



Rapport Commun

Projet Eau Pure

Rédacteurs : M. CONAN Nolan, M. LE SOURNE Mathys, M. SALOU Alexis

Table des matières :

1. Introduction.....	7
1.1. Contexte du projet.....	7
1.2. Analyse des besoins.....	8
2. Étudiant n°1 - Mathys LE SOURNE.....	9
2.1. Description du travail demandé.....	9
2.2. Tâche à réaliser.....	9
2.3. Réalisation d'un carnet de produit.....	10
2.4. Tâches réalisées.....	10
2.4.1. Le raspberry.....	10
2.4.2. Réalisation bouchon de test des capteurs.....	11
2.4.3. Réalisation bouchon de test prélevageur.....	11
2.5. Partie réseau.....	12
2.5.1. Plan réseau.....	12
2.5.2. Table d'adressage.....	13
3. Etudiant n°2 - Alexis SALOU.....	14
3.1. Description du travail demandé.....	14
3.1.1. Synoptique du travail demandé.....	14
3.1.2. Schéma du travail demandé :.....	14
3.2. Contexte :.....	14
3.3. Diagramme du cahier des charges :.....	15
3.4. Listes des tâches :.....	15
3.5. listes des Tâches réalisé :.....	15
3.6. Planning :.....	16
3.7. Outils utilisés :.....	18
3.7.1. GitHub :.....	18
3.7.2. PhpMyAdmin :.....	18
3.7.3. Docker :.....	18
3.8. Spécification techniques :.....	19
3.9. Tests effectués / validation :.....	20
4. Etudiant n°3 - Nolan CONAN.....	22
4.1. Description du travail demandé.....	22
4.1.1. Synoptique du travail demandé.....	22
4.1.2. Schéma du travail demandé.....	22
4.2. Contexte.....	23
4.3. Diagramme du cahier des charges.....	23
4.4. Liste des tâches.....	25
4.5. Planning.....	25
4.6. Travail réalisé.....	25
4.7. Les outils utilisés.....	27
4.7.1. Github.....	27
4.7.2. PHP.....	27

4.7.3. MySQL.....	28
4.7.4. Docker.....	28
4.7.5. Leaflet.....	28
5. Annexe.....	30
5.1. Etudiant n°1.....	30
5.2. Etudiant n°2.....	30
5.3. Etudiant n°3.....	31

1. Introduction

1.1. Contexte du projet

Le projet vise à automatiser la surveillance des eaux de surface et à diffuser en ligne des bulletins interactifs portant sur la qualité et la quantité de ces eaux. L'objectif principal est de contribuer à l'amélioration de la biodiversité et de la qualité de vie en fournissant un suivi précis et en temps réel des rivières de la région.

Ce suivi est assuré par le **Service Biodiversité Eau et Paysage (SBEP)**, un service rattaché à la **Direction Régionale de l'Environnement, de l'Aménagement et du Logement (DREAL)**. Le SBEP a pour mission de mettre en œuvre, sous l'autorité du préfet de région, les politiques publiques liées à la transition écologique.

Pour répondre à ces enjeux, le SBEP a déployé un réseau de **stations hydrologiques connectées** le long des cours d'eau. Chaque station est équipée des éléments suivants :

- Un système embarqué **Raspberry Pi** pour le pilotage et le traitement local.
- Deux capteurs : un **limnimètre** (niveau d'eau) et un **pluviomètre** (mesure des précipitations).
- Un **préleveur automatique** d'eau (actionneur).
- Un module de communication **LoRaWan** pour la transmission des données.
- Une source d'énergie **autonome** (ex : panneau solaire, batterie).

Ces stations, reliées à une **plateforme IoT**, forment un réseau de capteurs communicants via **LoRaWan**, un protocole sans fil à faible consommation d'énergie. Les données collectées sont transmises aux serveurs du SBEP via des passerelles LoRaWan connectées à Internet.

Les stations automatisent également le **prélèvement d'échantillons d'eau** en fonction des conditions hydrologiques et météorologiques (par exemple, lors d'une crue ou de fortes pluies). À chaque prélèvement, une **alerte par SMS** est envoyée à un technicien du SBEP, chargé de récupérer l'échantillon pour analyse en laboratoire.

Les résultats des analyses chimiques sont ensuite stockés dans une **base de données centralisée** et intégrés aux bulletins interactifs mis en ligne, permettant ainsi un suivi transparent et accessible de la qualité des eaux de surface.

Le SBEP occupe un bâtiment, dans ce bâtiment il y a 4 unités et chacune à son propre bureau. Il y a l'unité **pilotage et support** qui contient 3 personnes, l'unité **biodiversité** qui contient 6 personnes, l'unité **Natura 2000** qui compte 8 personnes et l'unité **eau** qui a un effectif de 10 personnes.

1.2. Analyse des besoins

Pour répondre aux exigences de la directive-cadre européenne sur l'eau (**2000/60/CE**), qui impose le respect de normes de qualité environnementale pour les masses d'eau, le SBEP a besoin d'un **système automatisé** capable de surveiller en continu la **qualité physico-chimique** des cours d'eau. Actuellement, les prélèvements d'échantillons d'eau sont réalisés manuellement et à intervalles réguliers par les techniciens, ce qui mobilise du temps et des ressources humaines. Le besoin est donc de **moderniser et automatiser ces tâches**, notamment en fonction des conditions météorologiques (pluie, crues) pour garantir des prélèvements pertinents. En parallèle, il est nécessaire d'assurer la **transmission en temps réel** des données mesurées (hauteur d'eau, pluviométrie) vers le SBEP, et de **diffuser ces données sur des plateformes publiques** (comme data.gouv.fr et le portail du SIE) pour assurer la transparence, la réutilisation nationale, et la gestion des risques (ex. Vigicrues). Enfin, le système doit aussi permettre la **signalisation immédiate des prélèvements** pour intervention humaine, et s'intégrer efficacement au **réseau informatique du SBEP**, tout en étant peu coûteux en maintenance grâce à une prise d'échantillon conditionnelle et intelligente.

2. Étudiant n°1 - Mathys LE SOURNE

2.1. Description du travail demandé

Dans ce projet, mon rôle est de raccorder sur une carte raspberry deux capteurs (un capteur limnimètre et un capteur pluviomètre) afin de réaliser une station hydrologique. Dans un second temps je dois pouvoir lire les données des capteurs, les transformer si besoin et les transmettre dans une base de données via une passerelle LoRaWan.

Une autre partie est également à faire, je dois mettre en place une infrastructure réseau afin de faire communiquer l'intégralité des équipements et des services (station hydrologique, service SBEP et les serveurs WEB et BDD).

