

Design Description

voor

Project Snuffelneus

Projectleider	:	Björn Braber
Projectleden	:	Raymen Scholten Edwin Schriek Pim van Dijk Peter Baan Paul Zoetewey Björn Braber
Begeleider	:	M. Dubbeld
Opdrachtgever	:	B. Flipsen
Datum van uitgifte	:	24 januari 2014
Referentie	:	Design.pdf

Versiebeheer

Versie	Datum	Wijzigingen	Auteur
0.1	24-09-2013	Aanmaak van document	Björn Braber

Inhoudsopgave

1	Introductie	1
1.1	Probleemstelling	1
1.2	Doelstelling	1
1.3	Opdracht	1
1.3.1	Omschrijving	1
1.3.2	Projectgroep	2
1.4	Doelgroep	2
1.4.1	Omvang	2
2	Vooronderzoek	4
2.1	Product scenario	4
2.1.1	Voor Kinderen	4
2.1.2	Voor volwassenen	5
2.1.3	Conclusie scenario	5
2.2	Fijnstof sensor	5
2.2.1	Sensortypen	6
3	Functioneel ontwerp	8
3.1	Data context diagram	8
3.2	Data flow diagram	8
4	Architectuur ontwerp	9
4.1	Architectuur hardware	10
4.1.1	Schema	10
4.1.2	Temperatuursensor	10
4.1.3	NO2 Sensor	10
4.1.4	Bluetooth module	11
4.1.5	voeding	11
4.1.6	Microcontroller	11
4.2	Architectuur server	11
4.2.1	Schema	12
4.2.2	Web API	12
4.2.3	Database	12
4.3	Architectuur applicatie	14
4.3.1	Schema	14
4.3.2	Bluetooth interface	15
4.3.3	GPS interface	15
4.3.4	GPRS interface	15
5	Detail ontwerp	16
5.1	Snuffelneus applicatie	17
5.1.1	Alarm manager	18
5.1.2	Background service	18
5.1.3	Bluetooth controller	18

5.1.4	GPS controller	18
5.1.5	Database controller	18
5.1.6	HTTP controller	18
5.1.7	User Interface	18
5.2	User Interface	20
5.2.1	Beginscherm	22
5.2.2	Map view	22
5.2.3	Instellingen	22
5.2.4	Log scherm	22
5.3	Communicatie	22
5.3.1	Van Snuffelneus naar App	22
5.3.2	Van App naar Snuffelneus	23
5.3.3	Van App naar Server	23
5.3.4	Van Server naar App	24
5.4	Snuffelneus hardware	24
5.5	Server	24
5.5.1	Web API	25
5.5.2	Database	25
5.5.3	Azure	26
5.5.4	Metingen opslaan	26
5.5.5	Metingen ophalen	27

1 Introductie

In dit document wordt het ontwerpproces van de snuffelneus in detail beschreven. In het eerste gedeelte wordt de architectuur beschreven, gevolgd door het detail ontwerp

1.1 Probleemstelling

Er wordt door veel onderzoekers en media beweert dat de luchtvervuiling sterk toeneemt en dat deze toename zorgt voor een verhoogd risico voor de volksgezondheid. Er zijn slechts op enkele plekken in een stad meetpunten voor de luchtvervuiling. Hierdoor kan niet goed in kaart gebracht worden welke invloed de luchtvervuiling op een persoon heeft. Een persoon verplaatst zich natuurlijk en wordt dus niet blootgesteld aan dezelfde vervuiling dan wat wordt gemeten door de vaste meetpunten.

Indien de luchtvervuiling gemeten zou worden door een individueel persoon, kan deze informatie in kaart gebracht worden wanneer de gegevens centraal worden verzameld. Hierdoor kan nauwkeuriger worden onderzocht hoeveel invloed de luchtvervuiling heeft op de mens.

1.2 Doelstelling

Het doel van dit project is om een apparaat te ontwikkelen voor een zeer lage prijs om de luchtkwaliteit te meten. Dit apparaat moet klein zijn en moet tevens data kunnen opsturen naar een centrale database.

1.3 Opdracht

Voor het tweede project binnen de minor Embedded Systems op de Hogeschool Rotterdam is een opdracht geformuleerd door een opdrachtgever. De projectleden dienen aan deze opdracht een volwaardige invulling te geven.

1.3.1 Omschrijving

De opdracht behelst het ontwerpen van een interactief product “de Snuffelneus”, waarmee de gebruiker de luchtkwaliteit in zijn of haar omgeving kan meten. Er moet rekening worden gehouden dat het product ten allen tijden de luchtkwaliteit moet kunnen meten en dat de gebruiker gemotiveerd moet zijn om dit ook daadwerkelijk te doen. Het product dient tegen een zo laag mogelijk kostprijs geproduceerd te worden. Tevens moet het platform over uitbreidingsmogelijkheden beschikken waardoor er mogelijke uitbreidingen of verbeteringen kunnen worden doorgevoerd in de nabije toekomst.

Kortom: Ontwikkel hardware waarmee de mate van luchtvervuiling waaraan een individueel persoon dagelijks wordt blootgesteld gemeten kan worden. De luchtkwaliteit moet kunnen worden teruggekoppeld aan de gebruiker. Tevens

is het van belang dat het apparaat eenvoudig te bedienen is voor ook niet-technische gebruikers en zo klein mogelijk wordt gebouwd.

1.3.2 Projectgroep

De samenstelling uit de projectgroep bestaat uit vier studenten van Technische Informatica (TI), twee studenten Elektro Techniek (ELE) en drie studenten van Industrieel Product Ontwerp (IPO). Verder zijn er twee docenten die het project begeleiden waarvan één docent elektrotechniek en de ander lector is binnen de Hogeschool Rotterdam

1.4 Doelgroep

Dit document is geschreven voor de volgende betrokken partijen: - Projectleden
- Begeleider / coach / docent van de opleiding elektrotechniek - Hogeschool Rotterdam - Geïnteresseerde studenten

1.4.1 Omvang

Er zijn zes projectleden actief van de minor Embedded Systems gedurende 18 tot 20 lesweken. Het aantal manuren is 6 (projectleden) maal 7,5 (ECTS) maal 28. Het aantal uren is dus 1260 uur evenredig verdeeld over de projectleden. Dat is 10,5 uur per week per persoon.

De drie projectleden van industrieel product ontwerp zijn tevens actief gedurende 18 tot 20 weken. Het aantal manuren is daarbij 3(projectleden) maal 15 (ECTS) maal 28 Het aantal uren is dus 1260 uur evenredig verdeeld over de projectleden. Dat is 21 uur per week per persoon.

De drie projectleden van industrieel product ontwerp zijn tevens actief gedurende 18 tot 20 weken. Het aantal manuren is daarbij 3 (projectleden) maal 15 (ECTS) maal 28 Het aantal uren is dus 1710 uur evenredig verdeeld over de projectleden. Dat is 21 uur per week per persoon.

2 Vooronderzoek

Voor het kunnen meten van de luchtkwaliteit is een sensor nodig die in staat is een meting uit te voeren die ook daadwerkelijk wat zegt over de luchtkwaliteit. In dit project moet een stof of gas gemeten worden die direct schadelijke invloed op de mens. Om te weten welke stof of gas geschikt is om te meten is wordt er een onderzoek verricht.

