

## Software Engineering

### ESP32 + Raspberry Pi datalogger

ESP doet metingen (temp, luchtdruk, ...) via wifi netwerk naar Raspberry pi.

De Raspberry Pi plaatst de gegevens op een webpagina met grafieken met meetgeschiedenis

#### Wat meten?

- Temperatuur
- Vochtigheid
- Licht intensiteit
- Luchtdruk

API documentatie: <https://app.swaggerhub.com/apis-docs/ldiy/datalogger/1.0>

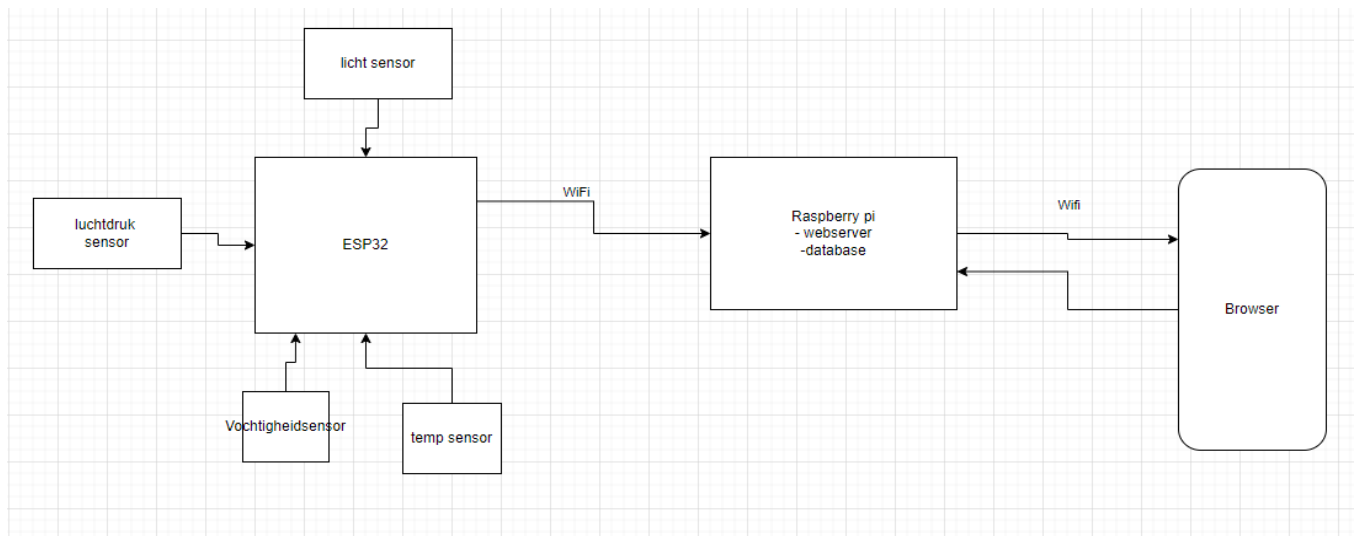
Git repo: <https://github.com/Sotware-Engineering-Datalogger/Datalogger>

Naam	GitHub username	Wat gedaan
Ben Blondeel	BenBlondeel	- Sensors uitlezen (DHT20)
Bram Dom	Theunamedcreator	- ESP32 data versturen via API - Sensors uitlezen (HP303B)
Lorenz Clijnen	ldiy	- Backend server (API, DB schema) - Server setup (Raspberry Pi) - ESP32 code samenvoegen en werkend maken (multicore)
Robin Van den Bergh	RobinVdB8	- Website design - Website functionaliteit (API communicatie)

#### Wat zit nog niet in V1.0

- Luchtdruk sensor (defect)
- Behuizing voor de sensor
- Meest recente meting weergeven bovenaan de webpagina
- Status van de sensor tonen op de webpagina
- Automatisch kijken of er nieuwe data beschikbaar is (workaround: webpagina vernieuwen)

