**Fiche Technique**

Sommaire

I -Capteurs et emplacement

1 Capteurs utilisé

2 Emplacement

II -Communication entre les capteur et serveur

1 Architecture générale

2 Module

3 Serveur

4-Base de données

III -Interface en développement

IV -Tâches restantes

II. Capteurs et emplacement

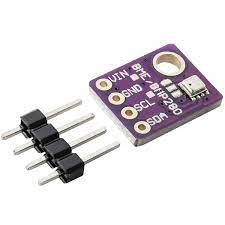
1 .Capteurs utilisé

**a. MQ135:**

Le capteur MQ-135 permet d' évaluer la qualité de l'air ambiant. Il mesure la concentration de différents gaz tels que le dioxyde de carbone (CO2), l'ammoniac (NH3), Son utilisation permet d'obtenir des données précieuses sur la santé environnementale.

**b.BME280:**

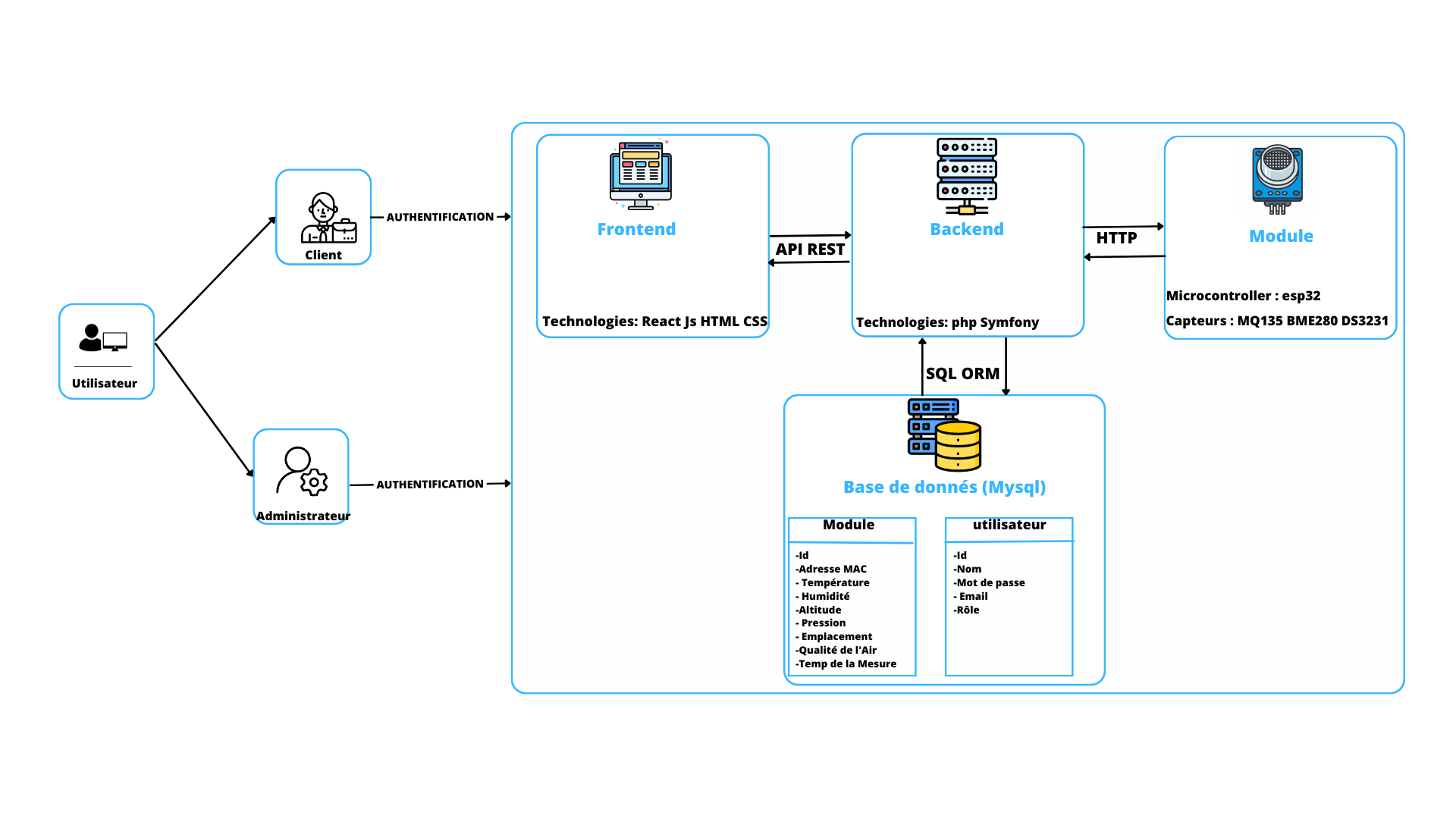
*Le BME280 est un capteur polyvalent qui possède une grande précision et qui est capable de mesurer la pression atmosphérique, la température, l'humidité et l'altitude .*