2.1 Product scenario

Voor het project zijn door de opdrachtgever twee “hoofd-scenario’s” gegeven. In het eerste scenario wordt een product bedoeld wat voor kinderen is bestemd. In het tweede scenario is het product voor volwassen bestemd. In dit hoofdstuk wordt beschreven wat de verschillen zijn tussen de scenario’s en waarop uiteindelijk gericht moet worden.

2.1.1 Voor Kinderen

Het product wordt ontwikkeld voor kinderen. Dit betekent dat er andere keuzes gemaakt moeten worden betreft hardware. Zo is het van belang dat een kind het product wil dragen. Een kind is zich namelijk niet of nauwelijks bewust van de negatieve gevolgen van fijnstof. In het geval dat het product bestemd is voor kinderen dient er dus een uitdagende / speelse factor toegevoegd te worden wat kinderen stimuleert het product te dragen. Kinderen van jonge leeftijd hebben over het algemeen gezien geen smartphone. Ondanks dat de drempel steeds lager wordt tegenwoordig heeft de doelgroep kinderen (6 -12) geen smartphone. Dit betekent dat het product standalone moet kunnen werken. Wanneer er fijnstof gemeten moet worden is ook van belang dat de locatie en de tijd waarop het gemeten wordt bekend is. Verder moet de data opgeslagen worden, wat op een later moment uitgelezen kan worden. In dat geval zijn de volgende componenten minimaal vereist:

- Fijnstofmeter
- GPS module
- Crystal (voor tijd)
- Data logger (Micro SD)
- Interactieve visuele weergave meting
- Voeding

Normaal gesproken kan de tijd en gps waarden m.b.v. een smartphone worden verkregen. Doordat er bij deze doelgroep niet van uit kan worden gegaan dat deze over een smartphone beschikken zullen de essentiële componenten zoals de GPS module en de crystal in het apparaat zelf moeten worden gebouwd. Deze twee componenten zijn relatief duur (Gemiddeld tien euro of duurder) waardoor de kostprijs flink wordt opgeschroefd.

2.1.2 Voor volwassenen

Wanneer de doelgroep volwassenen betreft kan er van uit worden gegaan dat met beschikt over een smartphone. Dit impliceert dat er gebruikt gemaakt kan worden van de interne GPS module en tijd van de smartphone. Daarnaast kan de verkregen data opgeslagen worden op de smartphone. Om gegevens echter te kunnen koppelen dient er een verbinding met de smartphone te zijn. Dit kan draadloos gebeuren of bedraad. Er is in dat geval één component extra nodig die de communicatie faciliteert maar daardoor kunnen er drie componenten weggelaten worden omdat gebruik gemaakt kan worden van de interne sensors van de smartphone. De communicatie module is relatief goedkoop ten opzichte van de GPS, crystal of micro SD card module. Een goed voorbeeld is daarbij een Bluetooth module die al voor minder dan acht euro kan worden aangeschaft. Dit is vele malen goedkoper als de drie modules die het vervangt.

2.1.3 Conclusie scenario

In het geval dat het product wordt ontwikkeld voor kinderen zal de kostprijs flink stijgen en hierdoor niet voldoen aan de eisen van de opdrachtgever (Minder dan 25 euro kostprijs). Daarnaast wordt het project gecompliceerder doordat de extra modules ook meer ruimte in beslag nemen. Dit zal negatieve gevolgen hebben voor de mobiliteit van het product. De omvang neemt toe en de batterijduur zal sterk afnemen. Deze factoren kunnen de haalbaarheid van het project in gevaar brengen. Wanneer het product wordt ontwikkeld voor volwassenen kan het product tegen een lagere prijs worden geproduceerd. Tevens zal het product dan ook kleiner zijn waardoor dat meer overeenkomt met de wensen van de opdrachtgever. Voor een eerste prototype van het product is de beste keuze om te kiezen voor het scenario dat het product bestemd is voor volwassenen. Voor een overzicht van alle min en pluspunten zie Bijlage I

2.2 Fijnstof sensor

Er zijn twee verschillende fijnstofsensoren getest. Met behulp van het programma ‘Serial Chart’ is het mogelijk om realtime de output op de seriële poort te bekijken en daarnaast maakt het programma een lijn-grafiek van de gemeten gegevens. Aan de hand van deze gegevens kunnen de sensoren getest worden.

Eén van de sensoren is een ‘TCST 2103’. Deze bestaat uit een module waar twee componenten in zitten, een infrarood LED en een lichtsensor. Er wordt gemeten hoeveel licht er vanaf de LED over komt over een afstand van enkele millimeters. Deze sensor is niet gevoelig genoeg en is eigenlijk gemaakt om te bekijken of er wel of geen voorwerpen tussen de twee onderdelen zitten. Ook is niet bekend welke fijnstof deeltjes de sensor meet.

Als tweede sensor is gekeken naar een ‘Grove – Dust Sensor’, deze sensormodule heeft als sensor de ‘SHINYEI PPD42NS’. Deze sensor moet verticaal worden gebruikt wat betekent dat de sensor bijvoorbeeld met zijn rug tegen de muur moet worden bevestigd. Het principe van deze sensor is gebaseerd op hetzelfde

principe als de andere sensor. Met een infrarood LED en een sensor die meet hoeveel licht er overkomt wordt de hoeveelheid fijnstof bepaald. De Dust Sensor bevat ook een lens. Door de lens kan de sensor zich op één punt richten, het brandpunt. Als er een stofje groter dan 1 micrometer langs komt gaat het signaal omlaag en wordt er het aantal keren dat het signaal laag is geteld. De sensor is ongeveer 30 seconden bezig met 1 meting waarbij het aantal stofdelen worden waargenomen. Ook bevat deze sensor een weerstand dat warm wordt. Door de verwarming ontstaat er een kleine luchtstroom wat de lucht langs de sensor voert.

Er is een prototype ontwikkeld die aangesloten is op de Dust Sensor. Elke 30 seconden wordt er gemeten waarbij de waarden worden opgeslagen in een bestand. Dit bestand wordt in CSV formaat opgeslagen op een SD kaart. Tevens worden de waarden verstuurd over de seriële poort. Via Excel kan er een grafiek gegenereerd worden aan de hand van de opgeslagen gegevens. Tijdens het werken aan de prototype zijn we op school waar weinig stof te meten is. De sensor is getest tijdens het koken van een hamburger. Hierbij werd er een verhoogde hoeveelheid van fijnstof gemeten.

2.2.1 Sensortypen

Binnensensoren De sensoren zijn beide fijnstofsensoren die uitsluitend geschikt zijn voor binnen. De gebruikte eenheid voor fijnstof is in mg/m^3 . Dit betekent het gewicht van de stofdeeltjes per inhoud lucht. De gebruikte sensoren meten beide geen luchtstroom. Er wordt dus uitgegaan van een constante luchtstroom in een statische omgeving binnenshuis. Op deze situatie is een kalibratie-tabel opgesteld waarmee de gemeten spanning omgezet kan worden in de standaard eenheid. De opdrachtgever wilt graag buiten gaan meten, maar buiten is er geen constante luchtstroom zoals binnen. Wanneer het hard waait zal er meer fijnstof gemeten worden met de huidige sensoren terwijl dat niet zo hoeft te zijn, de luchtflow is dan groter.

Dit betekent dat de sensoren onaangepast niet geschikt zijn om buiten te meten. Om dit op te lossen moeten de sensoren voor buitengebruik uitgebreid worden met extra sensoren om de luchtstroom te kunnen meten. Een andere optie is om een bepaalde behuizing te maken zodat de luchtstroom gelijk is als in de situatie binnenshuis.

