

– RAPPORT DE PROJET –

CONCEVOIR UNE APPLICATION INFORMATIQUE

« Découvrir l'IOT et son univers »

Projet réalisé par

Jonathan ZEREN

Corran LEBOUCHARD

Maëlle AKPWEH

Projet encadré par

Éric ARTIGALA

Farhat WERFELLI

SOMMAIRE

I. Introduction	3
II. Dossier de spécification	4
1. Besoin et objectif du projet	4
2. Définition du produit	4
III. Développement technique	5
1. L'architecture de la solution	5
2. Analyse du produit	6
a. De la sonde à l'API	6
b. Script	7
c. Serveur	7
d. Base de données	7
e. Maquette du front	9
3. Outils employés	11
IV. Gestion de projet	12
1. L'équipe	12
2. La planification de projet et les outils de gestion	12
V. Les plus	13
VI. Bilan du projet	14

I. INTRODUCTION

Dans le cadre du Bloc 2 – Concevoir une application informatique au CESI, il nous est demandé de réaliser un projet portant sur la conception et la modélisation d'une application. Pour ainsi dire nous devons élaborer un prototype de système de récupération de données météorologiques issues d'éléments fournis par M. Éric ARTIGALA et présenter une maquette fonctionnelle de l'application web de visualisation des données.

Notre groupe composé de Jonathan ZEREN, Corran LEBOUCHARD et Maëlle AKPWEH, soumet notre rapport d'un projet innovant à M. Éric ARTIGALA, M. Alexandre CLAIN et M. Mathieu LE PENNEC.

II. DOSSIER DE SPÉCIFICATION

1. BESOIN ET OBJECTIF DU PROJET

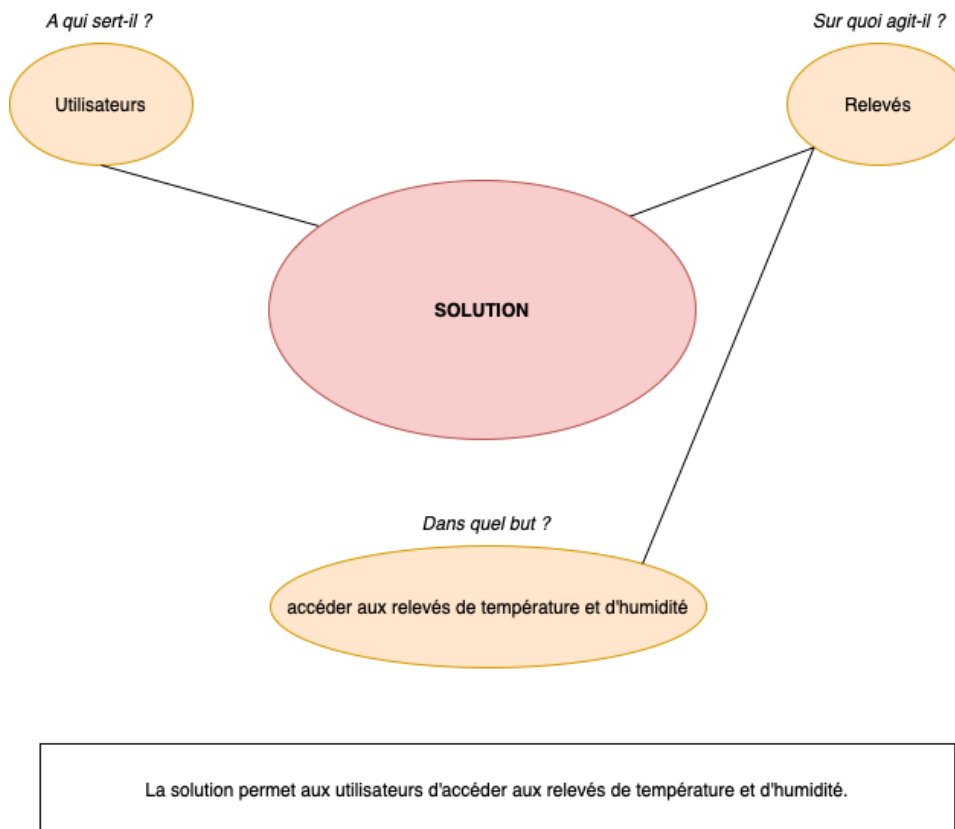
La startup “Atmos”, qui commercialise des objets connectés pour la maison, a été rachetée par la grande entreprise Lepetit. Ce rachat a impliqué une importante perte de popularité des stations météorologiques connectées et a fait craindre aux utilisateurs au sujet de la protection de leurs données privées. La solution actuelle subit des baisses de ventes dû au fait qu'elle tombe régulièrement en panne et n'est plus maintenu efficacement.

