**Fiche Technique**

Sommaire

I -Capteurs et emplacement

1 Capteurs utilisé

2 Emplacement

II -Infrastructure et Gestion Système

1 Architecture générale

2 Module

3 Serveur

4-Base de données

5-Interface graphique

II. Capteurs et emplacement

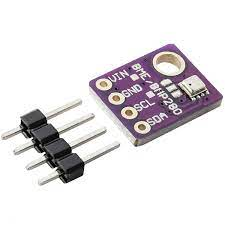
1 .Capteurs utilisé

**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 .*

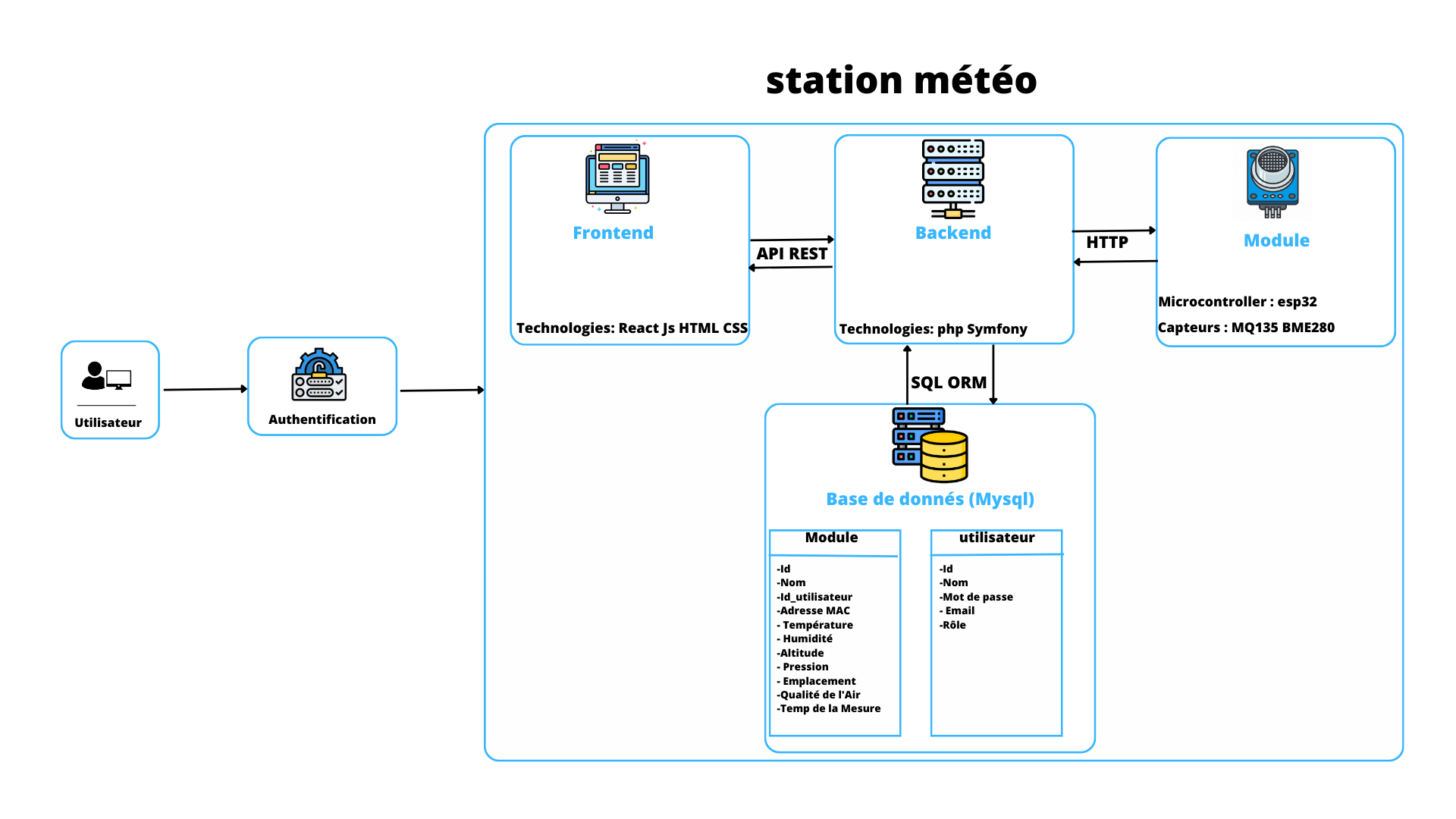


2 .Emplacement

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. Infrastructure et Gestion Système

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 et la valeur de qualité de l'air du MQ135

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**

**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

**a. Enregistrer les données du capteur**

La fonction handelSensorData permet d'enregistrer les données d'un nouveau capteur dans la base de données. Elle commence par récupérer et valider les données JSON. Ensuite, elle vérifie si un capteur avec la même adresse MAC et le même nom existe déjà. Si tel est le cas, elle met à jour les données du capteur existant avec les nouvelles valeurs, sinon elle crée une nouvelle entrée pour ce capteur. Elle associe également l'utilisateur et le lieu au capteur si ces informations sont disponibles. Enfin, elle persiste les données dans la base de données.