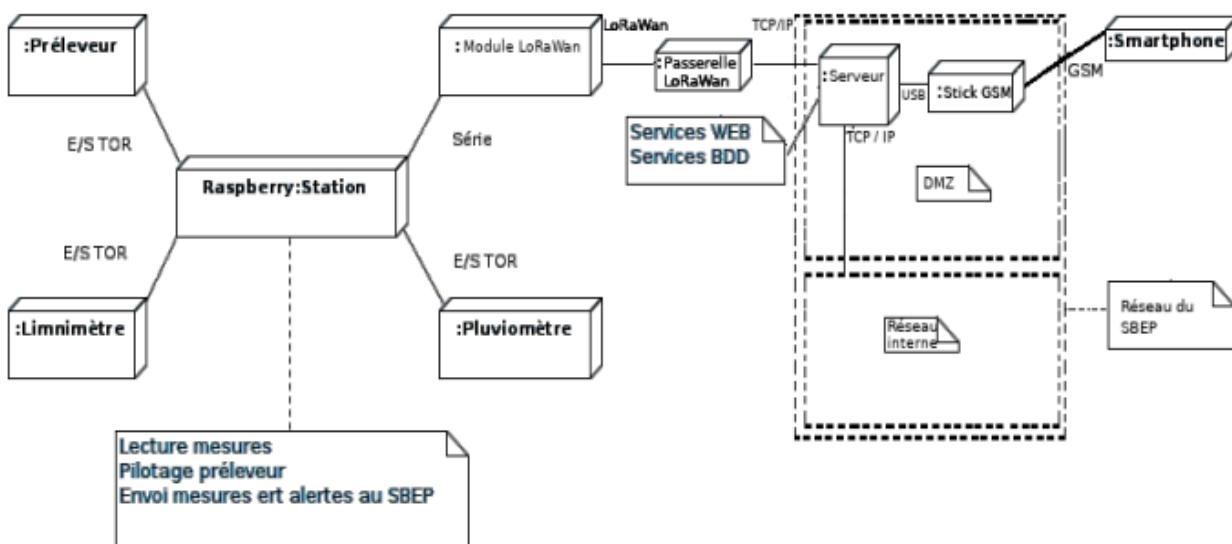


Diagramme de déploiement

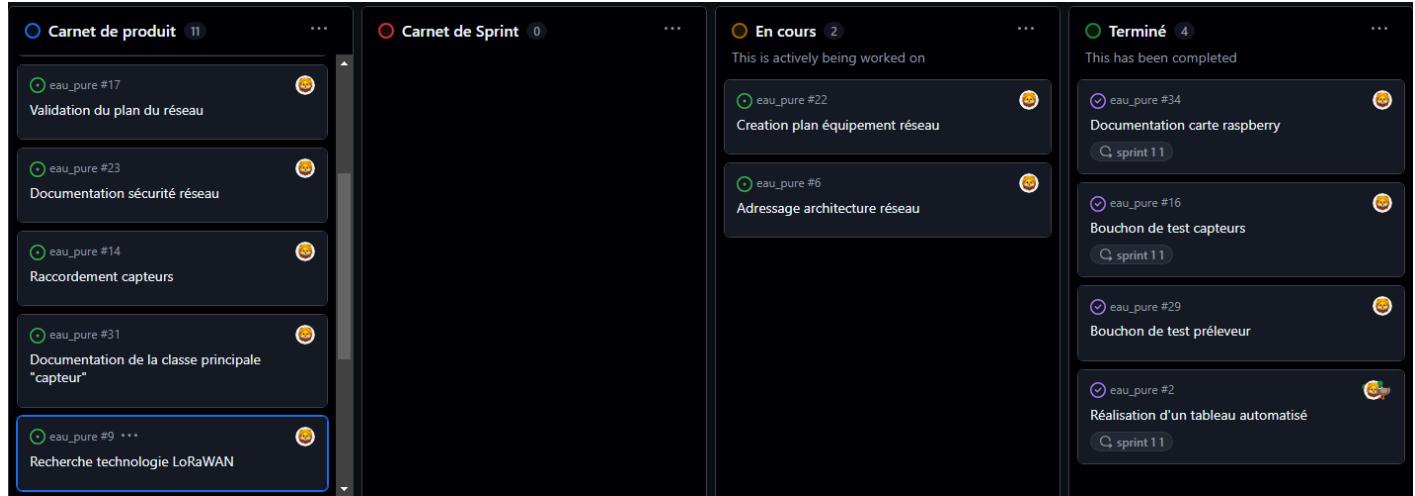
2.2. Tâche à réaliser

Pour réaliser ce projet plusieurs tâches m'ont été attribuées dans le cahier des charges. Il faut installer une carte raspberry et de l'environnement logiciel nécessaire. Raccorder les capteurs limnimètre et pluviomètre à godets à la carte. Alimenter le tout avec une source d'énergie autonome. Réaliser des bouchons de test afin de simuler le fonctionnement des capteurs et simuler l'ouverture du préleveur. Ensuite il faut envoyer les mesures dans une base de données avec une liaison LoRaWan.

Une deuxième partie met attribuées, la mise en place d'une architecture réseau. Un plan du réseau doit être réalisé, un plan d'adressage pour adresser chaque équipement. Étudier les différentes stratégies de sécurité à mettre en place pour garantir la sécurité du réseau. Pour finir mettre en œuvre l'architecture sur les équipements réels.

2.3. Réalisation d'un carnet de produit

Avant de se lancer concrètement dans la réalisation des tâches j'ai réalisé un carnet de produit, c'est un document contenant l'objectif de produit et la liste des récits utilisateur qui peuvent être réalisés au cours d'une phase d'un projet et qui sont classés en ordre de priorité.



Outil : Github

Ci-dessus nous pouvons voir le carnet de produit sur la colonne de gauche. J'ai pris une tâche du cahier des charge et je les découpe en plusieurs parties cela permet de réaliser la tâche dans le bon ordre et d'aller dans la bonne direction. Les récits utilisateurs peuvent être déplacés dans les différentes colonnes en fonction de leurs avancements, cela est pratique pour connaître son avancée dans le projet ainsi que celui des collègues.

2.4. Tâches réalisées

2.4.1. Le raspberry

J'ai commencé par réinstaller le raspberry pi pour avoir un environnement de travail propre, sans configuration antérieure pour ne pas poser problème à l'avancée du projet. Ensuite j'ai installé Python3 avec les bibliothèques nécessaires à la réalisation des codes (pip3, sqlite3, RPi.GPIO, mysql.connector).

J'ai rencontré un problème à l'installation de mysql.connector, un message d'erreur apparaît disant qu'il n'est pas possible d'installer un paquet python dans un environnement géré de manière externe donc j'ai créé un environnement virtuel sur ma machine avec les commandes.

```
sh
python3 -m venv env
Cette commande va créer un environnement virtuel nommé env.
```

3. Activer l'Environnement Virtuel

Sur Linux/macOS :

```
sh
source env/bin/activate
```

Un environnement virtuel Python est un environnement isolé qui permet d'installer des packages Python pour un projet particulier sans interférer avec les packages installés dans l'environnement global (système).

2.4.2. Réalisation bouchon de test des capteurs

La conception du bouchon de test des capteurs a été réalisée en python. Le programme se connecte à la base de données avec le module mysql.connector et les informations de connexion que j'ai renseignées (login, mot de passe). Ensuite une fonction génère des valeurs factices pour les deux capteurs avec le module random, de 0 à 20 mm pour le pluviomètre et de 0 à 10 mètres pour le limnimètre. Une deuxième fonction qui insère les valeurs factices dans une requête SQL. Pour finir la requête SQL s'exécute et envoie les informations vers la base de données. (**annexe 1**)

Suite à l'exécution du programme, des données sont enregistrées dans la base de données. (Le capteur numéro 1 correspond au pluviomètre, le numéro 2 au limnimètre)

	id	date	capteur	valeur	unite
	107	2025-03-12 10:49:10	1	0.34	L/m ²
	108	2025-03-12 10:49:10	2	8.98	m

Outil : Phpmayadmin (application Web de gestion pour les systèmes de gestion de base de données MySQL)