Aujourd'hui, la société Lepetit souhaiterait réaliser un nouveau système de collecte de données météorologiques à partir d'éléments spécialement sélectionnés dans le but d'une réduction des coûts. Afin de réaliser cette solution, nous devons concevoir une maquette fonctionnelle de l'application web de visualisation de données, ainsi qu'un prototype fonctionnel d'une sonde et du serveur de collecte des données météorologiques. La société va également abandonner son système d'hébergement cloud et le remplacer par un serveur local.

Notre solution permettra à la société Lepetit d'utiliser un matériel à moindre coût de production, et donc ainsi d'avoir des bénéfices économiques. Mais également, des bénéfices pour la société en elle-même, car il y aura une reprise des ventes et une augmentation du chiffre d'affaires.

2. DÉFINITION DU PRODUIT

Nous allons donc concevoir une solution logicielle permettant à l'utilisateur d'accéder aux relevés météorologiques capturés par les sondes, à partir de n'importe quel support tel qu'un téléphone, une tablette ou un ordinateur.



L'utilisateur pourra donc interagir avec le système via une interface web et consulter les données ainsi qu'un historique de ces données, il aura la possibilité de gérer des alertes de seuils et de renommer les capteurs.

III. DÉVELOPPEMENT TECHNIQUE

1. L'ARCHITECTURE DE LA SOLUTION

Pour répondre à l'objectif du projet étant de mettre en place un page web (en localhost) qui affichera les données météorologiques de différents capteurs, on devra disposer de :

- Un serveur, qui distribuera l'information
- Une base de données, dans laquelle le serveur viendra chercher ses données
- Un script, qui récupèrera les données toutes les 15 minutes et devra les convertir ainsi que les sauvegarder dans la base de données

Toute cette partie que nous avons à rédiger devra communiquer avec les objets fournis par Éric ARTIGALA, c'est-à-dire :

- L'API, qui récupère les données et les enregistre au format JSON
- Le receveur, qui sera la passerelle de communication entre le capteur et le réseau
- Les sondes, qui collecteront les données météorologiques en hexadécimal

```
sequenceDiagram
    participant User
    participant Front
    participant ServeurWeb as Serveur Web
    participant BaseDonnees as Base de Données
    participant Backend as Back-end
    participant APIArtigale as API Artigale
    participant Internet
    participant Box
    participant Receiver
    participant Sondes

    User->>Front: demande
    Front->>ServeurWeb: demande
    ServeurWeb->>BaseDonnees: demande
    BaseDonnees->>Backend: sauvegarde
    Backend->>APIArtigale: traduit
    APIArtigale->>Internet: envoie
    Internet->>Box: envoie
    Box->>Receiver: envoie
    Receiver->>Sondes: envoie
    BaseDonnees-->>Front: envoie json
    Backend-->>APIArtigale: requête json
```

Par la suite, nous mettrons en place un serveur web qui hébergera notre backend (API), la base de données et un front (une interface web) qui récupèrera et affichera les données météorologiques de chaque capteur.

a. De la sonde à l'API

```
[
    49215,

    "545A004C24240406020400000641884907900004160B0E0A1D1B00000008AAC0000001A404AF002400030B
    06182660000D6F4073FF3862182233000E5500855E5306190484000E5600B9FF3100533DEF0D0A",
    "Mon, 14 Nov 2022 09:23:53 GMT"
],
```

```
[  
  49214,  
  
  "545A004124240406020400000641884907900004160B0E0A181700000008AAC0000001A404B0001900020B  
  06182660000D6F4079FF3862182233000E5500835E510052CCF60D0A",  
  "Mon, 14 Nov 2022 09:18:49 GMT"  
]
```

b. Script

Les informations transmises des capteurs sont données par le réseau via une API en tant que données hexadécimales. Dans le but de les enregistrer, il nous faudra concevoir un algorithme qui récupèrera les données fournis par l'URL de l'API pour les envoyer vers un script qui va convertir ces données au format décimal.