De opdrachtgever wil indien mogelijk verschillende grootten van fijnstof kunnen onderscheiden. De huidige sensoren beschikken niet over deze functionaliteit. De manieren om dit mogelijk te maken is doormiddel van het toevoegen van een filter. Een nadeel is echter dat het filter na verloop van tijd vol loopt en vervangen moet worden. Er moet kunnen worden vastgesteld wanneer het filter verstopt is om dit probleem op te lossen.

Een andere optie kan zijn om het analoog te onderscheiden. Dit kan erg

complex worden. De IR LED schijnt op een bepaalde inhoud waarbij verwacht wordt dat de uitgangspanning hetzelfde is bij verschillende samenstellingen van fijnstof. Bijvoorbeeld wanneer twee stofdeeltjes van $2\mu\text{m}$ langs de sensor passeren als wanneer één stofdeeltje van $4\mu\text{m}$ passeert.

Er zijn waarschijnlijk nog een hoop andere manieren om de stofdeeltjes te kunnen onderscheiden. Er moet gekeken worden naar de specifieke eigenschappen van een groter stofdeeltje ten opzichte van de kleinere deeltjes. Mogelijk hebben deze verschillende frequentieresponsie of allerlei andere verschillende eigenschappen. Dit wordt te complex en zal waarschijnlijk niet haalbaar zijn.

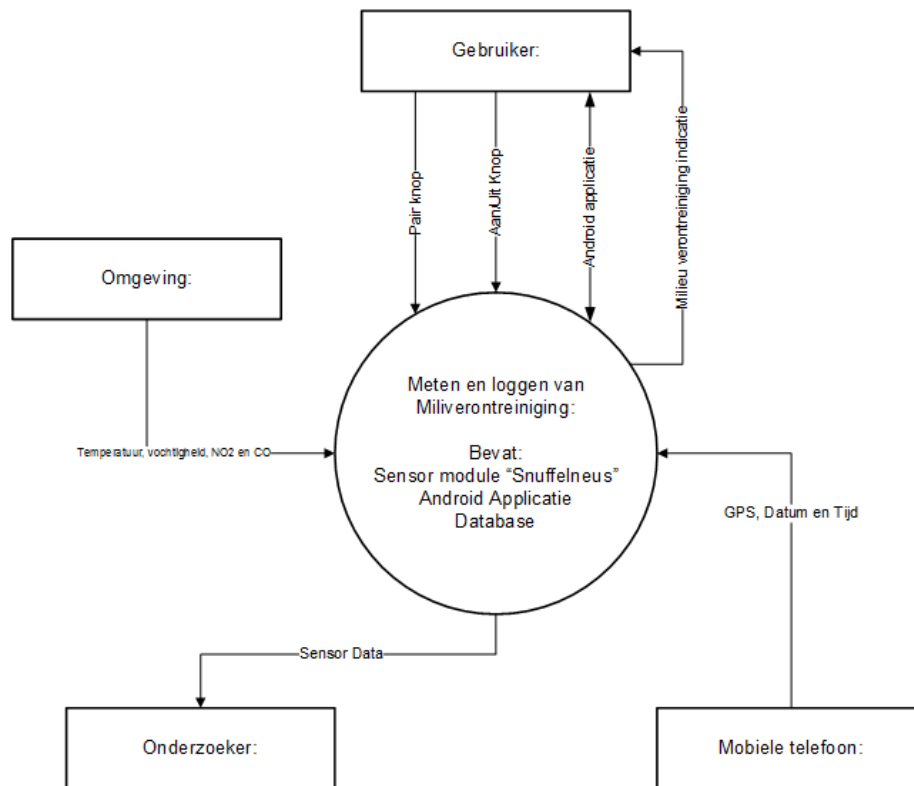
Buitensensoren

3 Functioneel ontwerp

In het functioneel ontwerp wordt er onderzocht wat het systeem moet doen en welke onderdelen benodigd zijn om dit te volbrengen.

3.1 Data context diagram

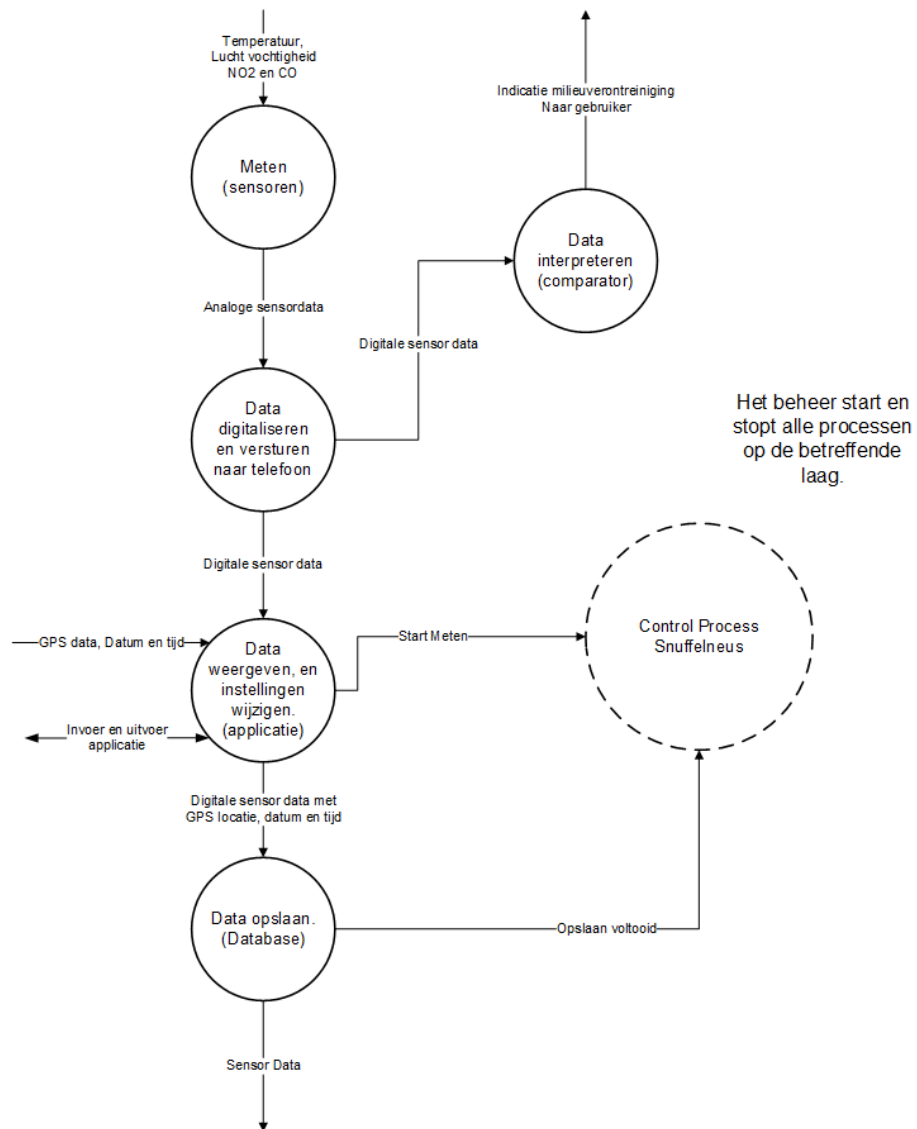
Het datacontextdiagram geeft de afbakening van het project weer (Zie Figuur 1). Het hoofdproces bestaat uit: Meten en loggen van de milieuverontreiniging. De afbeelding geeft weer hoe het snuffelneus systeem in zijn omgeving staat. De vierkante blokken staan voor de omgeving, deze vallen buiten de scope van het project, dit houdt in dat ze niet gemaakt hoeven worden.