2 .Emplacement

Chaque module, composé de l'ESP 32 et des capteurs mentionnés, sera déployé dans différentes salles et zones du campus. Cette répartition permettra une collecte diversifiée de données, couvrant ainsi l'ensemble de l'environnement du campus

III. Communication



1 .Architecture générale

2 .Module

**a. Gestion du wifi:**

Le code intègre la bibliothèque WiFiManager, offrant une gestion dynamique des connexions WiFi. En cas d'échec de connexion, le dispositif bascule automatiquement en mode de configuration pour permettre la saisie des informations de connexion WiFi. Cette approche simplifie la mise en réseau des capteurs.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**b. Récupération des valeurs:**

Le code utilise les bibliothèques Adafruit pour interagir avec les capteurs. Les valeurs de température, pression, humidité et altitude du BME280, la valeur de qualité de l'air du MQ135, ainsi que la température et le temps du DS3231 sont lues. Ces valeurs sont ensuite utilisées pour calculer une température moyenne.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**-Bibliothèques utilisées**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**-Instanciation des Capteurs et Configuration Initiale**

**Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

**-Récupération des Valeurs des Capteurs**

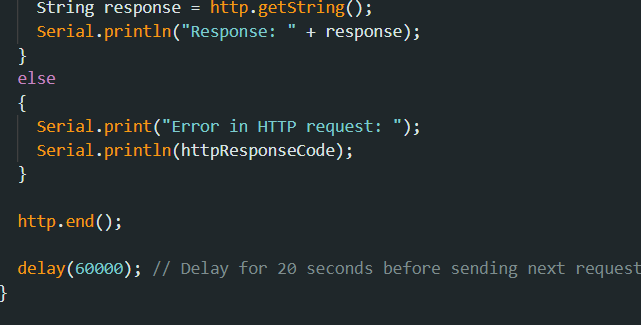
**Une image contenant texte, capture d’écran, menu, Police

Description générée automatiquement**

**c. Envoi des Données via HTTP**

Les données des capteurs sont stockées dans un objet JSON et envoyées via une requête HTTP POST vers le serveur symfony. Les valeurs de chaque capteur, ainsi que l'adresse MAC du dispositif, sont incluses dans le corps de la requête

**Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement**

3 .Serveur

Le contrôleur Symfony joue un rôle essentiel dans la gestion des données du capteur. Il récupère les informations au format JSON envoyées par le capteur, crée une nouvelle entrée dans la base de données avec ces données, confirme que l'enregistrement s'est déroulé avec succès et vérifie si l’utilisateur est lié au module.

**Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement**

**Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement**

4 .Base des données

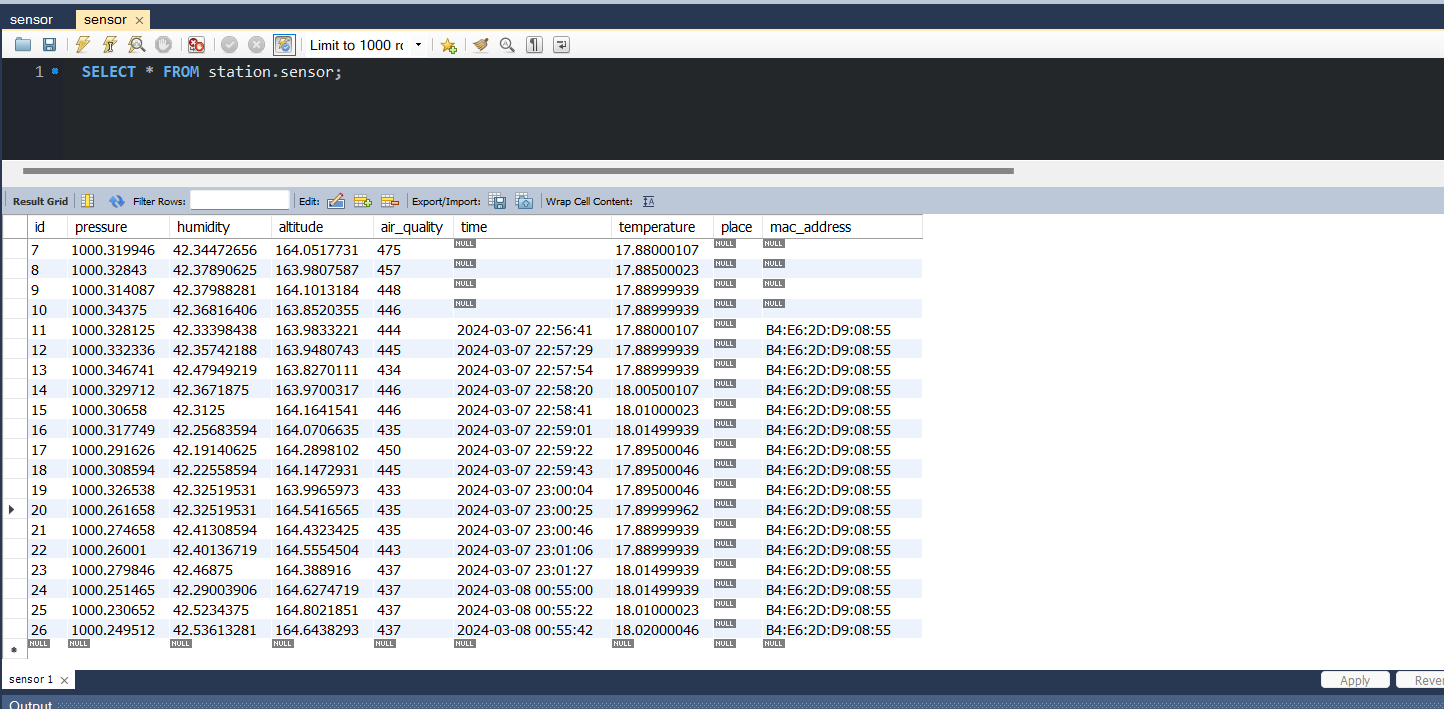
L'entité Symfony “Sensor” est créée pour stocker les informations dans la base de données. Elle comporte des attributs tels que pression, humidité, altitude, qualité de l'air, date et heure, température, emplacement, et adresse MAC du capteur. Ces attributs correspondent aux données collectées par les capteurs.

Une image contenant texte, capture d’écran

Description générée automatiquement

**Une image contenant texte, capture d’écran, Police

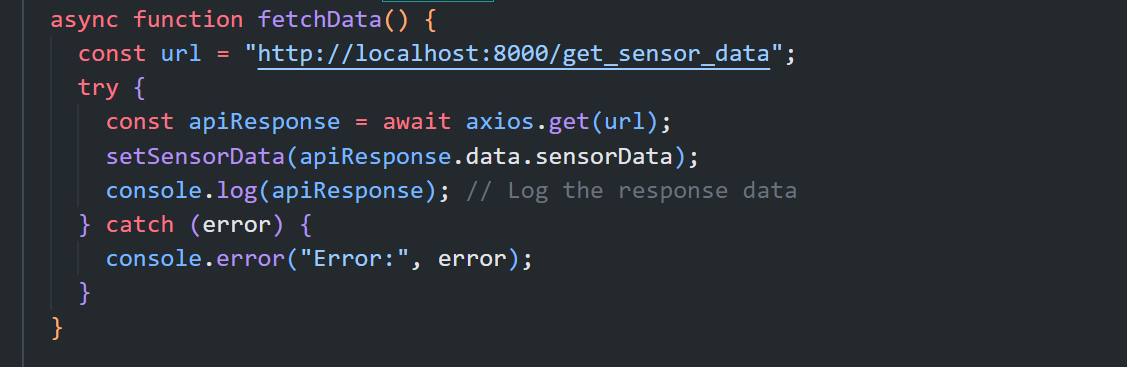
Description générée automatiquement**

**Schéma de la base de données** 

IV. Interface

Pour l'interface de la Station Météo nous avons utilisé React pour son efficacité et sa réactivité. Actuellement, une fonction asynchrone a été développée dans React qui récupère les données du capteur via une requête HTTP à l'URL

Cette fonction utilise Axios pour gérer les requêtes de manière asynchrone et met à jour les données des capteurs dans l'application.