## Blokschema



### Software onderdelen:

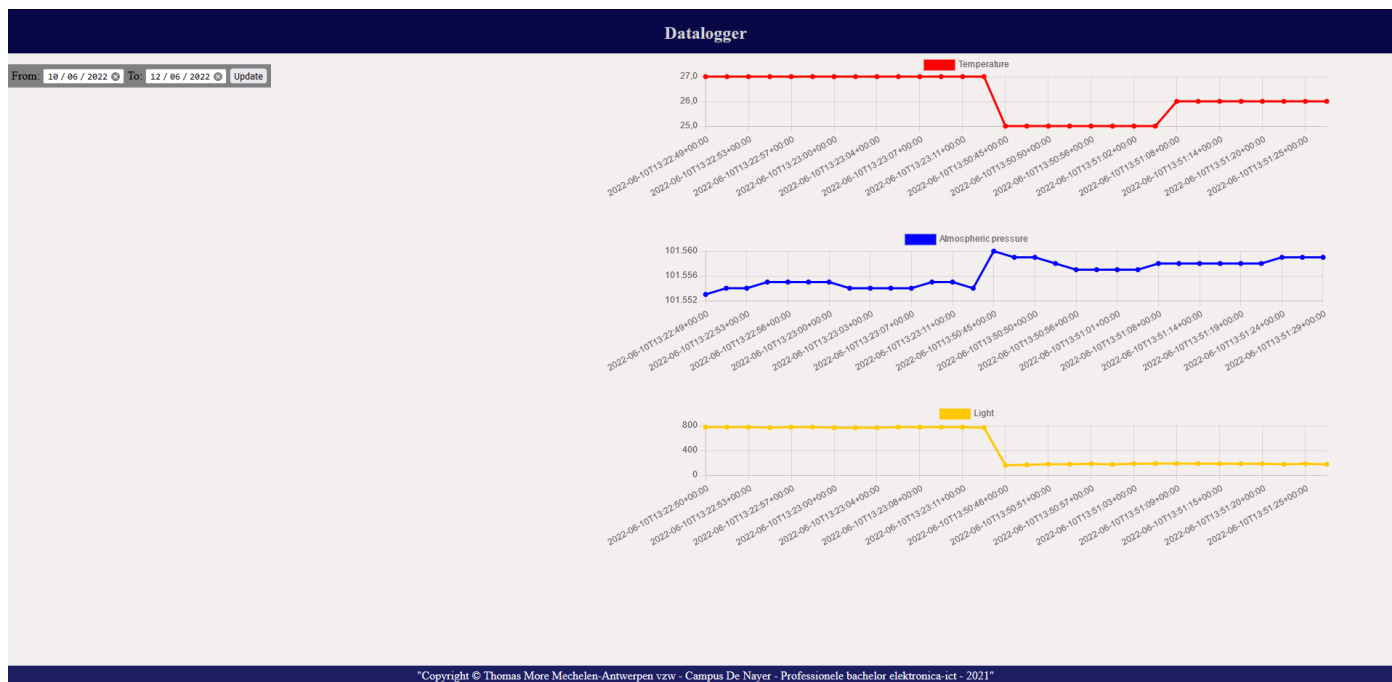
- Backend met API voor esp32 en website ( NGINX, MYSQL, PHP, Laravel)
- Website frontend ( HTML / CSS / JS)
- Esp32 HAL sensoren uitlezen (C arduino)
- ESP32 communicatie met API(C arduino)