Figuur 1: Data context diagram

3.2 Data flow diagram

Het dataflowdiagram deelt het hoofdsysteem op in allerlei subsystemen. De subsystemen geven de functionele bewerkingen weer.



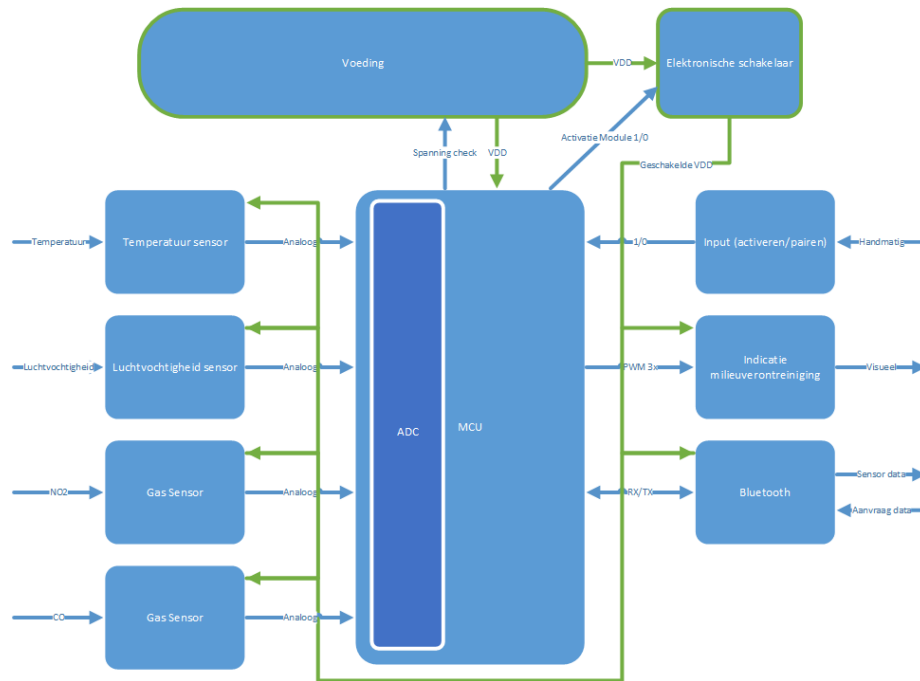
Figuur 2: Data Flow diagram

4 Architectuur ontwerp

In het functioneel ontwerp wordt er onderzocht uit componenten het systeem moet bestaan. De benodigde componenten en interfaces worden hier in kaart gebracht.

4.1 Architectuur hardware

4.1.1 Schema



Figuur 3: Hardware Schema

4.1.2 Temperatuursensor

Voor kalibratiedoeleinden is er een temperatuursensor aanwezig op de snuffelneus. De gebruikte gassensoren zijn temperatuurgevoelig. Door het meten van de temperatuur kunnen de sensordata softwarematig gekalibreerd worden. Belangrijke eisen aan de temperatuur sensor zijn:

1. Goedkoop
2. Klein
3. Zuinig

4.1.3 NO2 Sensor

Uit onderzoek is gebleken dat NO2 een betrouwbare indicatie geeft van de milieuverontreiniging. Om deze reden zal er in het snuffelneus project een NO2 Sensor worden gebruikt om de luchtkwaliteit te meten. Aan de sensor zitten een aantal eisen:

1. Klein
2. Uitleesbaar als weerstand
3. Kan buiten gebruikt worden

4.1.4 Bluetooth module

De bluetooth module zal zorgen voor de communicatie tussen de snuffelneus hardware en de mobiele telefoon waar de applicatie op draait. Hierdoor is het mogelijk om meetgegevens terug te koppelen aan de gebruiker. De bluetooth module heeft de volgende eisen:

1. Goedkoop
2. Compatible zijn met Bluetooth 2.0
3. laag verbruik

4.1.5 voeding

Het systeem moet gevoed worden uit een accu of batterij. De batterij moet voldoende capaciteit hebben om minimaal één dag mee te gaan. Er moet een circuit beschikbaar zijn die de batterij capaciteit indicatie kan meten zodat de gebruiker weet wanneer de batterij bijna leeg is.

1. Goedkoop
2. Klein
3. Minimale capaciteit van 500 mAh
4. Oplaadbaar

4.1.6 Microcontroller

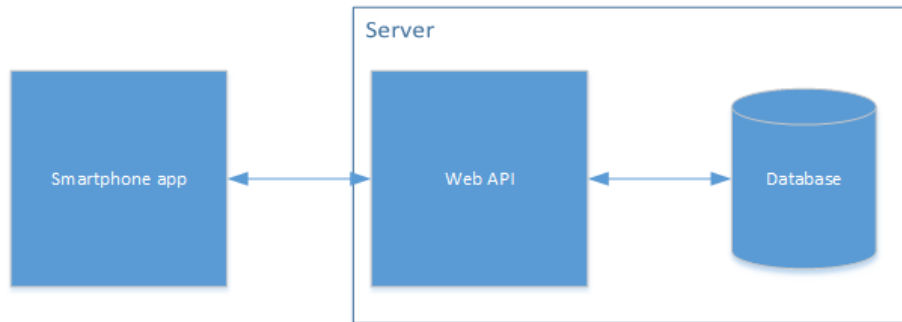
De microprocessor zorgt voor het aansturing van de hardware. De processor moet de sensoren starten en de gedigitaliseerde sensordata kunnen wegschrijven naar de bluetooth module. Belangrijke eisen aan de microcontroller zijn:

1. Laag verbruik
2. Minimaal acht analoog digitaal converters
3. Dertig beschikbare I/O ports, voor eventuele uitbreiding
4. Twee maal uart

4.2 Architectuur server

De server zorgt voor de samenvoeging van alle metingen die door de applicatie worden verstuurd. De server moet vanaf overal bereikbaar zijn zodat ten alle tijde metingen kunnen worden opgeslagen.