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

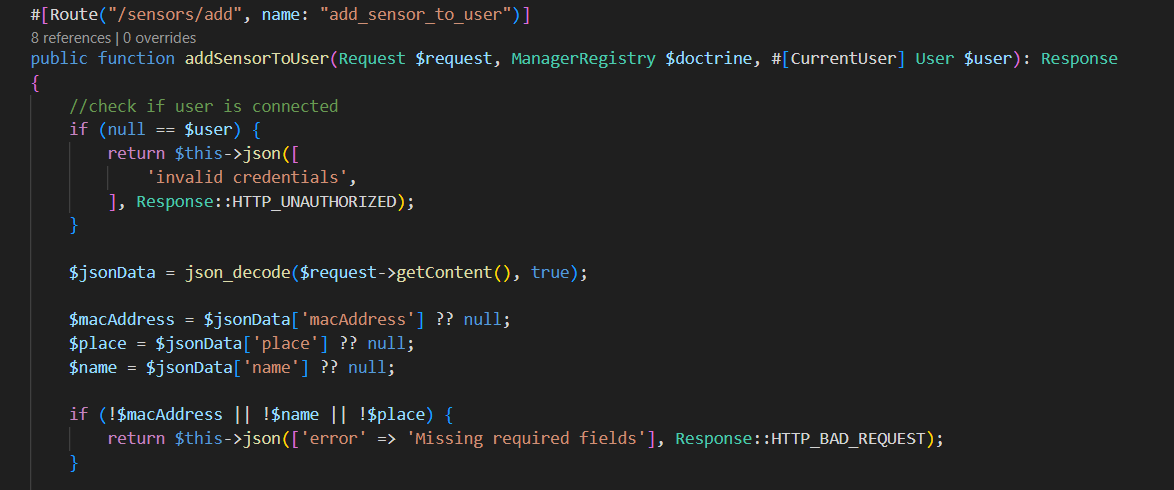
Description générée automatiquement**

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

Description générée automatiquement**

**b. Ajouter un capteur à un utilisateur**

La fonction addSensorToUser permet d'associer un capteur existant à l'utilisateur actuellement connecté. Elle récupère les données JSON de la requête pour obtenir l'adresse MAC du capteur, le nom du capteur et le lieu auquel il est associé. Si ces champs sont complets, elle recherche tous les capteurs correspondant à l'adresse MAC et au nom fournis , associe l'utilisateur et le lieu aux capteurs trouvés puis persister ces modifications dans la base de données



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

Description générée automatiquement

**c. Dissocier les capteurs d'un utilisateur**

La fonction unlinkSensorsFromUser permet de supprimer l'association d'un utilisateur avec tous les capteurs ayant une adresse MAC spécifique.Elle récupère l'adresse MAC du capteur à dissocier à partir des données JSON de la requête. Elle recherche ensuite tous les capteurs ayant cette adresse MAC dans la base de données. Pour chaque capteur trouvé, elle supprime l'association avec l'utilisateur et le lieu et persiste ces modifications dans la base de données

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

**d. Obtenir les capteurs de l'utilisateur**

La fonction getUserSensors récupère tous les capteurs associés à l'utilisateur actuellement connecté. Elle commence par vérifier si l'utilisateur est authentifié. Ensuite, elle interroge le référentiel des capteurs pour obtenir tous les capteurs associés à cet utilisateur puis retourne ces données structurées sous forme de réponse HTTP JSON.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

**f. Détails du dernier enregistrement du capteur**

La fonction getLastSensorDetails récupère les détails du dernier enregistrement d'un capteur spécifique pour l'utilisateur actuellement connecté. Elle commence par vérifier si l'utilisateur est authentifié. Ensuite, elle récupère l'adresse MAC et le nom du capteur à partir des paramètres de requête. En interrogeant la base de données, elle récupère les détails du dernier enregistrement de ce capteur

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

**g. Détails de tous les enregistrements des capteurs**

La fonction getAllSensorDetails récupère tous les détails des enregistrements de capteurs pour l'utilisateur actuellement connecté et une adresse MAC spécifique. Elle interroge le référentiel des capteurs pour récupérer tous les enregistrements de capteurs associés à cet utilisateur et à l'adresse MAC spécifiée. Elle structure les données des capteurspuis retourne ces détails sous forme de réponse HTTP JSON.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

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

Description générée automatiquement

**h. Gestion des 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

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.