**Sensor:** ESP32 + licht, luchtdruk, vochtigheid en temp sensor

**Server:** Rasperry Pi (met webserver en DB)

User requirements	System requirements
<ul style="list-style-type: none"> <li>• Uitlezen van luchtdruk, temperatuur, luchtvochtigheid, licht intensiteit</li> <li>• Meten van luchtdruk, temperatuur, luchtvochtigheid, licht intensiteit.</li> <li>• De datahistorie moet beschikbaar zijn via een webpagina. Weergegeven in een grafiek (metrische eenheden), tijdsbereik moet te kiezen zijn. De pagina zal verdeeld worden in 4 grafieken die elk één van de gemeten waarden visueel voorstelt met een basisinstelling van alle data. De gebruiker kan op de webpagina voor alle grafieken tegelijk de begin- en einddatum instellen.</li> <li>• Bovenaan de pagina zullen de meest recente metingen weergegeven worden.</li> <li>• Er moet een netwerk en WAP opgezet worden, de sensor zal hier</li> </ul>	<ul style="list-style-type: none"> <li>• De meter voor luchtdruk, temperatuur, luchtvochtigheid en licht intensiteit zal elke 5 minuten een meting uitvoeren.</li> <li>• De server ontvangt de data van de sensor en voegt de data in de webbrowser; dit zal altijd actief zijn want de webpagina moet altijd beschikbaar zijn.</li> <li>• Sensor stuurt de data van de meters elke 5 minuten nadat het een meting heeft uitgevoerd, naar de server.</li> <li>• De server moet de data bewaren voor minimaal 2 maanden</li> <li>• De sensor zal gevoed worden m.b.v. een USB</li> <li>• De server zal gevoed worden door het net.</li> <li>• De behuizing voor de sensor en server zal gemaakt worden met een 3D-printer.</li> <li>• Nauwkeurigheid temperatuur : +- 2°</li> </ul>

<p>mee verbinden. De webpagina ook via dit netwerk beschikbaar.</p> <ul style="list-style-type: none"><li>• Op de webpagina moet ook getoond worden of de sensor online/verbonden is.</li><li>• Plug&amp;Play: De sensor en server moeten enkel van voeding voorzien worden. Als het systeem is opgestart moeten de webpagina meteen beschikbaar zijn zonder verdere configuratie.</li></ul>	<ul style="list-style-type: none"><li>• Nauwkeurigheid vochtigheid: +- 5%</li><li>• Bereik temperatuur: 0° – 50°</li><li>• Bereik vochtigheid: 20% - 90%</li></ul>
--	--



Use case: Ingebruikstelling (Actor: gebruiker, server, sensor)

