



# **WeatherApp**

## Documentation technique

GUERRY Pierre-Jean  
KOWALSKI Mathias  
PENHOUE Marine  
PAIN Alyson

## Modèle 3D:



Une boîte de 7.0cm de long, 4.5cm de large et 4.4cm de haut.

La taille du boîtier permet tout juste de faire passer les composants. Ainsi, ils sont stables et ne bougent pas lorsque l'on déplace le boîtier.

Le boîtier possède 10 trous permettant le passage de l'air au sein de la boîte, et ainsi de laisser s'échapper la chaleur des composants électriques.

Une LED à l'avant servant d'indicateur sur l'état de la station:

- Clignotement jaune: recherche de wifi / portail captif activé
- Clignotement rouge: capteur défaillant
- Jaune constant: serveur non joignable
- Vert constant :tout va bien

Un port USB-C permettant de d'alimenter la station météo.

## **Choix des composants hardware:**

### **ESP-WROOM-32:**

L'ESP-WROOM-32 est un microcontrôleur qui possède un module WiFi intégré. Cela permet une connexion sans fil pour l'envoi des données météo vers le serveur en temps réel.

### **BME280:**

Le composant électronique BME280 est un capteur, qui possède un capteur barométrique précis, avec une précision de  $\pm 1$  hPa, un capteur de température avec une précision de  $\pm 1^\circ\text{C}$  et un capteur d'humidité relative avec une précision de  $\pm 3\%$ . De plus, le capteur possède une consommation de courant très faible. Enfin, ce capteur possède des interfaces de communication faciles à utiliser et à intégrer avec l'ESP-WROOM-32.

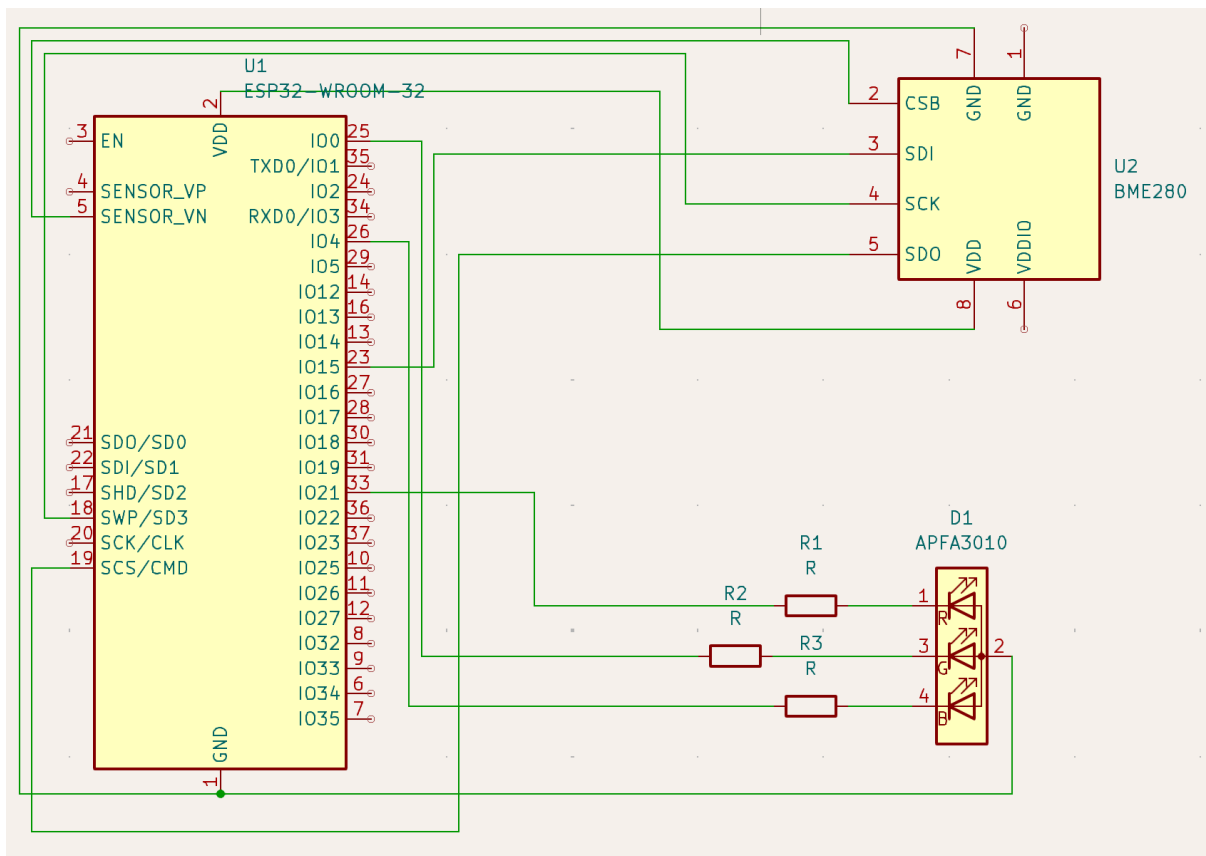
Ici nous utilisons le Serial Peripheral Interface.