Voici les données à décrypter :

- Date et heure (string : 'AAAAMMJJ HH:MM:SS')
- Nom du capteur (varchar)
- Température (float)
- Humidité (int)
- Niveau de batterie (float)
- Signal RSSI (int)

Une première fonction convertira les données hexadécimales vers le décimal. Une fois réalisée, une deuxième fonction stockera ces données dans une base de données à l'aide de requête SQL (INSERT INTO table VALUES ('valeur 1', 'valeur 2', ...)). Une dernière fonction permettra à l'aide de requête SQL et de la librairie matplotlib de réaliser des graphiques qui affichera l'historique des données.

c. Serveur

Le système sera déployé sur un serveur. Il hébergera le serveur web, le SGBDR (Système de Gestion de Base de Données Relationnelles) et l'API. Pour la communication avec la base de données, le serveur va réaliser une requête SQL (SELECT nom_du_champ FROM nom_du_tableau) pour récupérer les données des capteurs.

d. Base de données

Pour ce projet, une base de données est essentielle pour conserver les données récoltées, c'est pour cela que nous avons notre base de données à partir de la méthode MERISE.

Le Modèle Conceptuel des Données ou MCD est un modèle et une étape permettant de concevoir les entités, les relations nécessaires pour ensuite passer au MLD.

Par ailleurs, il nous a fallu reprendre le besoin de notre client et réfléchir à toutes les données nécessaires au bon fonctionnement de l'application :

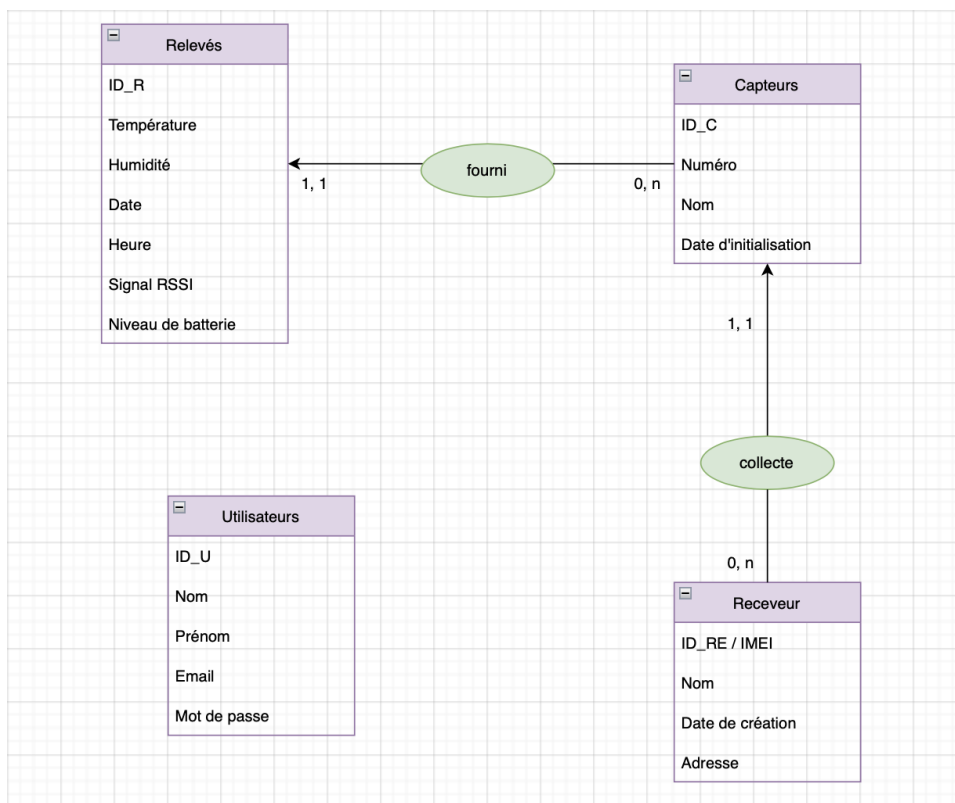
- Qui interagit avec quoi ?
- De quoi a-t-on besoin ?
- Quels sont les éléments nécessaires ?

Avant de définir les entités, nous avons défini le dictionnaire de données. Les données (température, humidité, niveau de batterie, signal RSSI, date et heure du relevé) doivent être enregistrées.