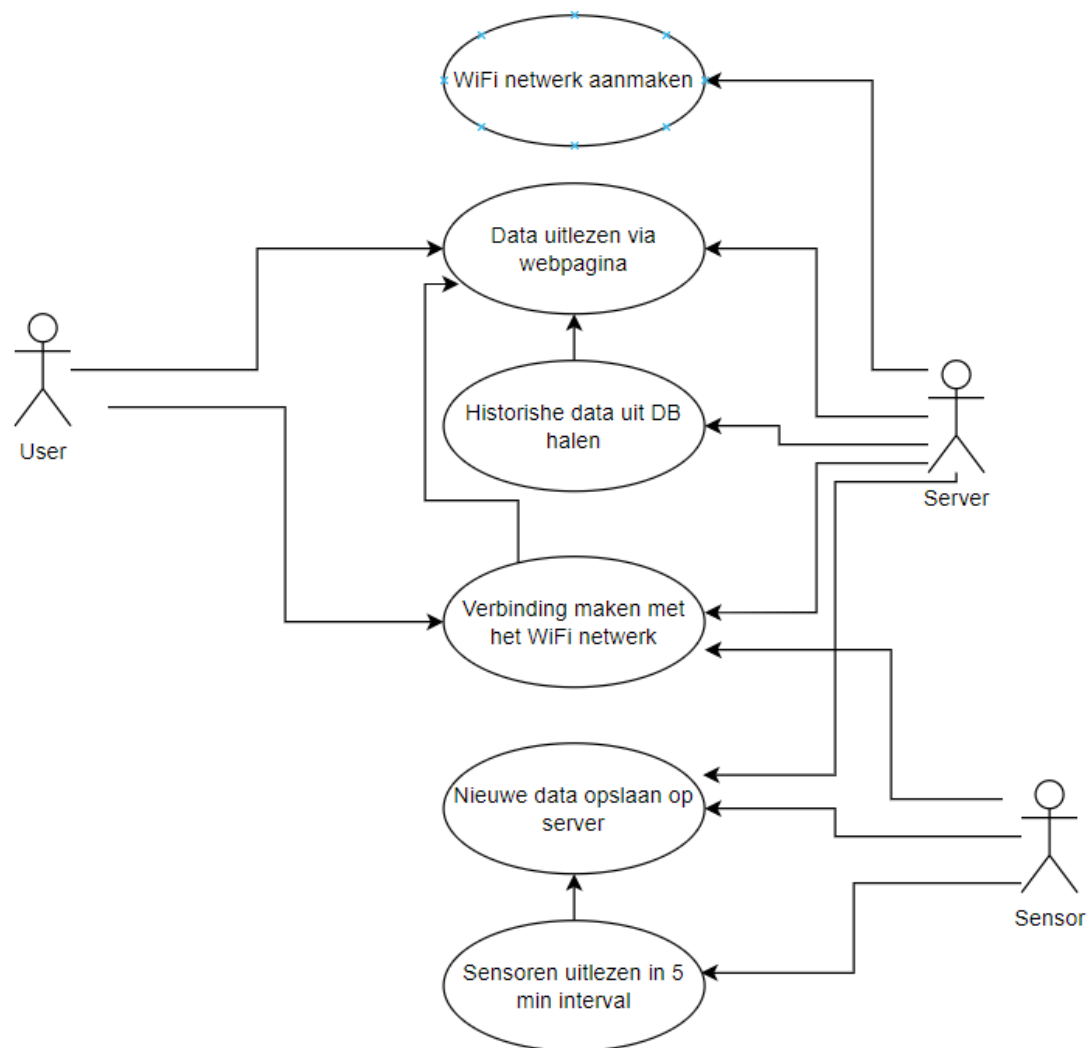
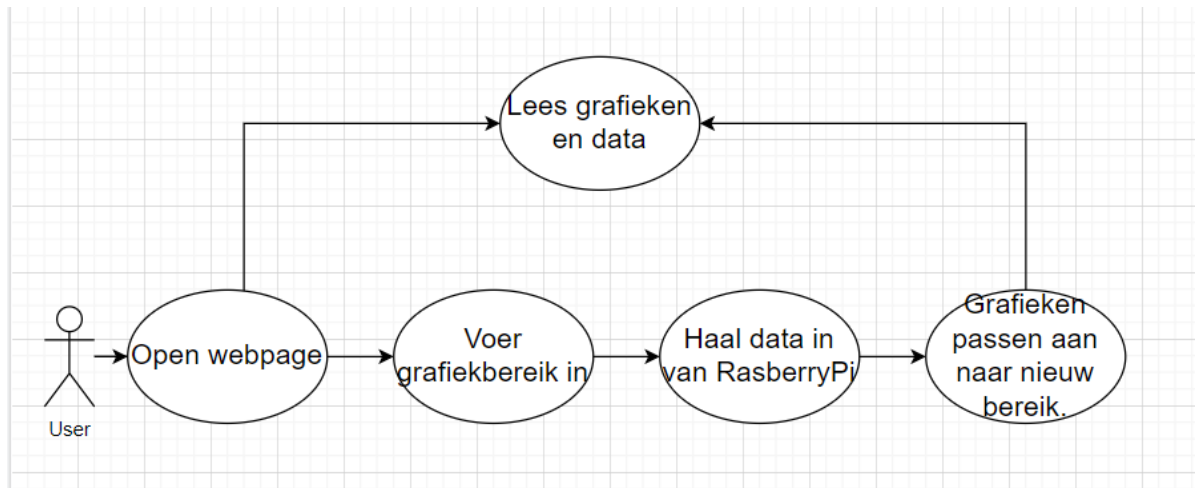
1. De gebruiker sluit de server aan op netstroom
2. De server start de software en het netwerk op
3. De gebruiker sluit de sensor aan op de USB voeding (\*1)
4. De sensor maakt verbinding met de server (\*1)
5. De gebruiker verbindt met het gecreëerde WiFi netwerk
6. De gebruiker bezoekt de webpagina en leest de sensor data uit.