4.2.1 Schema



Figuur 4: Server schema

4.2.2 Web API

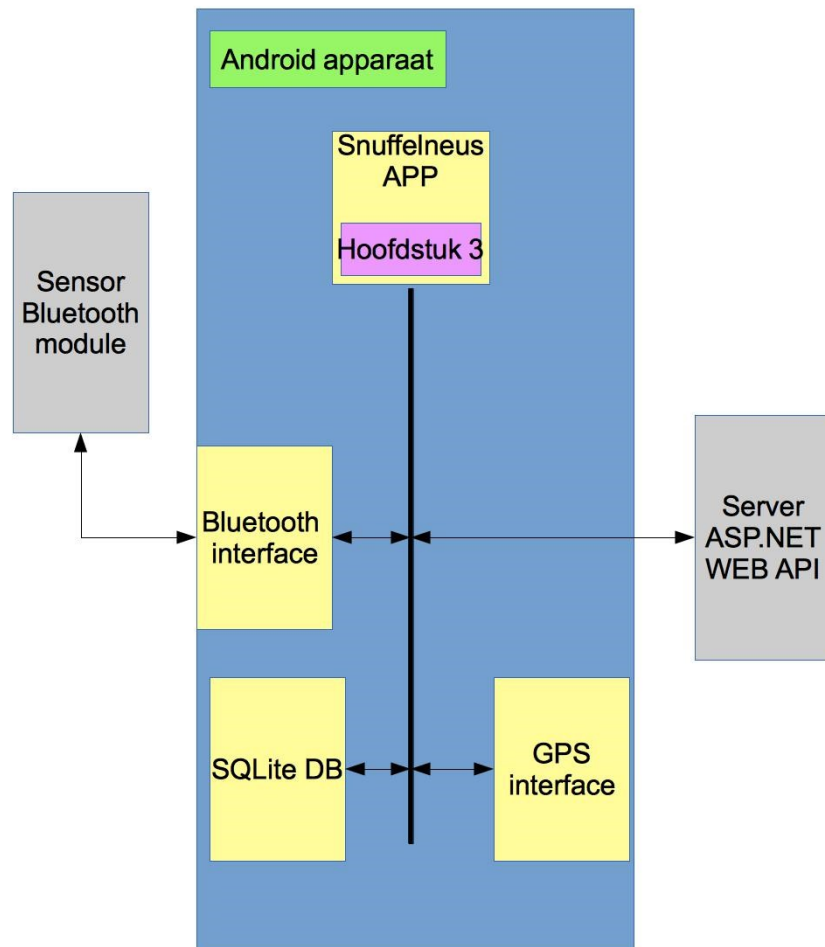
Om waarden op te slaan in een database wordt een API gemaakt. Deze API zal via HTTP aanvragen krijgen van de smartphone applicatie om data terug te sturen of op te slaan. Een API is een interface waarmee gecommuniceerd kan worden om acties uit te voeren op een resource.

4.2.3 Database

Voor het opslaan van alle metingen is een database vereist. Hierin zullen alle metingen met identificatie worden opgeslagen.

4.3 Architectuur applicatie

4.3.1 Schema



Figuur 5: Applicatie schema

De snuffelneus applicatie voor Android is verder uitgewerkt in diagram blabal. De applicatie maakt gebruik van verschillende hardware componenten die in het gros van de smartphones standaard aanwezig zijn. De onderdelen die benodigd zijn voor een goede werking van de snuffelneus applicatie zijn:

1. Bluetooth module
2. GPS module
3. GPRS module

Om deze modules aan te spreken is er in de applicatie voor elke module een interface aanwezig die communiceert met de desbetreffende hardware module in de smartphone. Verder maakt de applicatie gebruik van een SQLite database voor het lokaal opslaan van sensordata.

4.3.2 Bluetooth interface

De Bluetooth interface van Android praat met de Bluetooth module van het apparaat. De interface zal connecties en datastromen tussen de Snuffelneus en de applicatie regelen. De interface kan beide versturen en ontvangen.

4.3.3 GPS interface

De GPS interface van Android praat met de GPS module van het apparaat. Deze interface is hierdoor in staat een GPS coördinaat te verkrijgen waardoor de sensordata gekoppeld kan worden aan

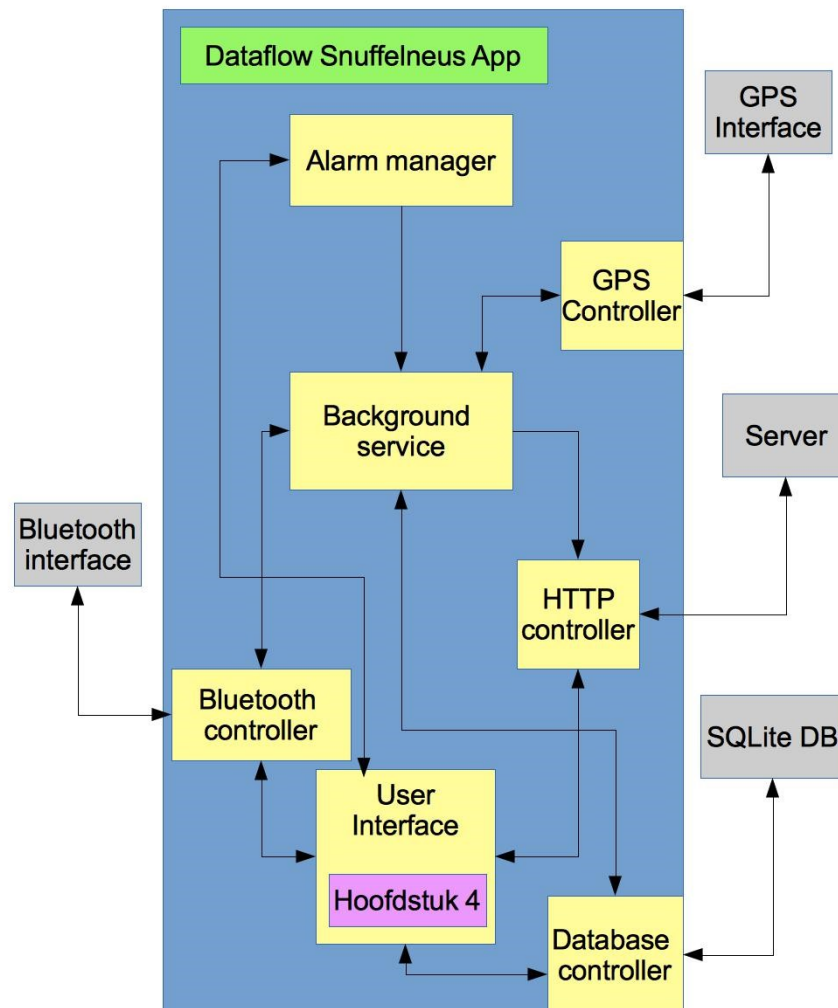
4.3.4 GPRS interface

De GPRS interface van Android praat met de GPRS module van het apparaat. Hierdoor is het mogelijk in de applicatie te communiceren met het internet. Via het internet wordt vervolgens gemeten data verstuurd naar de server.

5 Detail ontwerp

In dit hoofdstuk wordt de architectuur van het product in detail beschreven waardoor het systeem in detail inzichtelijk wordt.

5.1 Snuffelneus applicatie



Figuur 6: Applicatie Schema

5.1.1 Alarm manager

De Alarm Manager geeft toegang tot de alarm services van het Android systeem. Deze services maken het mogelijk om een applicatie te starten ergens in de toekomst. In ons geval wordt de Alarm Manager gebruikt om na elke periode de Background service te starten.

5.1.2 Background service

De background service regelt het meet proces (m.b.v. de Bluetooth controller), het loggen van de data (lokaal, m.b.v. de Database controller) en het loggen van de data (extern, m.b.v. de http controller). Voor het loggen van de data wordt ook de GPS controller aangesproken om de locatie te bepalen. De background service wordt gepland door de Alarm Manager. Zie figuur 7 op pagina 19.

5.1.3 Bluetooth controller

De Bluetooth controller zorgt er voor dat er op de juiste manier en tijdstippen de benodigde berichten naar de Snuffelneus worden verstuurd. De Bluetooth controller ontvangt ook de data van de Snuffelneus en geeft deze door naar de Background service. De Bluetooth controller wordt bestuurd door de Background service en de User Interface.