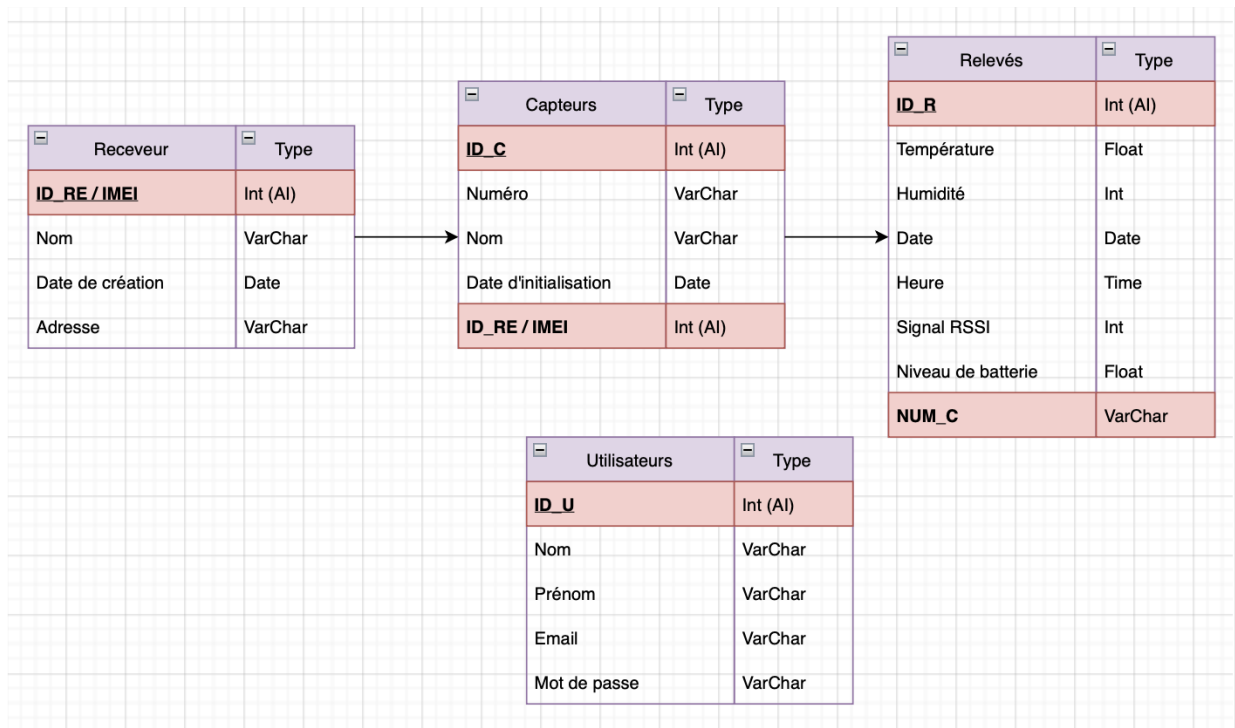
À partir de ce dictionnaire, nous avons défini quatre entités : Relevés, Capteurs, Receveur et Utilisateurs.

Ensuite, nous avons ajouté à chaque entité leurs caractéristiques. Nous avons également défini les relations avec leurs cardinalités :

- Un capteur réalise 0 à n relevés
- Un relevé est réalisé par un et un seul capteur
- Un capteur est collecté par un et un seul receveur
- Un receveur collecte 0 à n capteurs



Après avoir réalisé le MCD et le MLD, nous avons réalisé le Modèle Physique de Données ou MPD qui est la représentation physique de chaque tableau à concevoir dans la base de données. Chaque entité devient un tableau avec leurs clés primaires et leurs caractéristiques. Tenant compte des cardinalités, nous avons ajouté les clés secondaires dans les tableaux.



e. Maquette du front

L'objectif du projet étant de permettre à l'utilisateur d'avoir accès aux relevés données via une interface web, à la suite de nos premières démarches nous avons conçu la maquette de notre front.

ATMOS  Accueil Plan Mon compte

BIENVENUE SUR ATMOS - Stations météorologiques connectées !

Cette application vous permet un accès aux températures et taux d'humidité de vos sondes personnelles.