Use case: \*1 De sensor is niet verbonden met de server

1. De gebruiker bezoekt de webpagina
2. De status van de sensor wordt weergegeven als Offline
3. De pagina toont de laatst gemeten waarden
4. De historische data wordt nog steeds weergegeven in de grafieken

Use case: De gebruiker kiest een tijdsbereik

1. De gebruiker kiest een begin en eind tijdstip (\*2)
2. De webpagina vraagt de data aan de server
3. De webpagina verandert de grafieken naar het gewenste bereik



## Function list

### Server

1. Lees de data in dat over het netwerk werd verstuurd.
2. Decodeer de data
3. Slaag de data op in de database
4. Verstuur laatste record uit de DB naar de webpagina
5. Verstuur records tussen start en einddatum naar de webpagina
6. Verstuur sensor status naar de webpagina
7. Maak het WiFi netwerk aan
8. Start de webserver op
9. Start de database server op

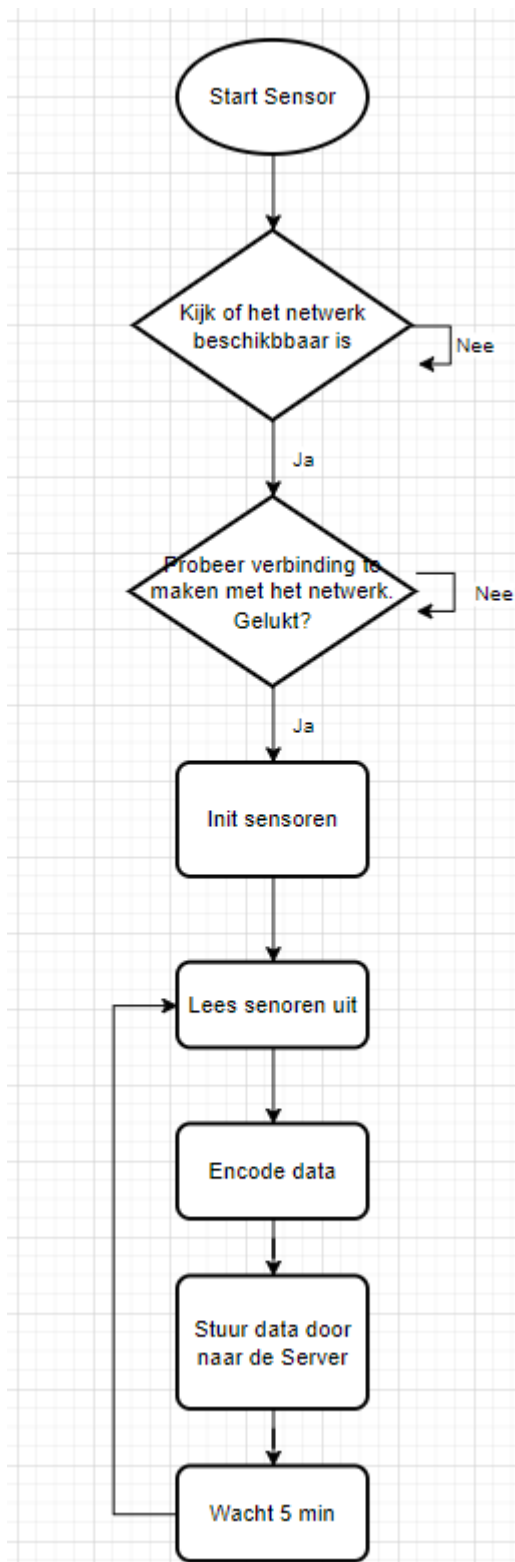
### Sensor

1. Initialiseren van de sensoren
2. Lees vochtigheidssensor uit
3. Lees temperatuur sensor uit
4. Lees lichtsensor uit
5. Lees luchtdruk sensor uit
6. Kijk of het WiFi netwerk beschikbaar is
7. Maak verbinding met het WiFi netwerk
8. Stuur data door naar de server via de API
9. Wacht/slaap voor 5min