2.4.3. Réalisation bouchon de test préleur

Pour faire le bouchon de test du préleur j'ai également utilisé Python. Le programme se connecte à la base de données comme le programme précédent. Puis il va lire les informations contenues dans celle-ci. Il analyse les mesures les plus récentes et compare avec les seuils définis. Si les mesures dépassent les seuils, le préleur s'ouvre . (**annexe 2**)

```
SELECT * FROM `Preleveur`
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Crée le code source PHP] [Actualiser]

Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

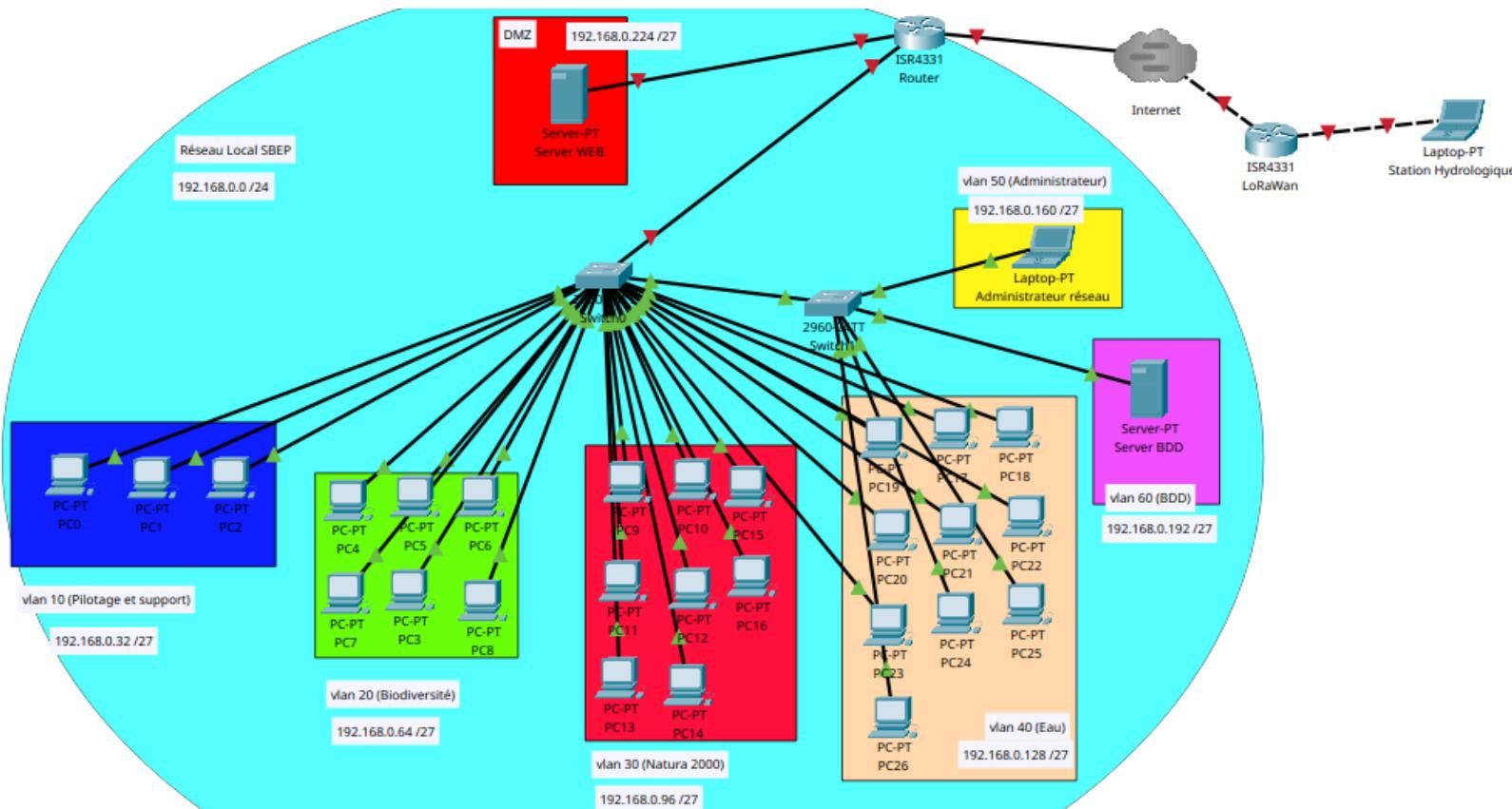
	id	capacite	reference	seuil_limnimetre	seuil_pluviometre	station_longitude	station_latitude
<input type="checkbox"/> Éditer Copier Supprimer	2	5	RPS20B	5	20	47.75	-3.36667

2.5. Partie réseau

2.5.1. Plan réseau

Pour mettre en place l'infrastructure réseau, il faut commencer par faire un schéma/plan avec tous les équipements mis à notre disposition. Il y a 2 switch, 1 routeur, 1 passerelle LoRaWan, 1 station hydrologique et les serveurs WEB et BDD. Dans le réseau SBEP il y a 27 ordinateurs répartis dans 4 services, Pilotage et support, Biodiversité, Natura 2000 et Eau.

Outil : Cisco Packet Tracer



2.5.2. Table d'adressage

Réseau	Adressage
Réseau local	@Réseau : 192.168.0.0 /24

Vlan	Adressage (sous réseau)
10 (Pilotage et support)	@Réseau : 192.168.0.32 /27 @Machine : 192.168.0.33 à 35 @Passerelle : 192.168.0.62 /27
20 (Biodiversité)	@Réseau : 192.168.0.64 /27 @Machine : 192.168.0.65 à 70 @Passerelle : 192.168.0.94 /27
30 (Natura 2000)	@Réseau : 192.168.0.96 /27 @Machine : 192.168.0.97 à 104 @Passerelle : 192.168.0.126 /27
40 (Eau)	@Réseau : 192.168.0.128 /27 @Machine : 192.168.0.129 à 139 @Passerelle : 192.168.0.158 /27
50 (Administrateur)	@Réseau : 192.168.0.160 /27 @Machine : 192.168.0.161 @Passerelle : 192.168.0.190 /27
60 (BDD)	@Réseau : 192.168.0.192 /27 @Serveur BDD : 192.168.0.193 @Passerelle : 192.168.0.222 /27

Zone	Adressage
DMZ	@Réseau : 192.168.0.224 /27 @Serveur WEB : 192.168.0.225 @Passerelle : 192.168.0.254 /27

Interface routeur	Adressage
Sous interface vlan 10 (GE 0/0.10)	192.168.0.62 /27
Sous interface vlan 20 (GE 0/0.20)	192.168.0.94 /27
Sous interface vlan 30 (GE 0/0.30)	192.168.0.126 /27
Sous interface vlan 40 (GE 0/0.40)	192.168.0.158 /27
Sous interface vlan 50 (GE 0/0.50)	192.168.0.190 /27
Sous interface vlan 60 (GE 0/0.60)	192.168.0.222 /27
DMZ (GE 0/1.10)	192.168.0.254 /27

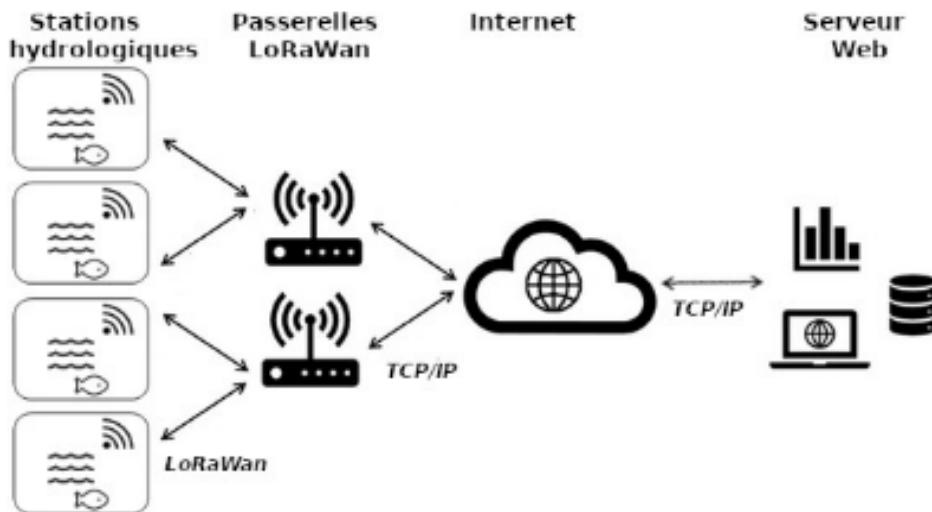
3. Etudiant n°2 - Alexis SALOU

3.1. Description du travail demandé

3.1.1. Synoptique du travail demandé

Le projet "Eau Pure" a pour objectif de vérifier la qualité de l'eau des rivières via une carte interactive. Les participants au projet sont Alexis Salou, Mathys Le Sourne et Nolan Conan. L'architecture du système se compose de stations de mesure gérées par Mathys, d'une base de données gérée par Alexis et d'un serveur web géré par Nolan. Les stations de mesure collectent les données physico-chimiques de l'eau et les envoient via LoRaWAN à la base de données. Les données sont ensuite récupérées par le serveur web et affichées sur une carte interactive. Les technologies utilisées incluent LoRaWAN pour la communication, PhpMyAdmin pour la gestion de la base de données, Docker pour la conteneurisation des applications et GitHub pour le partage de code et de documentation.

3.1.2. Schéma du travail demandé :



Images 1 (schéma explicative du projet)

3.2. Contexte :

Dans ce projet, je suis l'étudiant 2. Mon objectif est de faire le lien entre la partie "station" gérée par Mathys et la partie WEB gérée par Nolan. Pour cela, je dois mettre en place une base de données afin de récupérer les données des capteurs des stations mais aussi les données physico-chimiques transmises suite au résultat du laboratoire. En plus de cette base de données, je dois paramétriser la passerelle LoRaWAN afin de pouvoir relier les stations et ma BDD. De plus, je dois créer un lien à l'aide d'une clé GSM entre les techniciens du SBEP et la station, quand le préleveur sera prêt à rendre un prélèvement, un message devra être envoyé à un technicien du SBEP afin de le prévenir de ce dernier prélèvement.

3.3. Listes des tâches :

Étudiant 2
(Alexis S)

- Installation et paramétrage d'une passerelle LoRaWAN. Installation et paramétrage d'un serveur The Think Stack (The Think Network)
- Conception du format des messages à destination du SBEP. Conception et réalisation d'une classe de pilotage d'envoi de message via LoRaWAN en collaboration avec l'étudiant 1.
- Conception et réalisation d'une classe de pilotage de l'automate à état fini de gestion de la station, de la BDD du SBEP, du pilotage du préleveur ; ce pilotage étant conditionné à une analyse des informations fournies par les capteurs. Fourniture des requêtes SQL d'interrogation et d'alimentation de la BDD utilisées dans le SBEP.
- Paramétrage du serveur The Think Stack afin de communiquer au site Web les données.
- Installation d'un serveur LAMP local avec création et installation de la BDD. **Réception des mesures en coopération avec l'étudiant 1 et alimentation de la BDD.**
Conteneurisation du serveur LAMP en plusieurs conteneurs Docker (en collaboration avec l'étudiant 4-3) et mise en œuvre d'un orchestrateur (Docker Swarm/Portainer)
- **Paramétrage d'une clé GSM sur l'ordinateur cible. Configuration de l'environnement logiciel. Conception et réalisation d'une classe de gestion de la clé USB pour l'envoi des SMS d'alerte au technicien.**

Image 2: Cahier des charges étudiant 2

3.4. Listes des tâches réalisées :

Création du dépôt GitHub :

Dès le début du projet, J'ai créé un dépôt GitHub commun afin de centraliser et partager tous les codes sources, documentations et autres fichiers nécessaires au projet. Cela permet à toute l'équipe d'accéder facilement aux ressources et de collaborer efficacement.

Répartition des tâches avec le tableau Scrum :

Avec mon équipe, nous avons mis en place un tableau Scrum pour organiser et répartir les tâches. Nous avons défini des cas utilisateur et les avons planifiés dans des sprints de deux semaines afin de nous répartir les tâches.

Modélisation de la base de données :

J'ai conçu le modèle de la base de données en collaboration avec Nolan et notre professeur référent, M. Auffret. Nous avons défini les tables, les relations et les contraintes nécessaires pour stocker les données collectées par les stations de mesure et les données physico-chimiques.

Mise en place de la base de données sur PhpMyAdmin :

J'ai installé et configuré PhpMyAdmin pour gérer la base de données. J'ai également utilisé Docker pour déployer l'application, ce qui facilite la gestion et la maintenance de l'environnement de développement.

Lancement du serveur Docker :

J'ai configuré et lancé un serveur Docker pour héberger la base de données et permettre à mon équipe d'envoyer leurs données et de faire des requêtes. Cela garantit un environnement de développement stable et reproductible.

Planning :

Carnet de produit	Carnet de Sprint	En cours	Terminé
eau_pure #28 Mise en oeuvre architecture réseau	eau_pure #36 Interf BBD/form	eau_pure #13 Mise en place du Git	eau_pure #22 Creation plan équipement réseau
eau_pure #26 Mise en place sécurité réseau	eau_pure #45 réalisation de la BDD en graphique	eau_pure #24 Documentation de The think stack (LoraWAN)	eau_pure #34 Documentation carte raspberry
eau_pure #1 Tests automatisés	eau_pure #3 OpenStreetMap / Leaflet	eau_pure #27 Formulaire sécurisé	eau_pure #19 Documentation de la BDD
eau_pure #3 Prise en main du Serveur Jenkins	eau_pure #21 sprint 1.1	eau_pure #44 Création d'un formulaire d'inscription pour un technicien	eau_pure #35 Modélisation et mise en place de la BDD
+ Add item	+ Add item	+ Add item	+ Add item

Images 3 (Tableau Scrum)

Ce Tableau scrum, nous à permis dès le début de planifier notre projet, il nous a permis de mettre en place des cas utilisateur :

Modélisation et mise en place de la BDD #35

Alexisalou opened 2 weeks ago · edited by nolanconan

En tant qu'étudiant, je souhaite modéliser et mettre en place la base de données afin de pouvoir recevoir les données des capteurs et des prélevements.

Assignees: Alexisalou

Labels: No labels

Projects: Planning Eau Pure

Status: Réalisés

sprint 1: sprint 1 2 • Mar 10 - Mar 23 • Current

doc raspberry: No date

sprint2: Choose an iteration

Milestone: No milestone

Relationships: None yet

Image 4: (exemple cas utilisateur)

Ces cas utilisateur nous permettent d'énumérer nos tâches, de nous les répartir dans des sprints de 2 semaines et donc d'avoir un planning déjà attribué.



A Scrum planning board for March 2025. The board has two columns: 'February 2025' on the left and 'March 2025' on the right. The March column is divided into days 1 through 23. Tasks are listed in rows, each with a due date and a title. A red dot is placed over the cell for March 10, indicating the current sprint's progress.

February 2025	March 2025
22	23
24	25
26	27
28	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23

Tasks listed:

- 1. Interf BBD/form #36
- 2. réalisation de la BDD en graphique #45