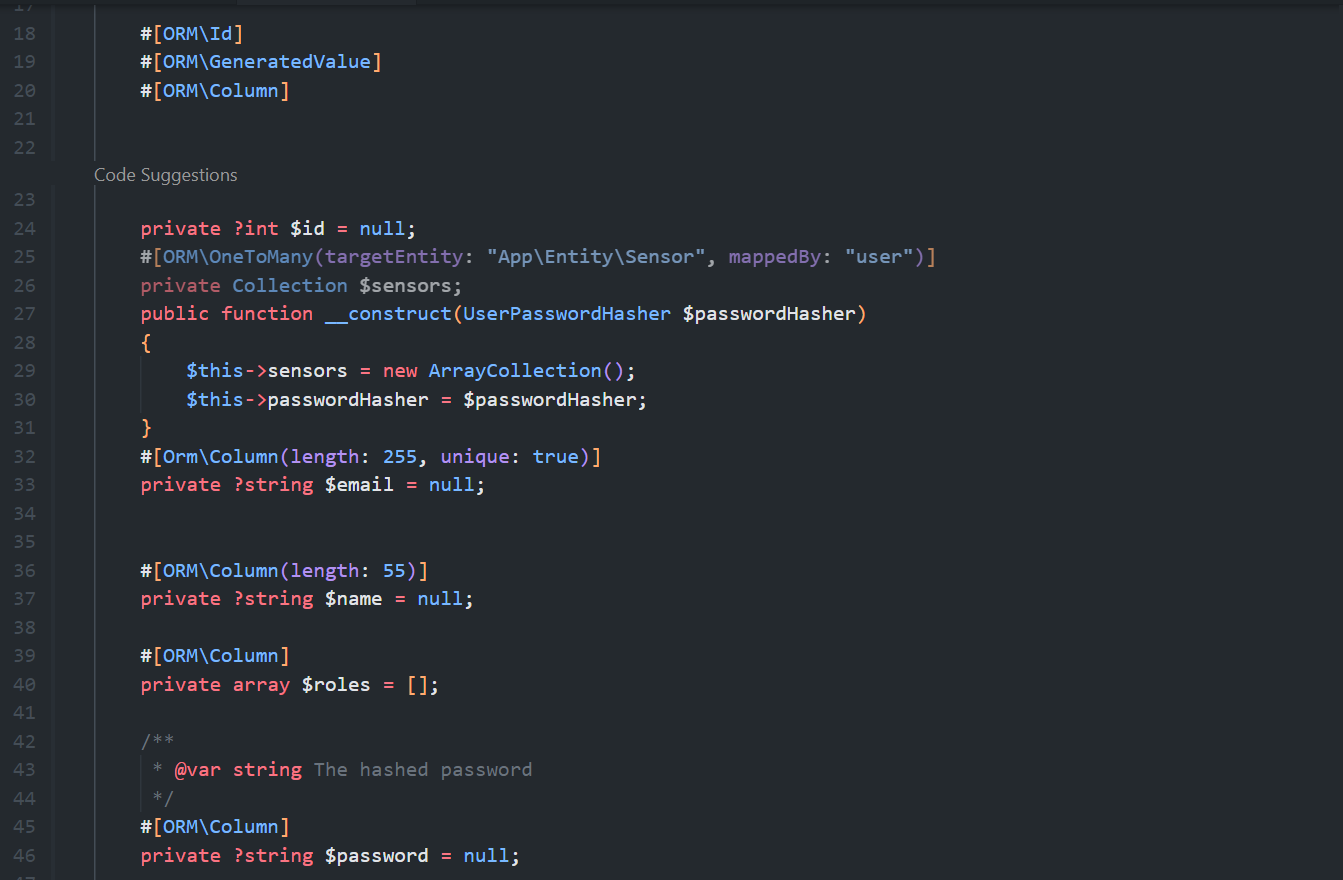
Une image contenant texte, capture d’écran

Description générée automatiquement

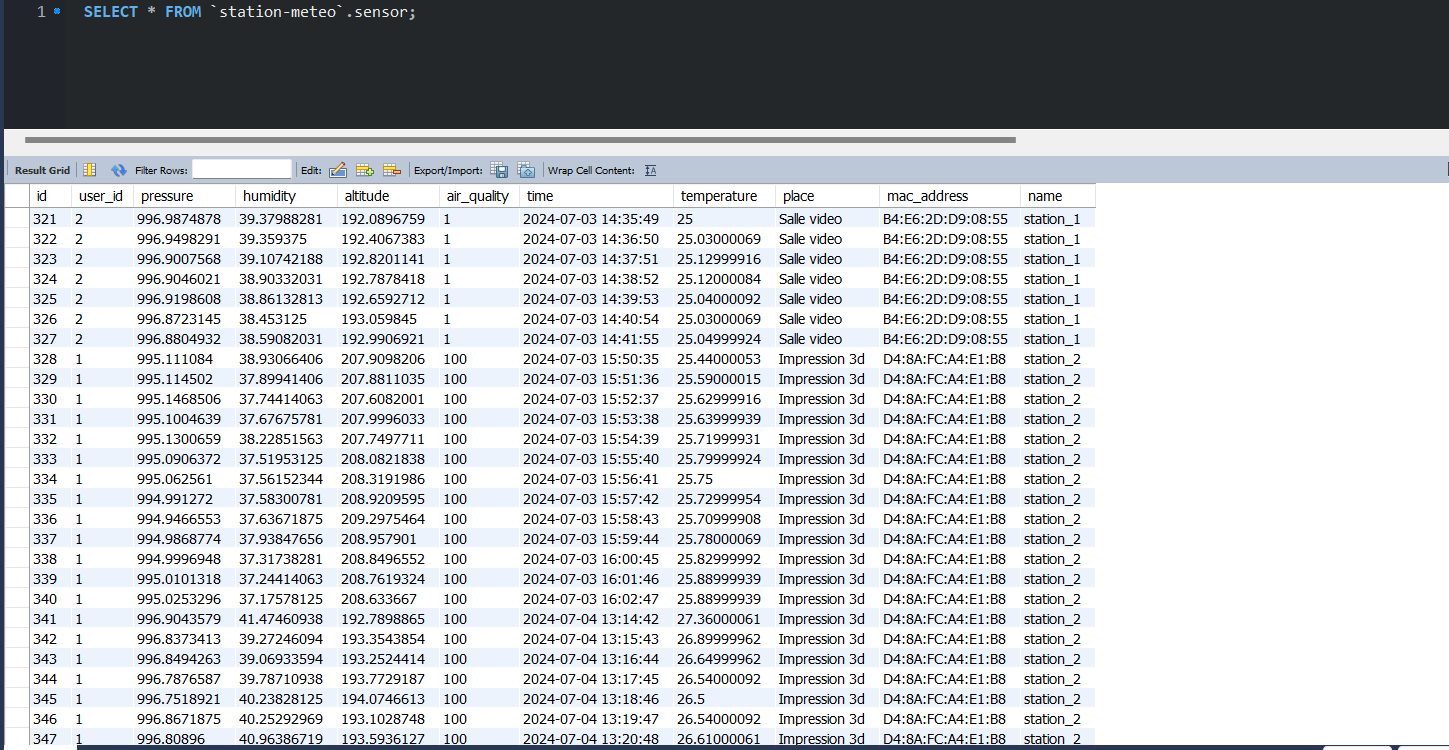
**Une image contenant texte, capture d’écran, Police