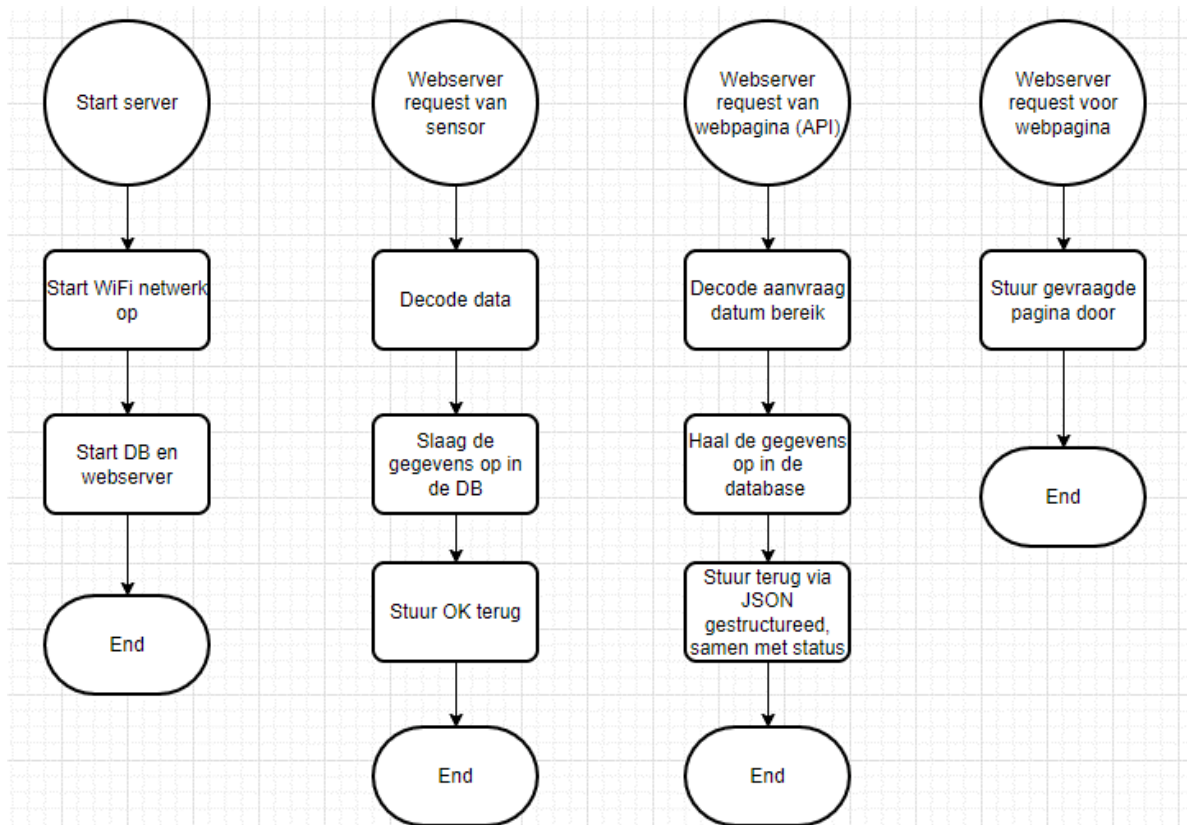
### Webpagina

1. Vraag de laatste bekende gegevens op elke x tijd (bv. 1min)
2. Vraag de gegevens van de server die aangeduid zijn binnen een bepaalde tijdszone door de gebruiker
3. Vraag de status van de sensor op
4. Parse de gegevens van de server
5. Update de grafiek met de verkregen waardes
6. Update de status met de laatste bekende gegevens