- 3. Mise en place du Git #13
- 4. Documentation de The think stack (LoraWAN) #24
- 5. Creation plan équipement réseau #22
- 6. Mise en oeuvre architecture réseau #28
- 7. Mise en place sécurité réseau #26
- 8. Documentation carte raspberry #34
- 9. Documentation de la BDD #19
- 10. Modélisation et mise en place de la BDD #35
- 11. Bouchon de test capteurs #16
- 12. Bouchon de test préleveur #29
- 13. Réalisation d'un programme permettant d'ha... #39

Images 5: (Planning sprint)

Ce planning sprint est compris avec le tableau Scrum, il nous permet de définir la durée de nos tâches et de savoir plus ou moins le temps que cela va prendre selon nos estimations mais aussi savoir ce que les autres étudiants de l'équipe sont en train de faire.

3.5. Outils utilisés :

3.5.1. GitHub :

GitHub : Partage de code et documentation, version installée : service en ligne toujours à jour.

Procédure d'installation : Service en ligne

Raison du choix : Outil utilisé en cours.

3.5.2. PhpMyAdmin :

PhpMyAdmin : Gestion de la base de données, version installée : 5.2.2.

Procédure d'installation : Installer via Docker.

Raison du choix : Interface web pour gérer les bases de données utilisées en cours.

Ligne de commande pour vérifier l'installation : docker ps

3.5.3. Docker :

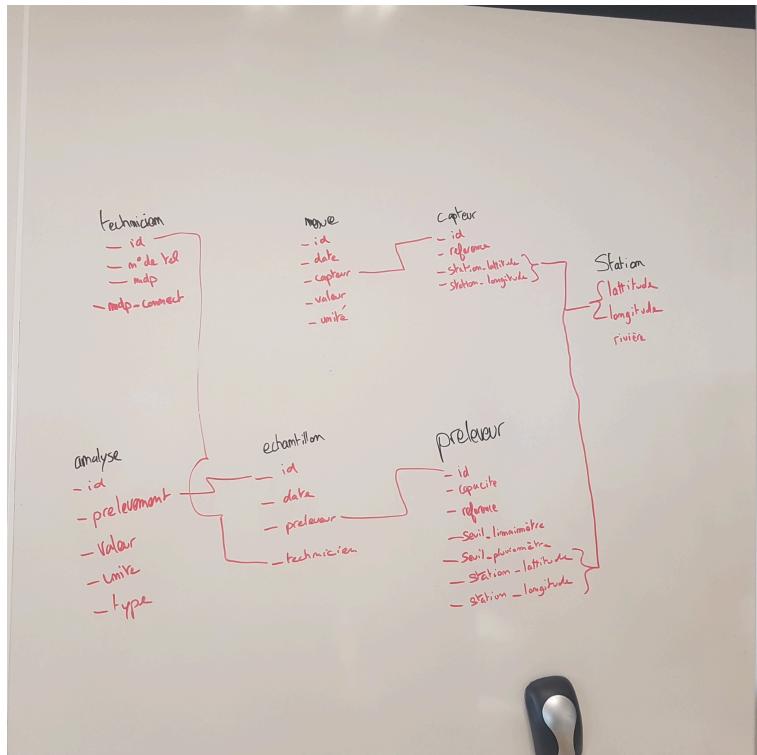
Docker : Conteneurisation des applications, version installée : 26.1.4.

Procédure d'installation : Outil déjà présent sur l'ordinateur.

Raison du choix : Facilite le déploiement et la gestion des applications déjà utilisé en cours.

Ligne de commande pour vérifier l'installation : docker --version

3.6. Spécification techniques :



Images 6 (modélisation de la BDD sur tableau)

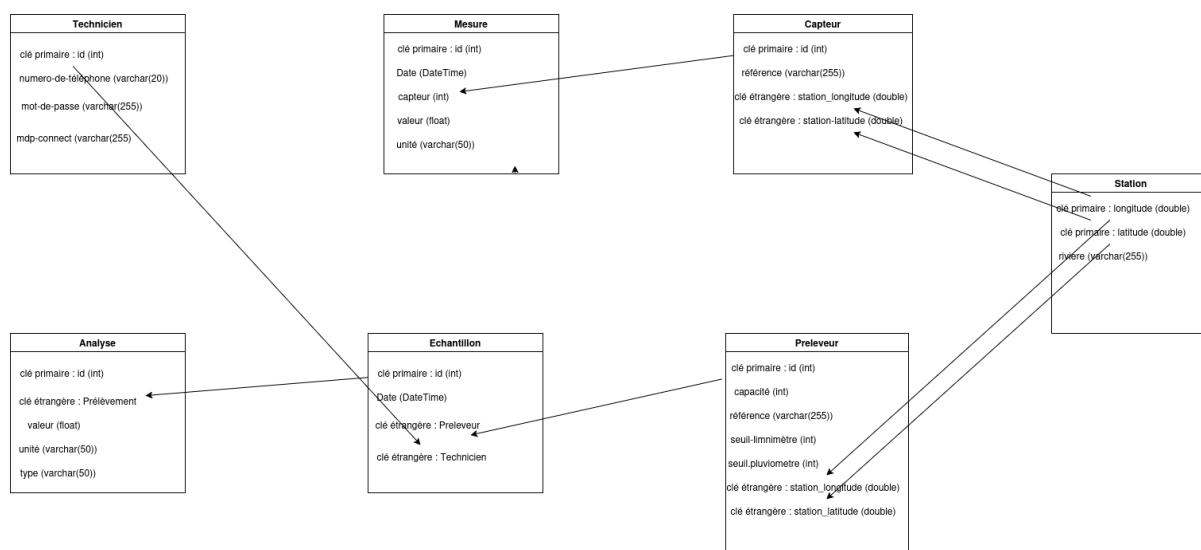


Diagramme 2 (modélisation de la BDD)

3.7. Tests effectués / validation :

A l'aide des bouchons de Test effectué par l'équipe, j'arrive à récupérer les données capteurs de la station et les données physico-chimiques.

					id	date	capteur	valeur	unité	
<input type="checkbox"/>		Éditer		Copier		Supprimer	144	2025-03-12 15:17:05	2	5 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	145	2025-03-12 15:17:06	1	19 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	167	2025-03-13 08:52:46	1	9 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	168	2025-03-13 08:52:46	2	3.67 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	169	2025-03-13 08:52:47	1	4.95 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	170	2025-03-13 08:52:47	2	8.86 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	171	2025-03-13 08:52:48	1	10.55 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	172	2025-03-13 08:52:48	2	3.62 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	173	2025-03-13 08:52:49	1	10.14 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	174	2025-03-13 08:52:49	2	3.04 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	175	2025-03-13 08:52:55	1	19.53 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	176	2025-03-13 08:52:55	2	2.93 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	177	2025-03-13 08:52:56	1	8.57 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	178	2025-03-13 08:52:56	2	5.99 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	179	2025-03-13 08:52:57	1	17.36 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	180	2025-03-13 08:52:57	2	3.04 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	181	2025-03-13 08:52:58	1	7.21 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	182	2025-03-13 08:52:58	2	2.96 m
<input type="checkbox"/>		Éditer		Copier		Supprimer	183	2025-03-13 08:52:59	1	19.51 L/m ²
<input type="checkbox"/>		Éditer		Copier		Supprimer	184	2025-03-13 08:52:59	2	3.27 m
<input checked="" type="checkbox"/>		Éditer		Copier		Supprimer	185	2025-03-13 08:53:00	1	7.67 L/m ²
<input checked="" type="checkbox"/> Console de requêtes SQL										

Image 8 : (Table Mesure de ma BDD avec les données capteurs)

Dans cette table, je récupère les informations envoyées par la station. avec un id une date un capteur attribué (capteur numéro 1 : pluviomètre, capteur numéro 2 : Limnimètre) et les données + unité.

	<input type="button" value="←"/>	<input type="button" value="T"/>	<input type="button" value="→"/>		id	prelevement	valeur	unite	type		
<input type="checkbox"/>		Éditer		Copier		Supprimer	92	1	7	ph	
<input type="checkbox"/>		Éditer		Copier		Supprimer	93	1	1000	µS/cm	conductivite
<input type="checkbox"/>		Éditer		Copier		Supprimer	94	1	500	NTU	turbidite
<input type="checkbox"/>		Éditer		Copier		Supprimer	95	1	7	mg/L	oxygene