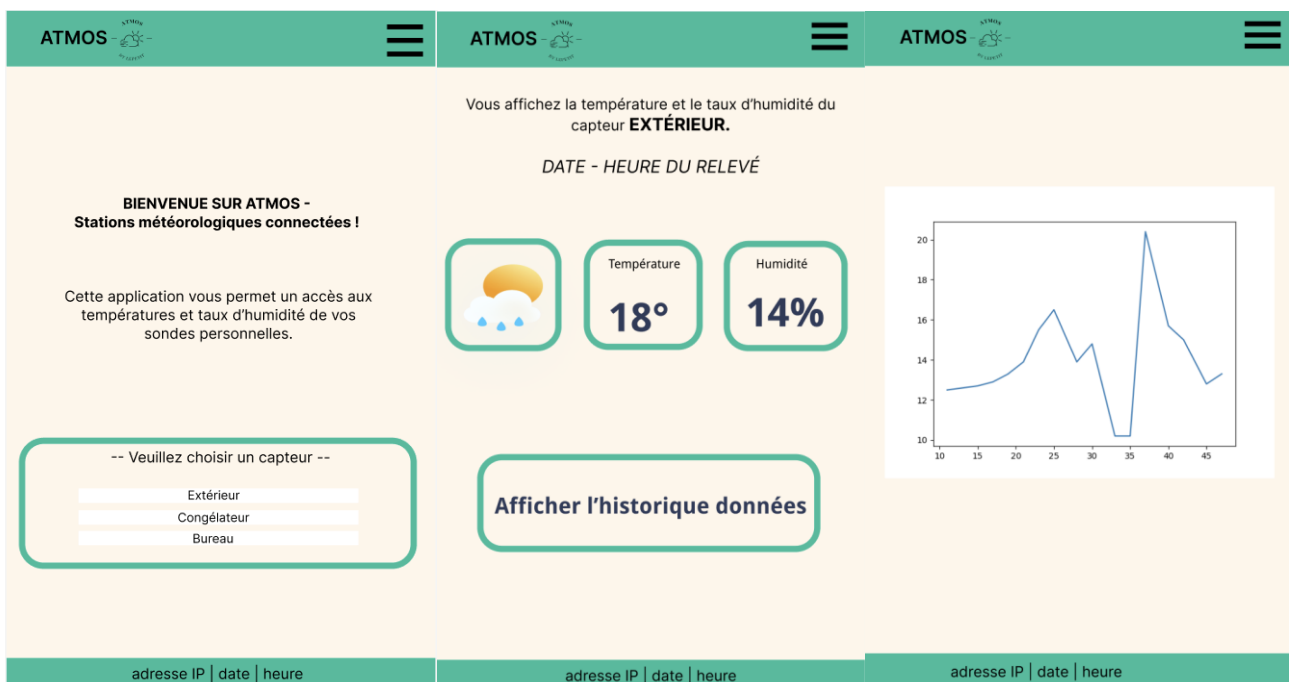
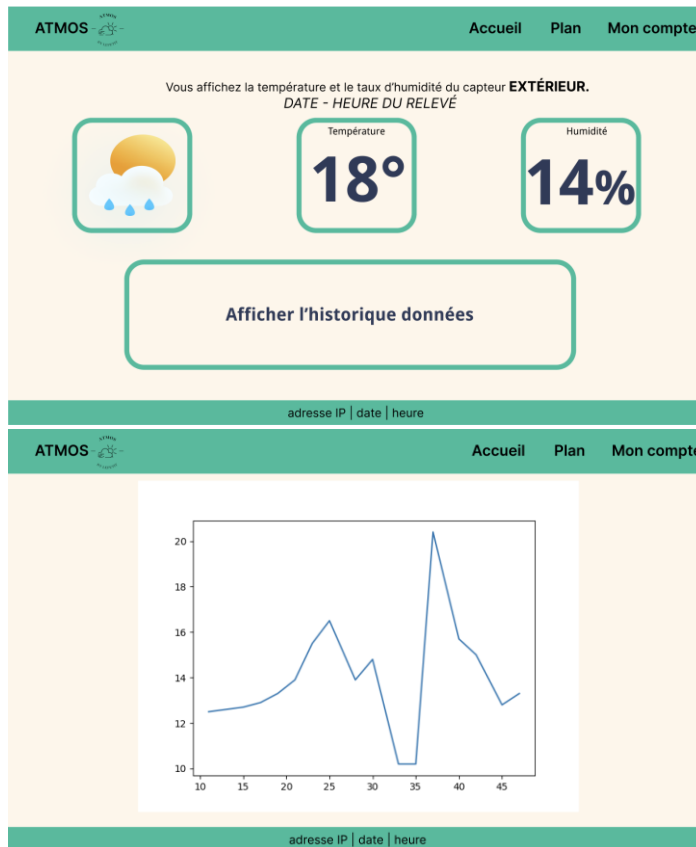
-- Veuillez choisir un capteur --