## Flowchart sensor

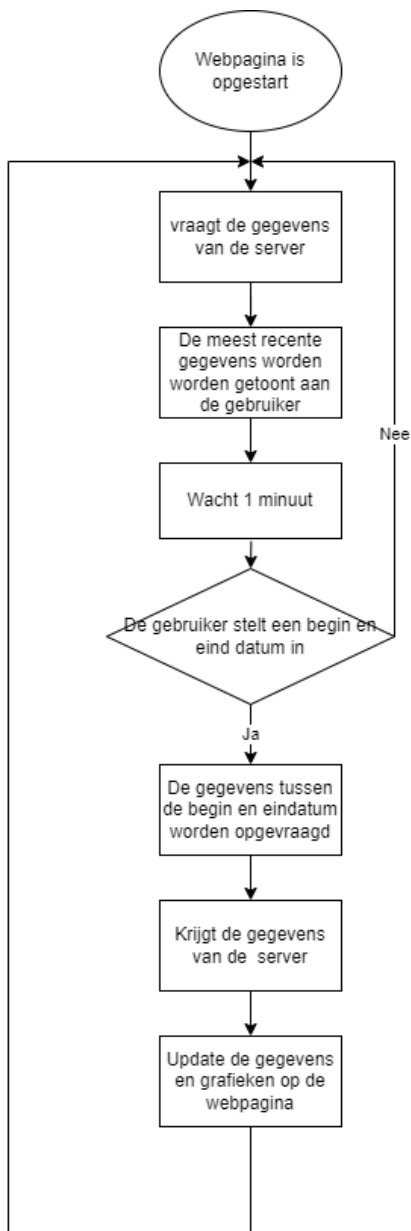


## Flowchart Server





## Flowchart webpagina



## Data storage

### Sensor

- WiFi SSID: const max 255 bytes dynamic
- WiFi Password: const max 255 bytes dynamic
- Server Adres: constdynamic
- Temperatuur: 4 bytes int static
- Vochtigheid: 4 bytes int static
- Lichtintensiteit: 4 bytes int static
- Luchtdruk: 4 bytes int static
- Response http status code: int static
- LOLIN\_HP303B object
- 

### Server

- SQL query resultaat: dynamic
- Http response code: static
- http response: dynamic
- ...

### Webpagina

- Server adres (via browser): dynamic string
- Sensor online status: static bool
- Temperatuur historisch array: dynamic float array
- Vochtigheid historisch array: dynamic float array
- Lichtintensiteit historisch array: dynamic float array
- Luchtdruk historisch array: dynamic float array
- http request: dynamic
- http response: dynamic
- start timestamp: static
- eind timestamp: static

## Prioriteitenlijst:

### Server:

#### *High:*

- Maak het WiFi netwerk aan
- Start de webserver op
- Start de database server op

#### *Medium:*

- Lees de data in dat over het netwerk werd verstuurd.
- Decodeer de data

#### *Low:*

- Slaag de data op in de database
- Verstuur laatste record uit de DB naar de webpagina
- Verstuur records tussen start en einddatum naar de webpagina
- Verstuur sensor status naar de webpagina

### Sensor:

#### *High:*

- Initialiseren van de sensoren
- Kijk of het WiFi netwerk beschikbaar is
- Maak verbinding met het WiFi netwerk
- Stuur data door naar de server via de API

#### *Medium:*

- Lees vochtigheidssensor uit
- Lees temperatuur sensor uit
- Lees lichtsensor uit
- Lees luchtdruk sensor uit

#### *Low:*

- Wacht/slaap voor 5min

### Webpagina:

#### *High:*

#### *Medium:*

- Vraag de gegevens van de server die aangeduid zijn binnen een bepaalde tijdszone door de gebruiker
- Parse de gegevens van de server

#### *Low:*

- Vraag de laatste bekende gegevens op elke x tijd (bv. 1min)
- Update de grafiek met de verkregen waarden
- Update de status met de laatste bekende gegevens

## Error Handling:

### Server:

Hard errors	
Er is geen data ontvangen van de esp (>20min)	Er wordt een melding getoond op de webpagina Dat aantoont dat er totaal geen verbinding meer is met de esp.
Kan de webserver niet opstarten (>30s)	Schrijft error naar een logfile
Kan het netwerk niet opzetten	Schrijf error naar een logfile
De database server start niet op (>30s)	De request zullen beantwoord worden met een http 500 status code Dat aantoont dat de database niet opgestart is geraakt.

### Sensor:

Soft errors	
Geen verbinding met een meetsensor.	Toon op webpagina een alert: Geen contact met sensor A
Krijgt geen antwoord van de request naar server.	Negeer, probeer pas opnieuw na nieuwe meting (5min)
Geen verbinding met het netwerk.	Probeert elke 0.5s opnieuw te verbinden.
Hard errors	
Geen voeding	Lampje aangesloten aan de voeding, als het niet brand dan weet de gebruiker dat de voeding niet is aangesloten.