### **LED (RGB):**

Cette LED permet de produire différentes lumières de différentes couleurs, cela permet d'indiquer divers états sur :

- Statut du Réseau Wi-Fi: Utiliser différentes couleurs pour indiquer la connexion ou la déconnexion du Wi-Fi.
- Statut de fonctionnement: Montrer que le système est actif, ou qu'il rencontre une erreur.

## Schéma électronique:



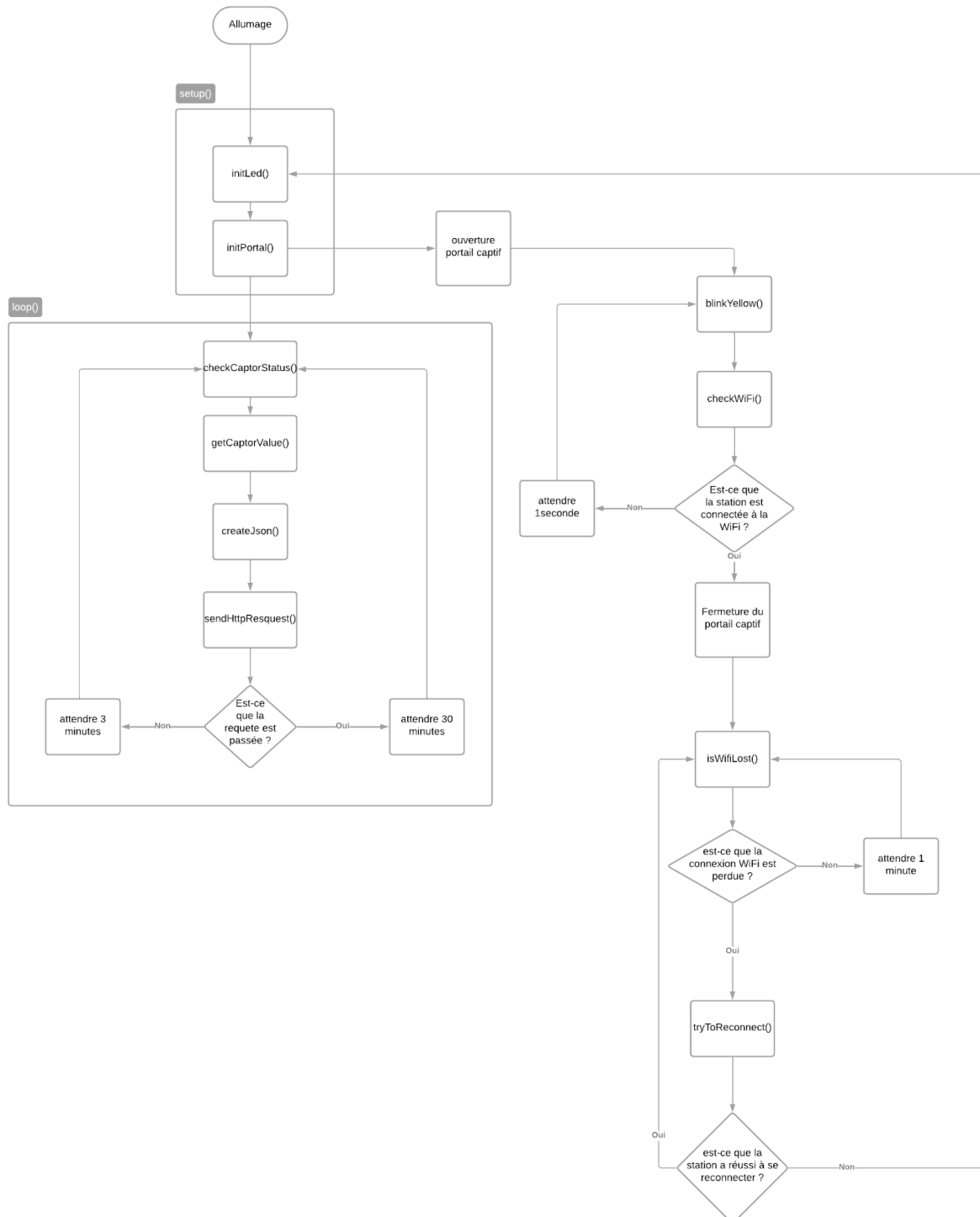
Au niveau du capteur BME280:

- la pin VDD connecté à la pin VDD de l'esp 32
- la pin GND connecté à la pin GND de l'esp 32
- la pin CSB connecté à la pin 5 de l'esp 32
- la pin SDI connecté à la pin 23 de l'esp 32
- la pin SCK connecté à la pin 18 de l'esp 32
- la pin SD0 connecté à la pin 19 de l'esp 32

Au niveau de la LED RGB:

- la pin 1 connecté à la pin 33 de l'esp 32 en passant par une résistance
- la pin 2 connecté à la pin GND de l'esp 32
- la pin 3 connecté à la pin 25 de l'esp 32 en passant par une résistance
- la pin 4 connecté à la pin 26 de l'esp 32 en passant par une résistance

## Algorithme arduino:



## Choix des technologies logiciel:

Concernant l'application mobile, la partie Frontend est développée en React Native.

Ce framework permet de développer une application qui peut fonctionner à la fois sur iOS et Android, réduisant ainsi le temps et les efforts de développement pour chaque plateforme.

De plus, il existe une vaste communauté de développeurs et de packages disponibles, ce qui facilite le développement, la maintenance et l'extension de l'application.

Pour ce qui est de la partie Backend, il est développé en PHP Symfony 7.

Ce framework est connu pour sa robustesse, sa sécurité et ses bonnes pratiques de développement. Il facilite la création d'APIs performantes et évolutives.

De plus, Symfony est basé sur un système de composants modulaires, ce qui permet de choisir uniquement les composants nécessaires au projet, garantissant ainsi une application légère et optimisée.

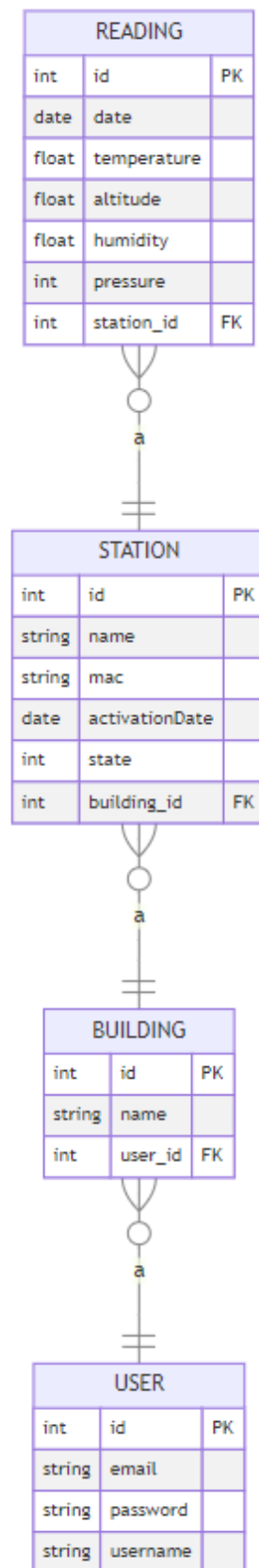
Enfin, Symfony bénéficie d'un support à long terme (LTS), assurant des mises à jour régulières et la correction de bugs pendant une période prolongée.

Pour ces deux langages, il s'agit de la spécialité de nos développeurs Frontend et de nos développeurs Backend.

Pour ce qui concerne la base de données, nous avons choisi MariaDB. C'est une base de données relationnelle, qui est un fork de MySQL. Elle ajoute des optimisations et améliorations de performances par rapport à MySQL.

Enfin, l'ORM Doctrine, que nous utilisons avec Symfony supporte MariaDB nativement.

## UML:



## Endpoints:

Pour accéder à la liste des endpoint, importer le fichier “**weatherstation-API.postman\_collection.json**” dans l’application Postman.