5.1.4 GPS controller

De http controller zorgt er voor dat er op de juiste manier gemeten data wordt verzonden naar de web api van de server. De http controller heeft ook de mogelijkheid om data van de server op te halen. De http controller wordt bestuurd door de Background service en de User Interface.

5.1.5 Database controller

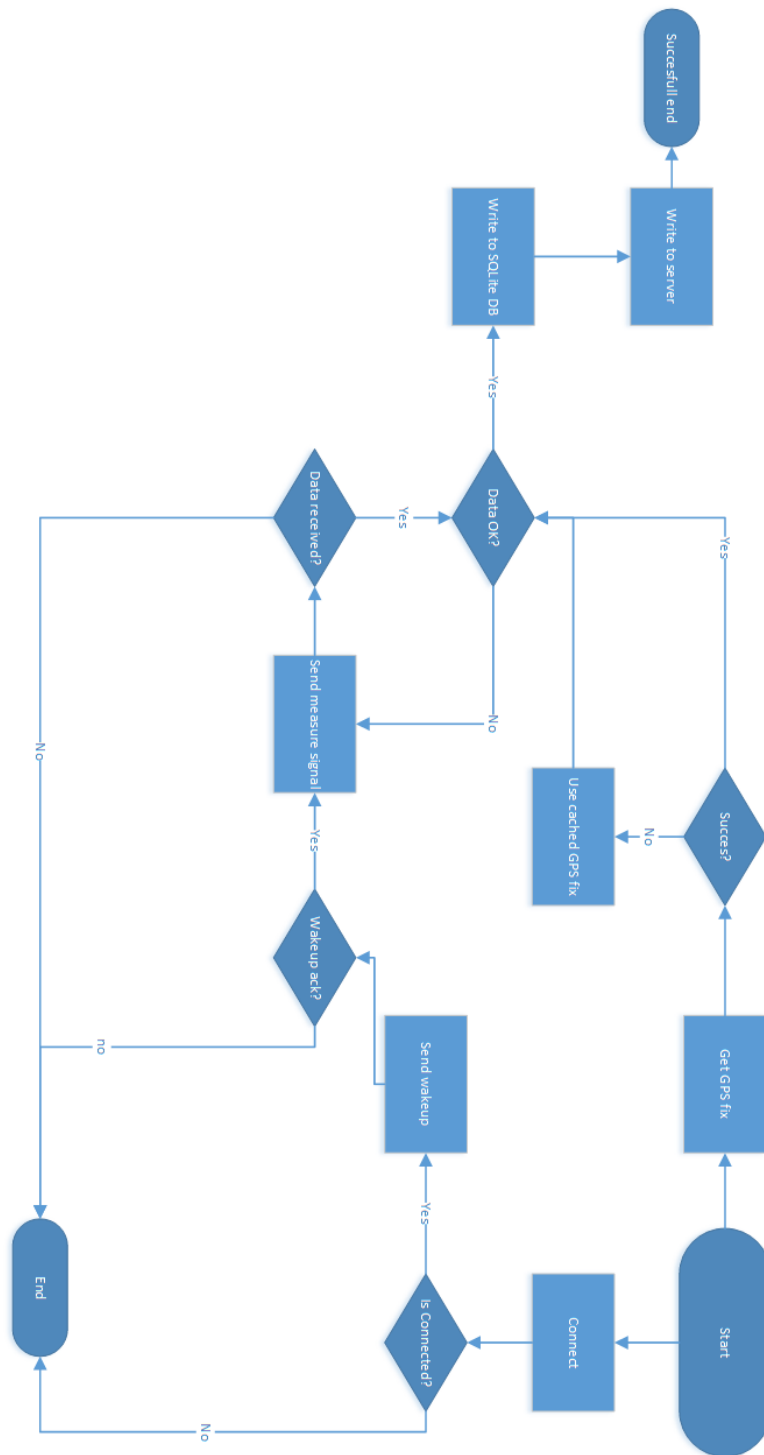
De database controller zorgt voor het lezen en schrijven naar de SQLite database. De database controller wordt bestuurd door de background service en de User Interface.

5.1.6 HTTP controller

De http controller zorgt er voor dat er op de juiste manier gemeten data wordt verzonden naar de web api van de server. De http controller heeft ook de mogelijkheid om data van de server op te halen. De http controller wordt bestuurd door de Background service en de User Interface.

5.1.7 User Interface

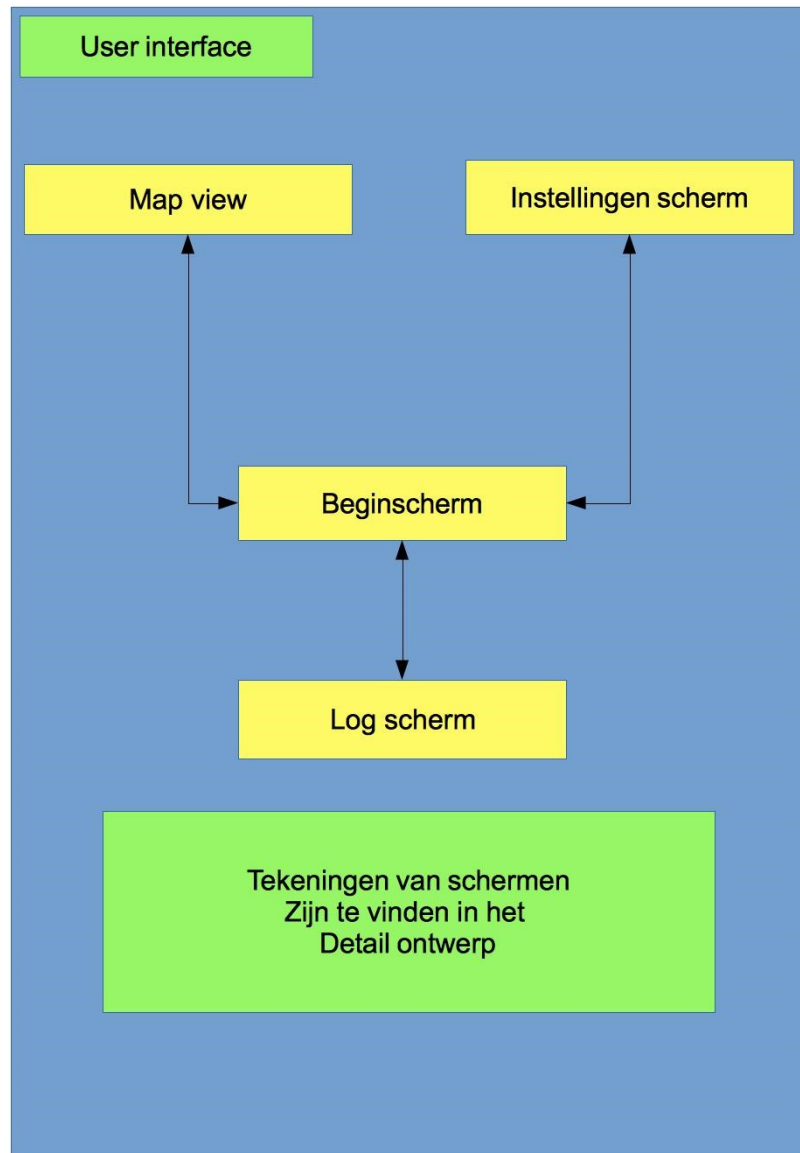
De user interface zorgt voor interactie tussen de gebruiker en de applicatie. De werking van de userinterface staat verder uitgewerkt in de volgende paragraaf



Figuur 7: Background service flow

5.2 User Interface

De user interface bestaat uit vier verschillende schermen waar de gebruiker kan komen. In de toekomst kan dit nog worden aangepast. Hoe de schermen eruit moeten komen te zien komt is te lezen in het nog te maken detail ontwerp.