### Webpagina:

Soft errors	
De webpagina kan de data van de server niet lezen.	De grafieken zullen geen data tonen
De gekozen datum zit buiten het bereik van de database	Enkel de data binnen het bereik zal doorgestuurd worden, de grafiek start vanaf het eerste beschikbare datapunt
De gekozen begindatum is later dan de gekozen einddatum.	Enkel de data binnen het bereik zal doorgestuurd worden, de grafiek stop bij het laatste beschikbare datapunt
Hard errors	
De grafiek kan niet gegenereerd worden.	Toon een lege afbeelding.

### Takendefinitie:

#### Server:

- Parse request
  - o Lees de data in dat over het netwerk werd verstuurd.
  - o Decodeer de data
- Start servers
  - o Maak het WiFi netwerk aan
  - o Start de webserver op
  - o Start de database server op
- Get data
  - o Verstuur laatste record uit de DB naar de webpagina
  - o Verstuur records tussen start en einddatum naar de webpagina
  - o Verstuur sensor status naar de webpagina
- Toevoegen nieuwe data
  - o Slaag de data op in de database

#### Sensor:

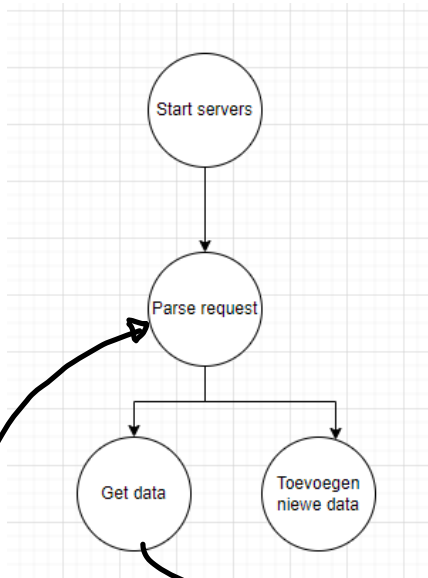
- Doe meting
  - o Lees vochtigheidssensor uit
  - o Lees temperatuur sensor uit
  - o Lees lichtsensor uit
  - o Lees luchtdruk sensor uit
- Verstuur data
  - o Stuur data door naar de server via de API
- Initialiseer
  - o Initialiseren van de sensoren
  - o Kijk of het WiFi netwerk beschikbaar is
  - o Maak verbinding met het WiFi netwerk
- Main
  - o Wacht/slaap voor 5min
  - o Doe meting
  - o Verstuur data

#### Webpagina:

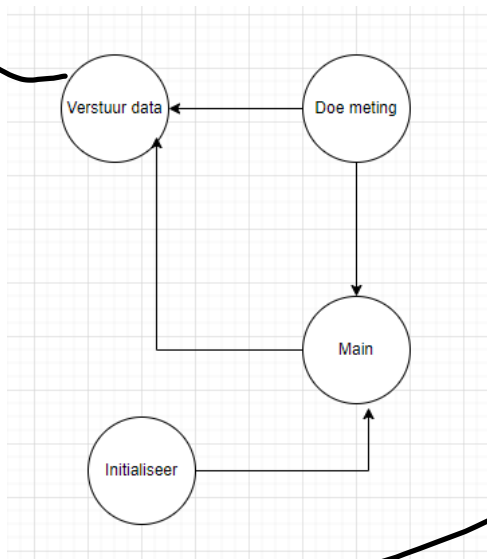
- Vraag data aan de server
  - o Vraag de laatste bekende gegevens op elke x tijd (bv. 1min)
  - o Vraag de gegevens van de server die aangeduid zijn binnen een bepaalde tijdszone door de gebruiker
  - o Vraag de status van de sensor op
  - o Parse de gegevens van de server
- Update weergave
  - o Update de grafiek met de verkregen waarden
  - o Update de status met de laatste bekende gegevens

## Flow diagram

Server:



Sensor:



Webpagina:

