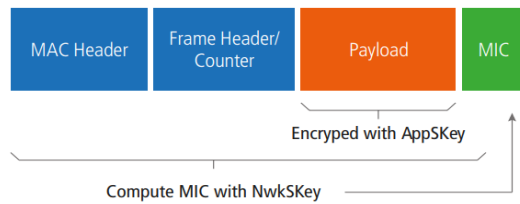
Data rate

Het LoRa netwerk biedt data rates tussen 0.3 kbit/s en 50 kbit/s. Voor de relatief kleine hoeveelheid berichten die verzonden zouden worden, moeten deze snelheden ruim genoeg zijn. Er is bovendien een limiet van 100 bytes applicatiedata per bericht. In tegenstelling tot SigFox (12 bytes) is dit aanzienlijk meer.

Energie

LoRaWAN is een zeer energie-efficiënt communicatieprotocol. Hierdoor wordt ervoor gezorgd dat de autonomie van de microcontrollers niet opvallend verkort wordt door het gebruik van dit protocol wat in toepassingen zoals vuilnisbakmonitoring zeer belangrijk is.

Security



Berichten die over het LoRa netwerk verstuurd worden zijn 128bit end-to-end encrypted. LoRaWAN trafiek is beschermd door twee session keys.



Elke payload is encrypted met AES-CTR en heeft een frame counter (om packet delay tegen te gaan) en een message integrity code (MIC) berekend met AES-CMAC.

LoRaWAN applicatie berichten zijn naast deze security ook:

- Origin authenticated
- Integrity protected
- Replay protected
- Encrypted zoals eerder beschreven

5.2 Uiteindelijke implementatie

Om voor Herentals een private LoRa netwerk op te zetten. Zouden we LoRa gateways willen aanschaffen. Deze gateways zouden we op 2 plaatsen willen installeren om in de volledige stad Herentals bereik voor onze (toekomstige) sensoren te voorzien. Eén in de Lakenhal kunnen installeren, de andere plaatsen we bijvoorbeeld in Morkhoven in de parochiezaal.

Deze gebouwen zijn beiden eigendom van de stad en liggen ook op een centrale plaats.



Afbeelding: kaart Herentals met locatie van de LoRa gateways

Future proof

Met deze investering kunnen we niet alleen onze efficiënte vuilbak monitoring voorzien van connectiviteit maar maken we Herentals ook klaar voor de toekomst. Herentals zal komende jaren meer en meer smart city oplossingen willen implementeren in hun stad (slimme water metingen, meten luchtkwaliteit, smart mobility, ...). Deze smart city oplossingen hebben tevens ook connectiviteit nodig die we met deze opstelling reeds voorzien zullen hebben.

Nieuwe sensoren kunnen makkelijk verbonden worden met het LoRa netwerk.

6 KIOSK OP MAAT

We zouden op verschillende plekken in Herentals een kiosk willen plaatsen (indoor of outdoor) waar mensen de Herentals app kunnen gebruiken of een enquête die uitgestuurd wordt door de gemeente kunnen invullen.

6.1 Indoor kiosk

Indien we een indoor locatie hebben (bv. de bibliotheek) Zouden we een vastgemaakt beveiligde tablet voorzien. Met beveiligd bedoelen we dat de gebruiker niets anders kan doen dan de Herentals app te gebruiken of een enquête invullen.

Het voordeel van het gebruiken van een tablet als een kiosk is dat deze heel goedkoop is. Bovendien kan deze ook overal geplaatst worden en neemt weinig ruimte in beslag.

Tablet kiosk modellen:

- Samsung Galaxy Tab A 10.1
- Samsung Galaxy Tab A 7.0

6.2 Outdoor kiosk

Voor een outdoor kiosk moeten we met veel factoren rekening houden:

- Goede bescherming tegen weersomstandigheden
- Scherm met hoge helderheid om goede leesbaarheid te garanderen.
- Intelligent koel en verwarmingssysteem om aanpassingen aan de interne temperatuur te maken bij verschillende temperaturen tijdens de koudere en warmere maanden.
- Gehard veiligheidsglas om de touchscreen te beschermen tegen vandalisme

We zouden dan beroep doen op een gespecialiseerd bedrijf die een kiosk op maat voor Herentals kan voorzien. De uiteindelijke uitrol van de software op de kiosk en de verbinding met het netwerk kunnen we zelf onder handen nemen.

Als extra veiligheidsfactor kunnen we in de kiosk een camera inbouwen die continu alles opneemt. Dit kan op zijn beurt dan weer gebruikt worden voor het monitoren van drukte in een omgeving, ...

In dat geval moeten we wel rekening houden met de GDPR-wetgeving.

(Dit laatste deel rondom de camera is een andere smart city oplossing die we niet voorzien. Maar we willen aantonen dat er veel extra mogelijkheden zijn bij de installatie van een outdoor kiosk).

SLOTWOORD

Zo komen we tot het slot van ons eindrapport. We geloven reeds veel bereikt te hebben, maar dit was echter nog maar het begin van het echte werk.

Na veel onderzoek te hebben geleverd vinden we onszelf gevuld met vertrouwen in ons gepland traject om de conceptfase tot een goed einde te brengen.

Snel genoeg beginnen we aan onze praktijkgerichte plannen om alle uitgedachte concepten om te zetten in een functionerende basisversie van toepassingen bruikbaar in een Smart City.

BIJLAGEN

Google Drive met alle onderzoek & keuze documenten + analyses:

<https://drive.google.com/drive/folders/1wJmCbCRYtJbLNOF2rIzfB3POpigsIDkA?usp=sharing>

BRONNEN

Meer over de Combell Shield security:

<https://www.combell.com/nl/hosting/combell-shield>

Oostende kiest voor LPWAN (LoRaWan):

<https://www.dtplan.be/post/2017/07/11/smart-city-oostende-datalogging-via-lpwan>

LoRa netwerk uitrol in de Antwerpse haven:

<https://www.computable.be/artikel/nieuws/ict-branche/5744010/5440850/nazca-rolt-lora-netwerk-uit-in-antwerpse-haven.html>