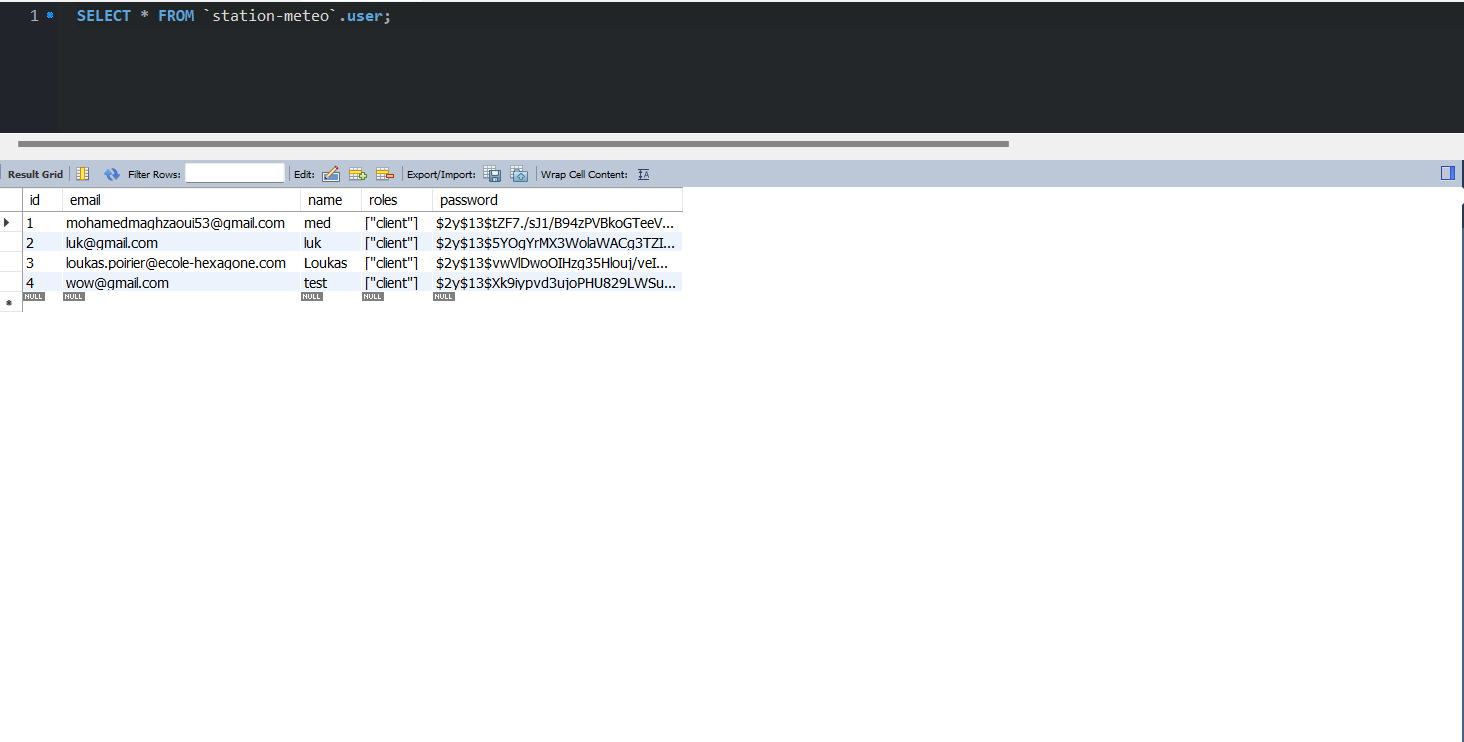
Description générée automatiquement**

L'entité Symfony User est créée pour stocker les informations des utilisateurs dans la base de données. Elle comporte des attributs tels que le nom, l'adresse e-mail, les rôles, le mot de passe, et les capteurs associés à l'utilisateur. Ces attributs permettent de gérer les informations d'identification et les autorisations des utilisateurs, ainsi que de lier chaque utilisateur aux données collectées par leurs capteurs respectifs.