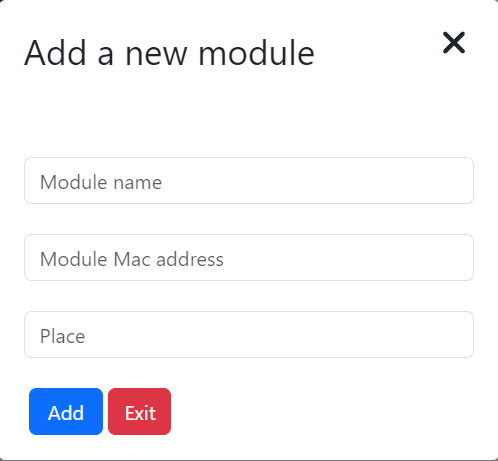
Côté Symfony, une fonction a été implémentée pour répondre à la requête HTTP de l'interface React. Cette fonction utilise Doctrine pour interagir avec la base de données, récupère toutes les entrées du capteur, les sérialise en un format adapté à la consommation par React, et renvoie les données sous forme de réponse JSON

**Exemple d’affichage des donnés en react**

**Une image contenant texte, capture d’écran, Police, logo

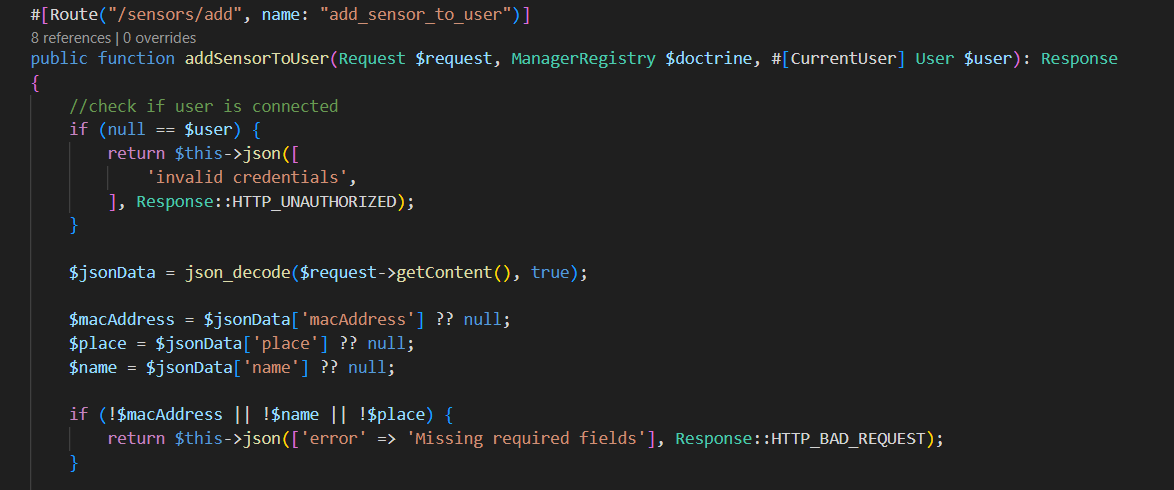
Description générée automatiquement**

Lorsqu'un utilisateur est connecté, il peut facilement améliorer la sécurité en ajoutant un module avec la spécification de son emplacement via l'adresse MAC qui serait intégrée dans l'impression 3D de chaque module. Cela garantit que chaque utilisateur ne peut visualiser que les données de son propre module



Gestion des modules

Pour gérer les modules, nous utilisons les fonctions suivantes dans SensorController :

Cette fonction permet donc lié un module à un utilisateur

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

Cette fonction permet de séparer un utilisateur d’un module: Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Cette fonction permet d’afficher toutes les modules d’un utilisateur donné : Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Cette fonction permet d’afficher la dernière valeur récupérée par un module donné lié à l’utilisateur actuellement connecté: Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Cette fonction permet d’afficher toutes les valeurs obtenues par un capteur donné lié à l’utilisateur actuellement connecté: Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

Gestion des Comptes Utilisateurs

Pour gérer les comptes utilisateurs, nous avons créé un objet utilisateur :

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Celui-ci est ensuite utilisé dans les fichiers SecurityController et UserController. SecurityController sert à connecter un utilisateur déjà inscrit ainsi qu’à se déconnecter : Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

UserController permet quant à lui de créer un utilisateur, mais aussi de vérifier que le compte est déjà existant ainsi qu’a lister les utilisateurs :

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Enfin cette fonction permet de récupérer les information de l’utilisateur connecté : Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

2 .Développement Frontend en React

Finaliser le développement du frontend en React et Améliorer l'expérience utilisateur