<input type="checkbox"/>		Éditer		Copier		Supprimer	96	1	500	mg/L	dco

Image 9 : (table Analyse)

Voici la table Analyse qui reçoit les données physico-chimiques. avec un id le numéro du prélèvement, la valeur l'unité et le type (ce qu'on mesure)

4. Etudiant n°3 - Nolan CONAN

4.1. Description du travail demandé

4.1.1. Synoptique du travail demandé

L'objectif de ce projet est d'automatiser la surveillance des eaux de surface et de publier en ligne des bulletins interactifs sur les aspects quantitatifs et qualitatifs de ces eaux. Le Service Biodiversité Eau et Paysage (SBEP), une branche de la Direction Régionale de l'Environnement, de l'Aménagement et du Logement (DREAL), évalue la qualité de l'eau des rivières pour améliorer la biodiversité et la qualité de vie. Le SBEP a implanté des stations hydrologiques le long des cours d'eau, équipées de Raspberry Pi, de modules de communication LoRaWan, de capteurs (limnimètre, pluviomètre), d'une source d'alimentation autonome et d'un actionneur (préleveur d'eau). Ces stations, connectées à une plateforme IoT via un réseau LoRaWan, centralisent et traitent les données mesurées. Les prélèvements d'eau, automatisés en fonction des conditions climatiques, déclenchent des alertes SMS pour le transport des échantillons vers un laboratoire d'analyse. Les résultats sont ensuite stockés dans la base de données du serveur Web du SBEP pour publication. Le SBEP dispose d'un réseau local interconnectant tous ses équipements informatiques, localisés dans la salle informatique du service, qui regroupe quatre unités.

4.1.2. Schéma du travail demandé

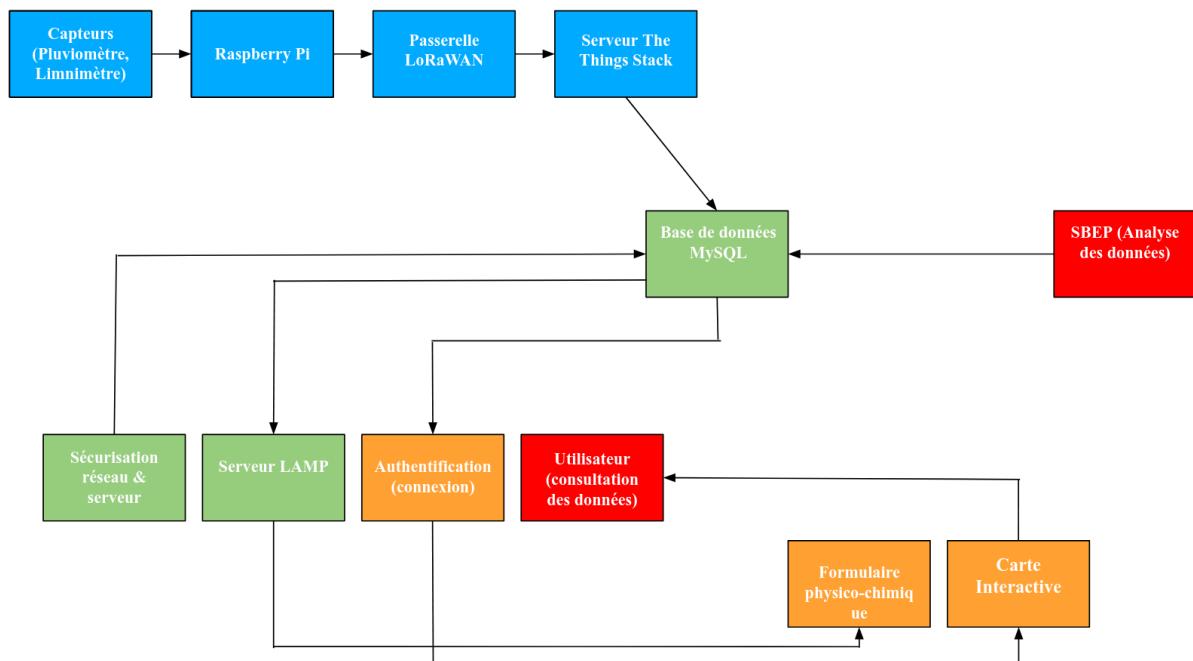


Figure 1 : Schéma résumant le travail demandé

● Acquisition & Transmission des données

- Capteurs → Raspberry Pi → Passerelle LoRaWAN → Serveur The Things Stack

● Stockage & Sécurité

- Base de données MySQL
- Serveur LAMP

- Sécurisation du réseau et des serveurs
- Interface utilisateur
 - Authentification des utilisateurs
 - Formulaire de saisie des analyses
 - Carte interactive pour visualiser les données
- Exploitation des données
 - Consultation par les utilisateurs
 - Analyse des données par le SBEP

4.2. Contexte

Dans le cadre du projet Eau Pure, mon rôle est centré sur le développement de la gestion des données physico-chimiques et leur affichage sur une carte interactive. Cette fonctionnalité est essentielle pour permettre au Service Biodiversité Eau et Paysage (SBEP) de surveiller en temps réel la qualité des eaux de surface.

Le projet vise à automatiser la collecte et la visualisation des données environnementales issues des stations hydrologiques. Après chaque prélèvement d'eau et analyse en laboratoire, les techniciens du SBEP doivent pouvoir renseigner les résultats dans un formulaire sécurisé. Ces informations sont ensuite stockées dans une base de données et doivent être accessibles de manière intuitive via une interface web.

Mon travail s'articule donc en plusieurs étapes :

1. Mise en place d'un formulaire sécurisé permettant la saisie des données physico-chimiques
2. Stockage des mesures dans la base de données pour assurer leur traçabilité.
3. Développement d'une carte interactive affichant les points de prélèvement ainsi que les résultats des analyses.

L'enjeu principal de ma partie est d'assurer une gestion fluide et sécurisée des données tout en offrant une interface claire et ergonomique pour les utilisateurs. Ce module s'intègre dans le système global en exploitant les mesures collectées par les stations et en les rendant accessibles pour le suivi et l'analyse environnementale.

4.3. Diagramme du cahier des charges

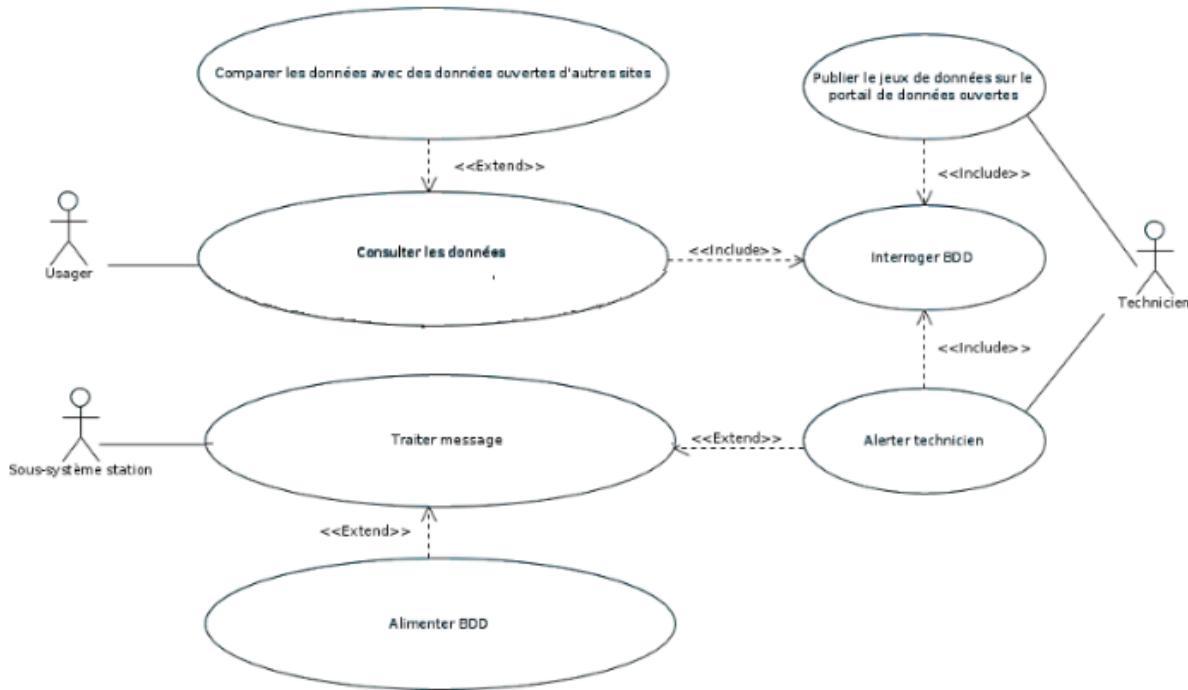


Figure 2: cas d'utilisation du sous-système SBEP

Le cas d'utilisation **Traiter message** est déclenché par tout message reçu du sous-système station. En fonction des données récupérées le cas d'utilisation **Alimenter BDD** est déclenché s'il s'agit de mesures à enregistrer dans la BDD ou bien c'est le cas d'utilisation **Alerter technicien** s'il s'agit d'une alerte. Celui-ci déclenche à son tour le cas d'utilisation **Interroger la BDD** pour connaître le numéro de téléphone du technicien c'est-à-dire l'opérateur de prélèvement de l'échantillon d'eau. Une fois la BDD alimentée le technicien peut partager les données relatives à la mesure sur le portail de données ouvertes du SIE afin d'être ré-exploitées au niveau national, c'est le cas d'utilisation **Publier le jeux de données sur le portail de données ouvertes**. Le cas d'utilisation **Consulter les données** permet de consulter à partir d'un navigateur Internet, les dernières données et les derniers prélèvements effectués. La visualisation s'effectue de manière géolocalisée en pointant sur une carte la station choisie, ce qui provoque l'affichage des informations souhaitées. Ce cas déclenche le cas d'utilisation **Interroger la BDD** pour obtenir les informations nécessaires. Le cas d'utilisation **Comparer les données avec les données ouvertes d'autres sites** exploite des données publiées sur le portail de données ouvertes et permet de comparer les mesures de la station de prélèvement avec celles d'autres stations en France.