**Schéma de la base de données**

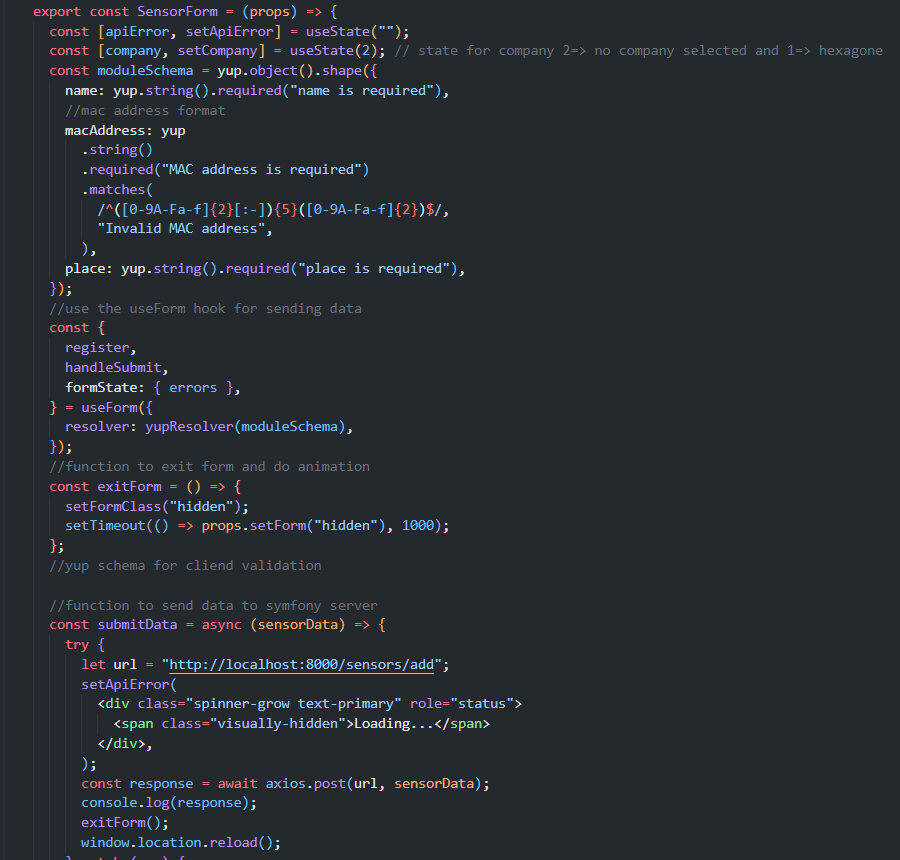
****

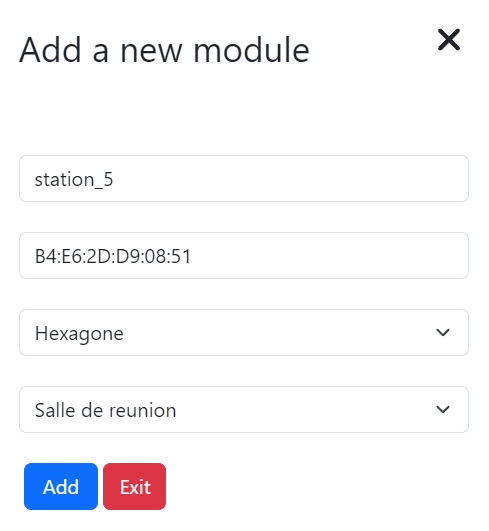


5 .Interface Graphique

**a. Formulaire de Capteur**

Le composant SensorForm permet à l'utilisateur d'ajouter un nouveau capteur. Il utilise le hook useForm pour gérer le formulaire et yup pour valider les entrées de l'utilisateur, comme le nom, l'adresse MAC et l'emplacement. Les données sont envoyées au serveur Symfony, et l'utilisateur peut sélectionner une entreprise et un emplacement associé.

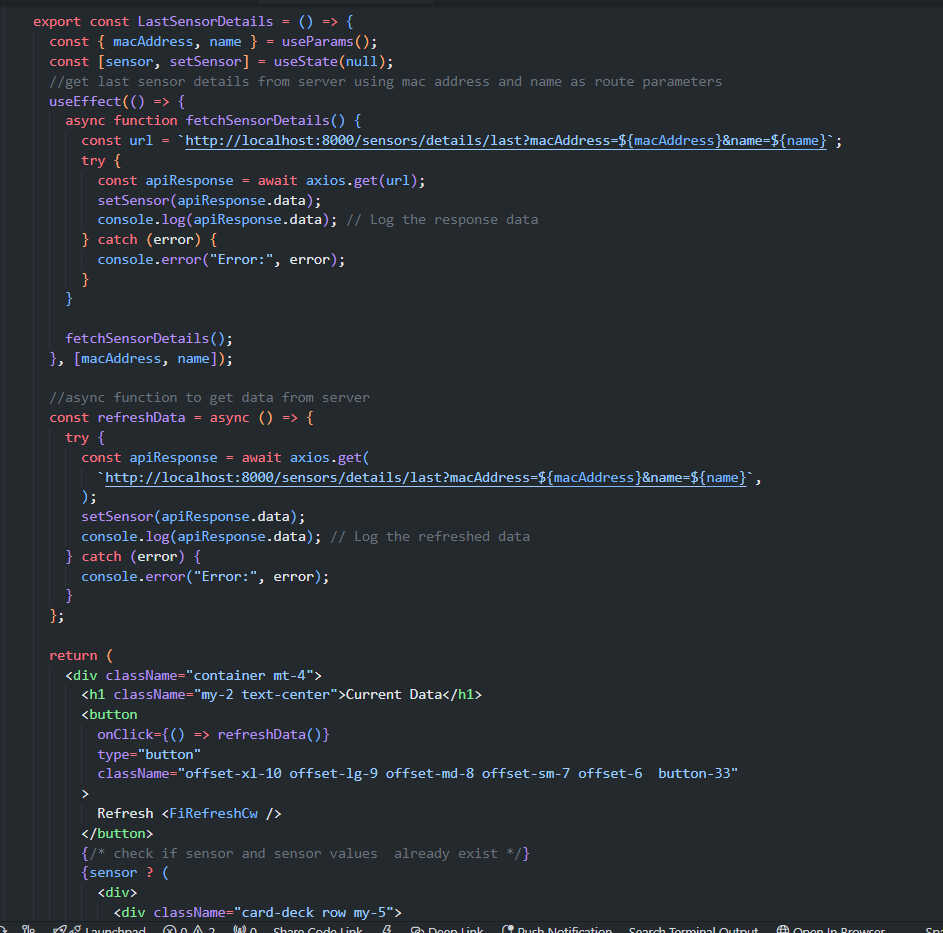




**Formulaire pour ajouter un module**

**b. Détails du Dernier Capteur**

Le composant LastSensorDetails récupère et affiche les dernières données d'un capteur spécifique, telles que la pression, l'humidité, l'altitude, la qualité de l'air, la température et l'heure. Il permet également de rafraîchir les données en les récupérant à nouveau depuis le serveur.