Extérieur

Congélateur

Bureau

adresse IP | date | heure



3. OUTILS EMPLOYÉS

SQL

SQL est un langage informatique servant à exploiter des bases de d

Pratiquement tous les systèmes relationnels utilisent le langage SQL pour interroger les bases de données. Ce langage permet de demander des opérations d'algèbre relationnelle telles que l'intersection, la sélection et la jointure. Dans le cadre de notre projet, l'ensemble de nos requêtes présentes dans notre programme est effectué avec ce langage.

HTML - CSS - JS

HTML est le langage de balisage conçu pour représenter les pages web. HTML permet de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.

Le CSS forme un langage informatique qui décrit la présentation des documents HTML. CSS est couramment utilisé dans la conception de sites web et bien pris en charge par la plupart des navigateurs web.

Ces deux langages ont été principalement utilisés pour la mise en page des différents onglets de notre application. Le CSS a tout particulièrement servi à donner un style à notre application et à améliorer l'expérience utilisateur et réaliser le design responsive de notre page web. De manière globale, ce sont les langages que nous avons utilisés dans la partie frontend de notre programme.

SQL Server

SQL Server est le système de gestion de bases de données (SGBD) fourni avec le MySQL server. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public que par des professionnels.

Flask

Flask est un "micro-framework" web en python. Il permet de créer des petits sites très rapidement. De plus python permet une gestion des données facilitée par sa syntaxe minimaliste.

Drawio

Drawio est un outil de dessin vectoriel gratuit qui permet de réaliser des schémas. Nous l'avons utilisé pour le schéma UML et notre MCD.

IV. GESTION DE PROJET

1. L'ÉQUIPE

Après avoir pris connaissance des besoins clients communiqués par notre Product Owner, nous nous sommes organisés en conséquence. Nous avons donc défini des domaines d'applications propre à chacun (tout en étant flexible dans un esprit de travail d'équipe et d'entraide) :

- Maëlle : Cheffe de projet, partie backend et frontend.
- Corran : Soutien général.
- Jonathan : Soutien général.