4.4. Liste des tâches

Étudiant n°3	<ul style="list-style-type: none"> • <i>Conception et réalisation du site WEB local en utilisant le framework JavaScript React.</i> • <i>Présentation des données de mesures et d'alerte de manière graphique et géolocalisée en utilisant le mécanisme OpenStreetMap / Leaflet.</i> • <i>Conception d'un formulaire sécurisé permettant d'alimenter la base de données avec les résultats de l'analyse physico-chimique</i> • <i>Exploitation des données et API ouvertes data.gouv.fr et api.gouv.fr.</i> • <i>Automatisation du formatage des données et de leurs validation contre un modèle de données (schéma) en vue de leur publication par le technicien sur le portail de données ouvertes.</i> • <i>En collaboration avec l'étudiant 2 : conteneurisation du serveur LAMP en plusieurs conteneurs Docker.</i>
--------------	--

4.5. Planning

2 semaines : réalisation d'un programme permettant d'hasher un mot de passe ; création d'un formulaire de connexion sécurisé ; création d'un formulaire pour les données physico-chimiques ; réalisation d'un tableau Scrum.

4.6. Travail réalisé

Mise en place de l'authentification sécurisée

Pour ce projet, j'ai commencé par insérer les différentes tâches à réaliser à partir du cahier des charges dans le tableau Scrum créé sur GitHub avec mes camarades. Le cahier des charges comprend diverses tâches plus ou moins imposantes à réaliser. Il était donc essentiel de les diviser au maximum afin de progresser judicieusement sans oublier certaines étapes qui pourraient nous retarder.

Une fois les tâches planifiées, nous avons évalué leurs difficultés à l'aide du Planning Poker. Cette méthode nous a permis d'estimer l'effort nécessaire pour chaque tâche en attribuant une valeur, puis en ajustant ces estimations grâce aux discussions jusqu'à atteindre un consensus. J'ai décidé de commencer par la conception d'un formulaire sécurisé permettant d'alimenter la base de données avec les résultats de l'analyse physico-chimique. Cela me semblait plus cohérent pour l'ensemble du projet. Cette phase consiste à réaliser un formulaire de connexion sécurisé afin que le technicien du SBEP (Service Biodiversité Eau et Paysage), qui a prélevé l'échantillon d'eau, puisse, une fois l'analyse effectuée, entrer les données physico-chimiques dans ce formulaire.

J'ai d'abord travaillé sur la mise en place d'un système d'authentification sécurisé. Pour cela, j'ai créé un fichier « password_hash.php » permettant de hacher les mots de passe des utilisateurs avant de les enregistrer en base de données, garantissant ainsi une meilleure protection des informations d'authentification.

Ensuite, j'ai conçu un formulaire de connexion (index.html) dans lequel l'utilisateur doit renseigner son numéro de téléphone et son mot de passe. Le traitement de l'authentification est géré dans « connexion.php », qui vérifie si les identifiants correspondent à ceux stockés en base de données. Pour renforcer la sécurité, j'ai utilisé la fonction « password_verify() » afin de comparer le mot de passe saisi avec celui haché.

Une fois connecté, l'utilisateur est automatiquement redirigé vers la page « données_physico_chimiques.html », où il peut renseigner les résultats des analyses effectuées sur l'échantillon d'eau.

Lors de la mise en place de l'authentification, j'ai rencontré des problèmes de connexion entre mon formulaire et la base de données, causés par « Fortinet », qui bloquait les échanges entre mon application et le serveur MySQL. Cette contrainte m'empêchait de valider les identifiants et donc d'accéder aux pages suivantes.

Pour contourner cette difficulté, mon professeur référent m'a fourni un serveur « LAMP », ce qui m'a permis de travailler en local. Une fois MySQL installé sur mon poste, j'ai pu interagir correctement avec la base de données. J'ai également utilisé la commande suivante dans le terminal pour accéder à mon serveur web via Docker et exécuter les scripts MySQL :