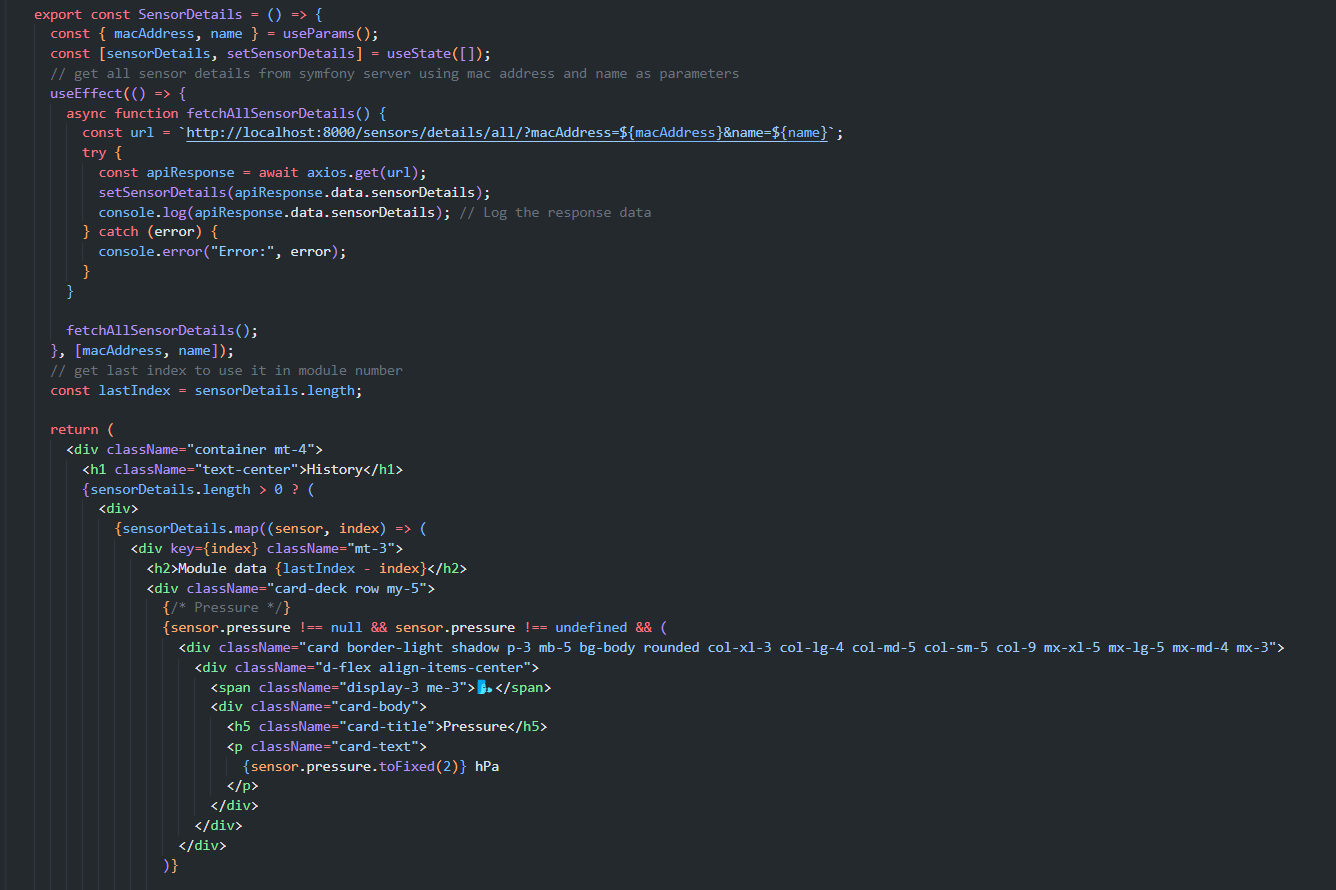
**Une image contenant texte, capture d’écran, Police, logo