Figuur 8: Interface Schema

5.2.1 Beginscherm

In het beginscherm komen gegevens over het gekoppelde embedded system en de status van de verbinding. Hierin kan ook worden weergegeven wanneer de laatste communicatie is geweest en ander nieuws over de app.

5.2.2 Map view

In de map view moet een Google Maps kaart worden weergegeven. Hierover komt een laag met de kleur van de luchtkwaliteit. De gebruiker moet gegevens kunnen filteren op bijvoorbeeld alle gebruikers of alleen de gebruiker zelf. Ook kan worden gekozen of een enkele data moet worden weergegeven, denk hierbij aan temperatuur, luchtvochtigheid of de kwaliteit in zijn algemeenheid.

5.2.3 Instellingen

Door middel van het scherm Instellingen kan de gebruiker enkele instellingen aanpassen. Bijvoorbeeld het opnieuw koppelen met een embedded system.

5.2.4 Log scherm

In het log scherm moeten de gegevens worden weergegeven welke de App uit de sensor heeft gekregen. Dit kan simpel door de waardes welke zijn verkregen in een lijst te laten zien. In de toekomst verdwijnt dit scherm mogelijk, omdat deze moet worden gemaakt om te alle gegevens uit te lezen tijdens het testen.

5.3 Communicatie

Om het de communicatie tot in detail te bekijken, wordt hieronder besproken welke variabelen kunnen worden verstuurd tussen de verschillende apparaten. De apparaten zijn de Snuffelneus, de telefoon (de App) en de server.

Bij de verbinding tussen de Snuffelneus en de App wordt de grootte van de variabelen vastgesteld. Dit is nodig, omdat de Snuffelneus in bits werkt. Dit kan in de App worden afgevangen en omgezet worden in waardes in de vorm van bijvoorbeeld een integer of een String.

5.3.1 Van Snuffelneus naar App

De communicatie tussen de Snuffelneus en de App wordt gedaan met het Bluetooth Protocol RFCOMM. Dit protocol wordt door bijna alle apparaten met Bluetooth gebruikt.

De variabelen welke verstuurd kunnen worden zijn:

- sensor1 (10 bits)
- sensor2 (10 bits)
- sensor3 (10 bits)

- batterylvl (8 bits)

Voor batterylvl moet worden bekeken of dit kan worden berekend door de Snuffelneus.

5.3.2 Van App naar Snuffelneus

De App kan alleen bytes versturen. Dit houdt dus in dat alleen waardes verstuurd kunnen worden met groottes van meervouden van 8 bits. De variabelen welke verstuurd kunnen worden van de App naar de Snuffelneus zijn:

- wakeupcall (8 bits)

Na de wakeupcall kan de App ontvangen van data, de inputstream gaat open. Ook hier geldt dat batterylvl afhankelijk is van of het mogelijk is om in de Snuffelneus het batterijniveau te berekenen.

5.3.3 Van App naar Server

De communicatie tussen de App en de Server wordt tot stand gebracht over het internet. De App kan dit doen met behulp van de interfaces van de telefoon, zoals 3G, 4G en Wi-Fi. De Server is direct met het internet verbonden door middel van WiFi of een ethernet kabel.

De data wordt verstuurd met JSON. De grootte van deze data kan variabel zijn.

De variabelen welke verstuurd kunnen worden zijn:

- aanvraag
- timestamp
- sensor1
- sensor2
- sensor3
- longitude
- latitude
- imei(of andere identifier)

5.3.4 Van Server naar App

De gegevens worden opgeslagen in de SQLite database. Door een verzoek te versturen weet de Server welke data terug gestuurd moet worden. Keuze tussen data van alle gebruiken en alle data wordt gemaakt door de App. De variabelen welke verstuurd kunnen worden zijn:

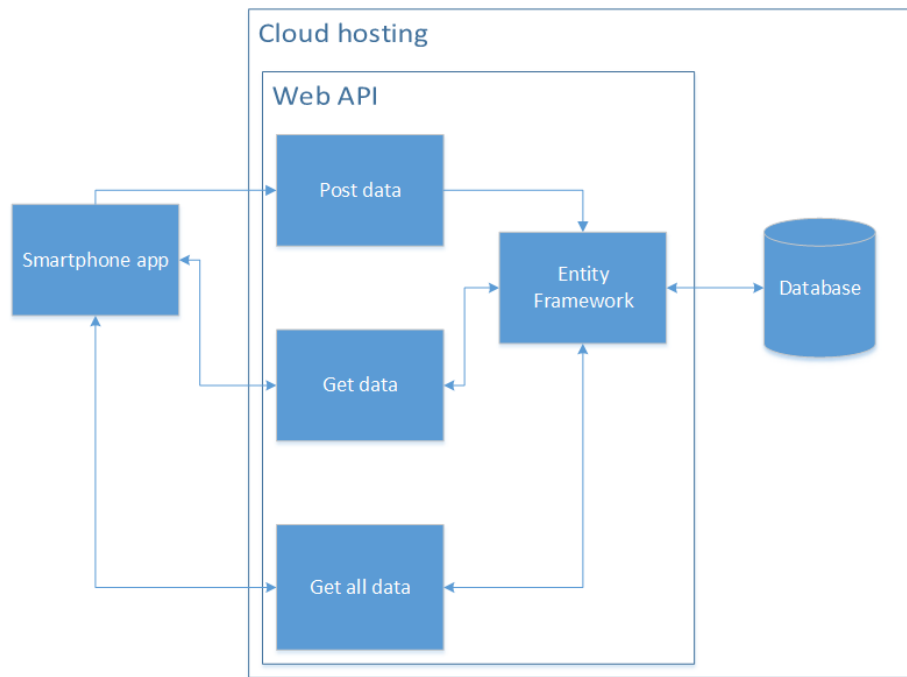
- timestamp
- sensor1
- sensor2
- sensor3
- longitude
- latitude

5.4 Snuffelneus hardware

Alle informatie over de Snuffelneus Hardware is te vinden in "snuffelneus-hardware.zip".

5.5 Server

De server bestaat uit meerdere onderdelen die in de komende hoofdstukken worden uitgelegd. Voor het programmeren van de Web API wordt het ASP.NET Web API framework gebruikt. Deze API zal communiceren met de database die in SQL Server wordt gehost. Beiden instanties staan in de cloud op Azure. Al deze technologieën zijn van Microsoft. Na de beschrijvingen van de onderdelen worden alle mogelijke acties beschreven. Het overzicht van de server is te vinden in Figuur 9.