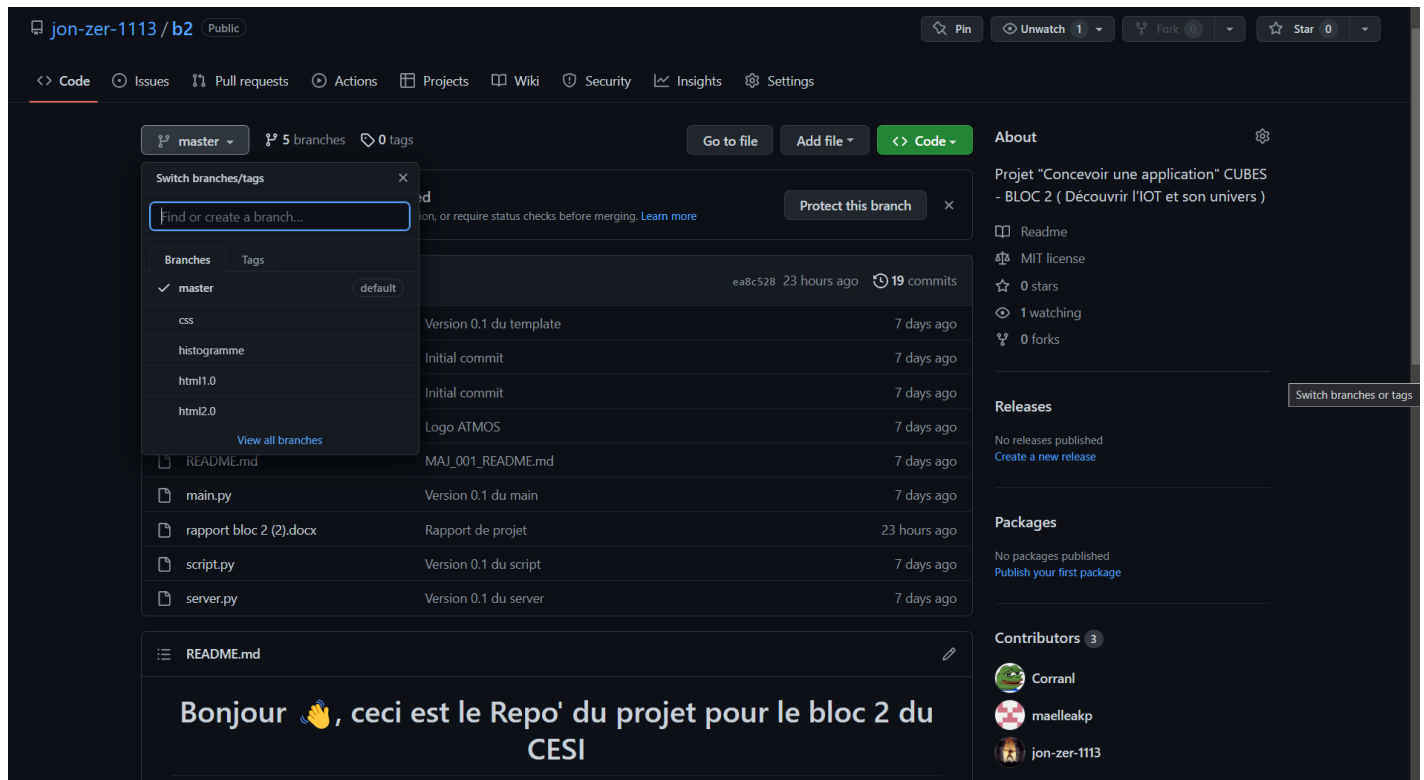
2. LA PLANIFICATION DE PROJET ET OUTILS DE GESTION

La polyvalence de notre équipe nous a permis de surmonter les obstacles de toutes les tâches à réaliser. Cette polyvalence a notamment été mise à profit dans l'attribution des tâches afin que chacun tire parti de ses accomplissements

Pour la gestion de cette mission (planification et répartition des tâches) nous avons décidé d'utiliser le logiciel JIRA, un logiciel simple et reconnu, parfait pour la gestion de projet numérique.

The screenshot displays the Jira Software interface for a project named 'Projet IOT'. The left sidebar shows navigation options like 'Feuille de route', 'Backlog', and 'Tableau'. The main area is titled 'Feuille de route' and shows a timeline view with columns for months (T., NOV., DÉC.). Tasks are organized into sprints, with some marked as 'TERMINÉ(E)' and others as 'EN COURS'. A 'Quickstart' button is visible in the bottom right corner.

Bien-sûr, nous avons aussi sécurisé la production du code avec Git (logiciel de gestion de versions décentralisé, libre et gratuit), tout ceci hébergé sur le site web GitHub, pour suivre l'évolution des différentes versions du code à distance.



Enfin, les temps impartis à la conception du système furent bref et il a fallu faire preuve de flexibilité et persévérance, parfois pour respecter les délais, parfois pour respecter les contraintes technologiques imposées par le projet.

V. LES PLUS

Arrivés vers la fin du temps impartie à notre projet, nous avons tout de même eu des réflexions sur les perspectives que devrait avoir notre projet.

Au niveau du backend et du frontend, nous pourrions

- Gérer des alertes de seuils de température et d'humidité par mail
- Gérer l'authentification des utilisateurs, les différents axes de sécurisation des données
- Automatiser la récupération des données toutes les n minutes
- Ajouter une carte de localisation des sondes
- Ajouter un pictogramme en accord avec la température récoltée
- Permettre à l'utilisateur de renommer les capteurs

Ces perspectives d'évolution de notre projet pourraient être réalisées avec un temps et des compétences plus importantes.

VI. BILAN DU PROJET

Ce projet permet de voir la conception d'une application informatique en réponse à la problématique de la société Lepetit, à savoir la transformation de son ancien système de collecte de données météorologiques vers un nouveau système moins coûteux, "open source" et faisant office de serveur local.

Nous pouvons voir les différentes étapes de réalisation de ce projet dans ce rapport, qui sont : la mise en place d'une stratégie de gestion de projet, l'analyse du besoin, la sécurisation et la gestion du code, la modélisation de la base de données, la partie backend et serveur, l'API, le frontend et enfin la restitution du projet.

De notre réflexion est née une mise en production pour le moins enrichissante et formatrice. Nous avons mené une étude du papier à la conception, modélisation du produit. En maîtrisant la production dans sa globalité, nous avons eu un aperçu complet du processus de conception d'une application informatique, des compétences et des connaissances qui y sont attachées.