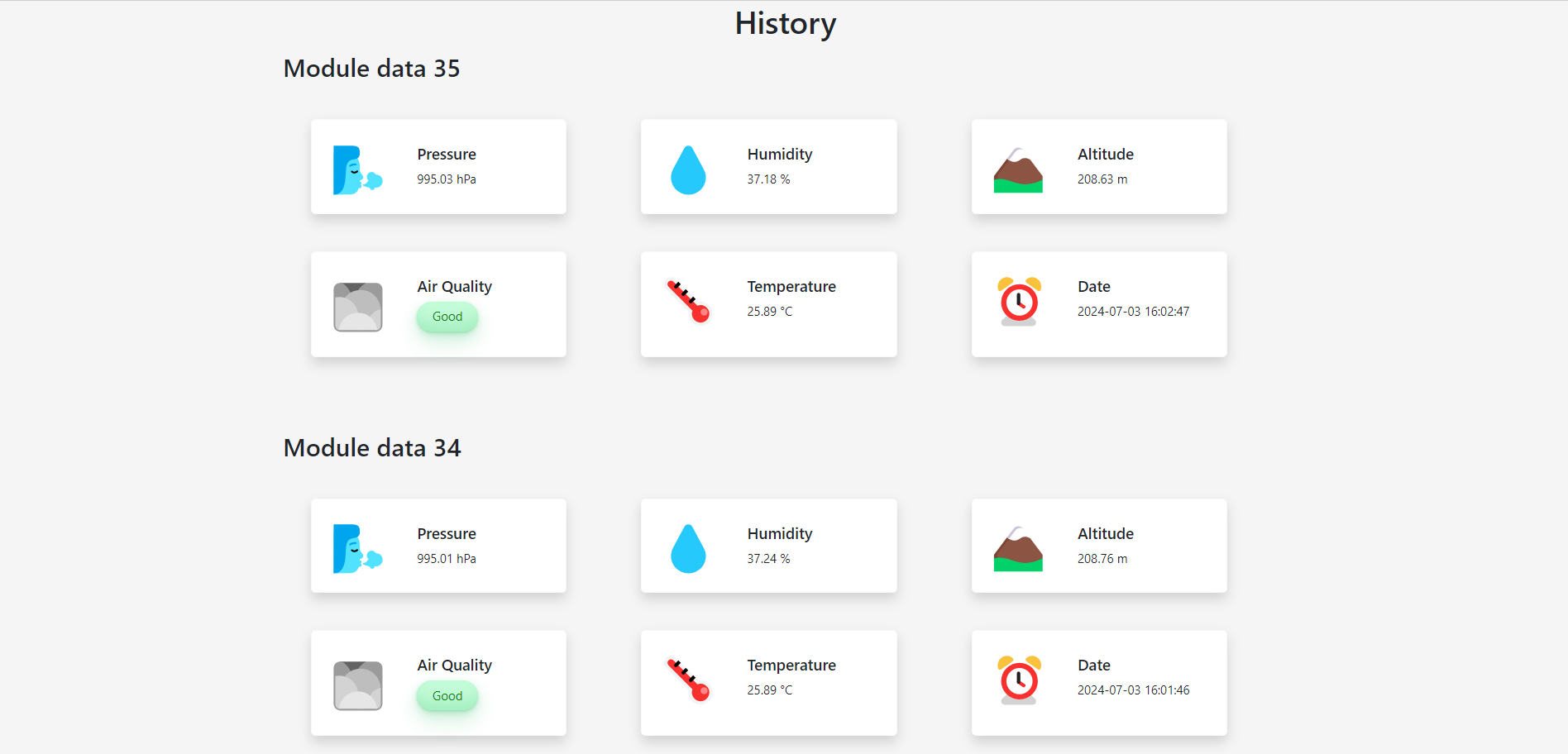
Description générée automatiquement**

**Affichage des dernières données**

**c. Historique des Données de Capteur**

Le composant SensorDetails affiche l'historique complet des données pour un capteur donné, en récupérant toutes les valeurs stockées dans le serveur Symfony. Les informations affichées incluent la pression, l'humidité, l'altitude, la qualité de l'air, la température et l'heure pour chaque enregistrement.

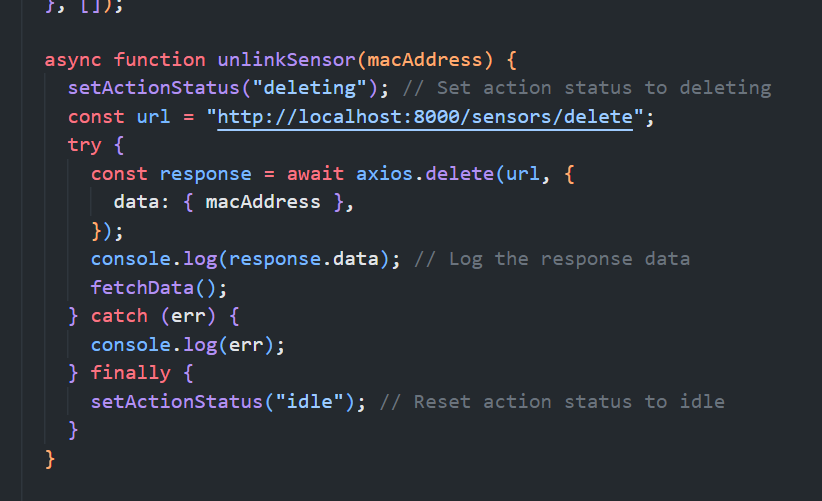


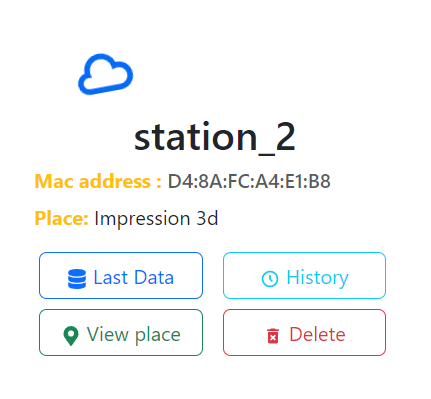


**Affichage des tous les données**

**d. Gestion de la Suppression de Capteur**

La fonction unlinkSensor est une fonction asynchrone qui gère la suppression d'un capteur en envoyant une requête HTTP DELETE



****

**Gestion des module**

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