```
sudo docker exec -it web_tp sh
```

Développement du formulaire de saisie des données physico-chimiques

Une fois l'authentification sécurisée fonctionnelle, j'ai développé le formulaire de saisie des données physico-chimiques (données_physico_chimiques.html). Ce formulaire permet aux techniciens du SBEP de renseigner les résultats des analyses d'eau effectuées en laboratoire. Les principales mesures physico-chimiques intégrées sont :

- pH (0-14)
- Conductivité électrique ($\mu\text{S}/\text{cm}$)
- Turbidité (NTU)
- Oxygène dissous (mg/L)
- Demande chimique en oxygène (DCO) (mg/L)

Chaque champ est accompagné d'un texte informatif détaillant les valeurs typiques attendues afin d'aider l'utilisateur à saisir les bonnes données.

Le traitement des données envoyées est géré dans données_physico_chimiques.php, qui se charge de :

- 1.Récupérer les valeurs saisies.
- 2.Vérifier leur validité.
- 3.Insérer ces données dans la base de données eau_pure en utilisant des requêtes SQL préparées pour éviter les failles de sécurité comme les injections SQL.

4.7. Les outils utilisés

4.7.1. Github

- **Nom de l'outil :** GitHub
- **Fonctionnalité :** GitHub est une plateforme de développement collaboratif qui permet de gérer des projets, de suivre les versions du code source et de collaborer avec d'autres développeurs.
- **Version installée :** Service en ligne, toujours à jour
- **Procédure d'installation :** GitHub est une plateforme en ligne accessible via un navigateur web. Il n'y a pas d'installation nécessaire pour utiliser GitHub. Il suffit de créer un compte sur « <https://github.com> » et de créer ou rejoindre des dépôts pour commencer à collaborer.
- **Raison du choix de cet outil :** J'ai choisi GitHub pour sa popularité et ses fonctionnalités robustes de gestion de version, de collaboration et d'intégration continue. GitHub facilite également la gestion de projets en utilisant des outils comme les Issues, les Projets (tableaux Kanban), et les Pull Requests.
- **Ligne de commande pour vérifier l'installation :** `git --version`

4.7.2. PHP

- **Nom de l'outil :** PHP
- **Fonctionnalité :** PHP est un langage de script côté serveur utilisé pour le développement web. Il permet de créer des pages web dynamiques et interactives.
- **Version installée :** PHP 8.2
- **Procédure d'installation :** Pour installer PHP, j'ai utilisé Docker pour créer un conteneur avec l'image officielle de PHP. La procédure d'installation est donc intégrée dans le fichier « `Dockerfile` » utilisé par Docker.
- **Raison du choix de cet outil :** J'ai choisi PHP car c'est un langage de script très répandu et largement utilisé pour le développement web. De plus, il est compatible avec de nombreux systèmes de gestion de bases de données, dont MySQL, ce qui facilite l'intégration et le développement.
- **Ligne de commande pour vérifier l'installation :** `php -v`

4.7.3. MySQL

- **Nom de l'outil :** MySQL
- **Fonctionnalité :** MySQL est un système de gestion de bases de données relationnelles. Il permet de stocker et de gérer les données de manière structurée et efficace.
- **Version installée :** MySQL 5.7
- **Procédure d'installation :** L'installation de MySQL a également été réalisée via Docker. J'ai utilisé l'image officielle de MySQL et configuré le service dans le fichier « docker-compose.yml ».
- **Raison du choix de cet outil :** MySQL est choisi pour sa fiabilité, ses performances et sa facilité d'utilisation. Il est également bien intégré avec PHP, ce qui simplifie le développement et la gestion des données.
- **Ligne de commande pour vérifier l'installation :** mysql -V

4.7.4. Docker

- **Nom de l'outil :** Docker
- **Fonctionnalité :** Docker est une plateforme de conteneurisation qui permet de créer, déployer et exécuter des applications dans des conteneurs. Cela garantit une exécution cohérente et isolée des applications.
- **Version installée :** Docker 20.10
- **Procédure d'installation :** Docker peut être installé en suivant les instructions officielles sur le site de Docker. Pour ce projet, j'ai utilisé Docker pour créer des conteneurs pour PHP, MySQL et PHPMyAdmin.
- **Raison du choix de cet outil :** Docker est choisi pour sa capacité à isoler les environnements de développement, ce qui permet de travailler de manière cohérente et réproductible sur différentes machines. Il simplifie également le déploiement et la gestion des dépendances.
- **Ligne de commande pour vérifier l'installation :** docker –version

4.7.5. Leaflet

- **Nom de l'outil :** Leaflet
- **Fonctionnalité :** Leaflet est une bibliothèque JavaScript open-source utilisée pour créer des cartes interactives. Elle est légère, performante et facile à utiliser, ce qui la rend idéale pour des projets nécessitant des cartes interactives sur le web.

- **Version installée :** La version installée est Leaflet v1.7.1.
- **Procédure d'installation :** `npm install leaflet`
- **Raison du choix de cet outil :** Leaflet a été choisi pour sa simplicité, sa légèreté et ses performances élevées. Il offre une large gamme de fonctionnalités pour créer des cartes interactives et est largement adopté par la communauté des développeurs. De plus, son intégration facile avec d'autres bibliothèques JavaScript et ses nombreuses extensions en font un choix idéal pour des projets nécessitant des cartes dynamiques.
- **Ligne de commande pour vérifier l'installation :** `npm list leaflet`

5. Annexe
5.1. Etudiant n°1



Projet Eau Pure
Session 2025

5.2. Etudiant n°2

5.3. Etudiant n°3

The screenshot shows a GitHub project board for 'Planning Eau Pure'. The board is organized into five columns: 'Carnet de produit', 'Carnet de Sprint', 'En cours', 'Réalisés', and 'Validés'. Each column contains several items, each with a small icon, a title, and a brief description. The 'En cours' column has one item: 'Documentation RGPD'. The 'Réalisés' column has three items: 'modélisation graphique de la BDD', 'Documentation de la BDD', and 'Modélisation et mise en place de la BDD'. The 'Validés' column has six items: 'Réalisation d'un tableau SCRUM', 'Documentation de la BDD', 'Réalisation d'un programme permettant d'hasher et saler un mot de passe', 'Création d'un formulaire de connexion sécurisé', 'Creation plan équipement réseau SBEP', and 'Addresse architecture réseau'.

Figure n°1 : Réalisation d'un tableau Scrum créé sur GitHub à partir du cahier des charges

The screenshot shows a GitHub issue page for 'Recherche technologie LoRaWAN #9'. The issue details include the assignee (MathysLS), labels (No labels), and projects (Planning Eau Pure). The activity log shows interactions between MathysLS and Alexsalou. Below the issue page is a 'ScrumPoker-online.org' interface. The interface shows a grid of cards with values 0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, and 100. The results table shows the following story points assigned to team members:

Name	Story Points
alexis	8
Eau_pure	8
Mathys	8
Nolan	8

Figure n°2 : Évaluation de la difficulté des tâches à l'aide du Planning Poker

Mot de passe haché : \$2y\$10\$1360lzCjiBcMcueF1MNxQ.bRwmh5yx6pAgSz3BtAEUSdvkBe9mEXe

Figure n°3 : Réalisation d'un programme permettant d'hasher un mot de passe



The screenshot shows a user interface for hashing a password. It features a light orange header bar at the top. Below it is a white rectangular form with rounded corners. Inside the form, there are two input fields: one for the phone number and another for the password. Both fields have a thin gray border. Below these fields is a large orange button with the text "Se connecter" in white. The entire form is set against a background that transitions from white at the top to light orange at the bottom.

Figure n°4 : Réalisation d'un formulaire de connexion

Formulaire de Données Physico-Chimiques pour le Traitement des Eaux

pH (0-14):

Le pH est une mesure de l'acidité ou de la basicité d'une solution.
Plages de Mesure Typiques :

- Acides forts : 0 à 3
- Acides faibles : 3 à 6
- Neutre (eau pure) : 7
- Bases faibles : 8 à 11
- Bases fortes : 12 à 14

Conductivité Électrique ($\mu\text{S}/\text{cm}$) (0.05-10000):

La conductivité électrique est une mesure de la capacité de l'eau à conduire un courant électrique.
Plages de Mesure Typiques :

- Eau ultra-pure : 0.05 à 1 $\mu\text{S}/\text{cm}$
- Eau de pluie : 2 à 100 $\mu\text{S}/\text{cm}$
- Eau potable : 50 à 1500 $\mu\text{S}/\text{cm}$
- Eau de rivière propre : 100 à 2000 $\mu\text{S}/\text{cm}$
- Eau de mer : 30 à 50 mS/cm (30,000 à 50,000 $\mu\text{S}/\text{cm}$)
- Eaux usées : 1000 à 10000 $\mu\text{S}/\text{cm}$

Turbidité (NTU) (0-1000):

La turbidité est une mesure de la clarité de l'eau.
Plages de Mesure Typiques :

- Eau très claire : 0 à 1 NTU
- Eau potable : 0 à 5 NTU
- Eau de rivière propre : 1 à 50 NTU
- Eau de rivière polluée : 50 à 200 NTU
- Eau très brouillée : 200 à 1000 NTU
- Eaux usées non traitées : 1000 NTU et plus

Oxygène Dissous (mg/L) (0-14):

L'oxygène dissous est une mesure de la quantité d'oxygène présente dans l'eau.
Plages de Mesure Typiques :

- Eau très propre : 8 à 14 mg/L
- Eau potable : 6 à 12 mg/L
- Eau de rivière propre : 6 à 12 mg/L
- Eau de rivière polluée : 2 à 6 mg/L
- Eaux usées : 0 à 2 mg/L

Demande Chimique en Oxygène (DCO) (mg/L) (0-1000):

La Demande Chimique en Oxygène (DCO) est une mesure de la quantité d'oxygène nécessaire pour oxyder chimiquement les matières organiques et inorganiques présentes dans l'eau.
Plages de Mesure Typiques :

- Eau très propre : 0 à 20 mg/L
- Eau légèrement polluée : 20 à 50 mg/L
- Eau de rivière polluée : 50 à 200 mg/L
- Eaux usées domestiques traitées : 20 à 100 mg/L
- Eaux usées domestiques non traitées : 200 à 600 mg/L
- Eaux usées industrielles : 200 à 1000 mg/L ou plus

Envoyer

Figure n°5 : Réalisation du formulaire pour les données physico-chimiques

				id	prelevement	valeur	unite	type
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	16		1	3	ph
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	17		1	1000	ÂµS/cm conductivite
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	18		1	500	NTU turbidite
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	19		1	14	mg/L oxygene
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	20		1	500	mg/L dco

Figure n°6 : Implémentation des valeurs physico-chimiques dans la base de données