Figuur 9: Server detail schema

5.5.1 Web API

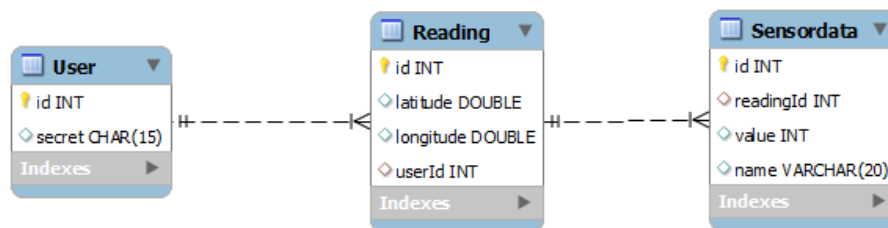
De Web API ontvangt HTTP aanvragen van de smartphone applicatie om acties uit te voeren. Deze aanvragen worden automatisch doorgestuurd naar een controller. De API zal bestaan uit één controller die de verschillende data aanvragen afhandelt. Een controller bevat de volgende methodes om de data aanvragen uit te voeren:

- PostData
Data opslaan naar de database
- GetData
Apparaatspecifieke data terugsturen
- GetAllData
Gecombineerde data terugsturen

5.5.2 Database

Om alle gegevens op te slaan is wordt een relationele database gebruikt. De ERD is te vinden in Figuur 10. De database bevat drie tabellen voor alle gegevens die verbonden zijn door middel van vreemde sleutels. Elke snuffelneus of smartphone wordt aangegeven als User die geïdentificeerd wordt door een sleutel

van 15 karakters. Wanneer er data wordt ontvangen en deze gevalideerd is dan wordt de Reading tabel bijgevuld met locatiegegevens en van welke gebruiker de informatie verstuurd is. In de ontvangen data staat een lijst met sensorlezingen, voor elke van deze lezingen wordt de Sensordata tabel ingevuld met de waarde en de naam van de sensor. De sensorlezingen worden verbonden met de Reading tabel. Op deze manier kan er vanaf elke tabel informatie opgevraagd en via vreemde sleutels ook van andere tabellen.



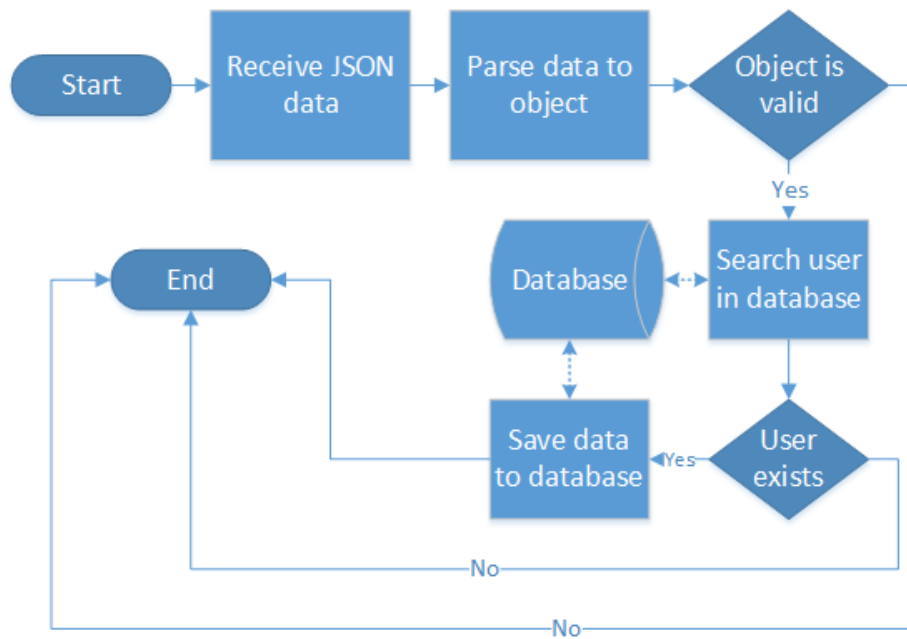
Figuur 10: Database schema

5.5.3 Azure

Zowel de API als database worden gehost in de Azure cloud van Microsoft. Er wordt gebruik gemaakt van een hosted website, hierin draait de API dat technisch gezien een website is. De database wordt gehost in een instantie van SQL Server. Bij beide diensten is er geen onnodige onderhoud nodig zoals het bijhouden van systeemupdates en besturingssysteem beveiliging.

5.5.4 Metingen opslaan

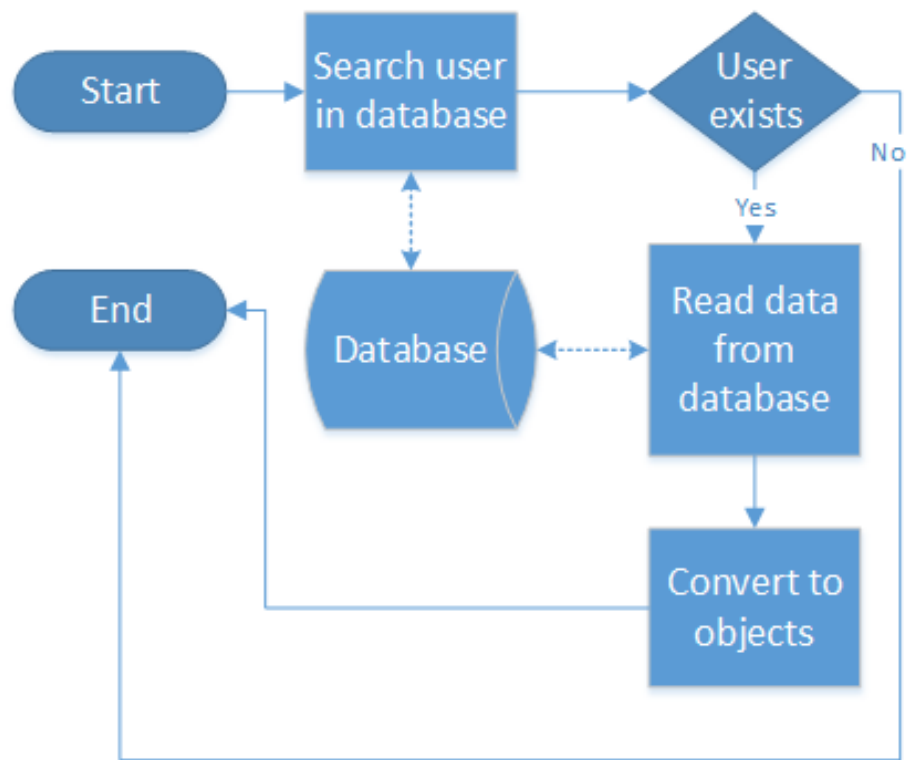
Voor het opslaan van metingen worden er JSON code verstuurd naar de server vanuit de smartphone applicatie. JSON is een formaat om objecten als leesbare tekst te versturen of op te slaan. De ontvangen informatie wordt omgezet naar een object, als dit niet lukt dan wordt er een error terug gestuurd. Wanneer de interpretatie succesvol is dan wordt er aan de hand van de ontvangen informatie gezocht naar de gebruiker die de aanvraag doet. Als de gebruiker niet bestaat wordt er ook een error gegeven. Zo niet, dan wordt de ontvangen data opgeslagen in de database en wordt er een bericht gegeven dat de operatie succesvol is. Dit hele proces is te zien in Figuur 11.



Figuur 11: Data opslaan flowchart

5.5.5 Metingen ophalen

Bij het ophalen van metingen wordt de geheime code gebruikt. Deze code wordt direct opgezocht in de database, zoals te zien is in Figuur 12. Als de gebruiker niet bestaat dan wordt er een error teruggegeven. Wanneer de gebruiker wel gevonden is dan worden alle sensordata geselecteerd van de desbetreffende gebruiker. De metingen worden samengevoegd tot objecten die vervolgens uit worden geprint als reactie.



Figuur 12: Data verkrijgen flowchart