



TP FINAL 2024 MS2D

Alvin Carneiro
Randy Bourdon

01 juillet 2024





Table des matières

Table des matières.....	2
I. Répartition des tâches.....	3
II. Rappel du cahier des charges.....	3
Partie de l'Application qui Réalise l'Arrosage.....	4
La Partie Électronique du Système Embarqué.....	4
Algorithme de l'IOT en Mode Automatique.....	5
Les Applications Web et Mobile.....	5
III. Planning prévisionnel.....	6
IV. Base de données Realtime.....	7
A. Règle de sécurité.....	7
B. Arborescence.....	7
V. IOT.....	8
A. Branchement de l'ESP32.....	8
B. Liste des bibliothèques du projet.....	9
VI. Application Web et mobile.....	10
C. Diagramme Use Case.....	10
D. Conception Graphique.....	11
E. Spécificité de l'application web.....	12
1. Objectif et Fonctionnalités.....	12
2. Technologies Utilisées.....	12
3. Architecture technique.....	12
4. Fonctionnalités Implémentées.....	13
F. Spécificité de l'application mobile.....	14
1. Objectif et Fonctionnalités.....	14
2. Technologies Utilisées.....	14
3. Fonctionnalités Implémentées.....	14
G. Tableau de toutes les routes.....	15
VII. Échange entre les différentes parties.....	16
VIII. Versionning du Projet.....	16
IX. Conclusion.....	17



I. Répartition des tâches

Tâche	Qui
Spécification	Ensemble
Planning prévisionnel (Gantt)	Alvin
Diagramme Use Case	Randy
Maquette graphique Web	Alvin
Maquette graphique Mobile	Randy
Tableau de toutes les routes	Randy
Schéma de branchement de l'esp32 et ses équipements	Alvin
Arborescence base de donnée	Alvin
Diagramme d'échange des différentes parties	Alvin
Code IOT	Alvin
Branchement ESP32	Alvin
Application Mobile	Randy
API Application Web	Randy
Front-end Application Web	Ensemble
Document de rendu	Ensemble

II. Rappel du cahier des charges

Nous nous proposons de réaliser un système de mesure, de contrôle et de distribution d'eau pour des jardins urbains et périurbains. Ce système est automatisé et consultable à distance.

Un apport en eau des plantes doit être régulier. Un manque hydrique prolongé est nuisible à la vie de la plante et en réduit la croissance. Le système envisagé permet de mesurer la température et l'humidité, de provoquer éventuellement un arrosage, et de tenir informé le jardinier, sans qu'il ait à se déplacer.

Il nous est demandé de réaliser un prototype d'arrosage automatique connecté pour en vérifier la faisabilité et l'efficacité tant du point de vue fonctionnel qu'économique. Ce



prototype ne traite pas l'aspect économique (coût du système, durée de vie, retour sur investissement). Nous nous concentrons sur la faisabilité technique de la partie numérique.

L'idée de notre client (association de maraîchers ou jardiniers) pour ce prototype :

- Le système possède son autonomie énergétique par l'usage d'un panneau solaire et batterie, ou est relié au secteur.
- Le système communique avec l'extérieur en Wifi, avec l'assurance d'avoir un point d'accès Wifi (Access Point) installé à proximité du dispositif.
- Les mesures physiques à contrôler pour ce prototype sont :
 - Température de l'air
 - Humidité de l'air
 - Humidité du sol

Partie de l'Application qui Réalise l'Arrosage

Cette partie utilise les composants suivants :

- Une électrovanne : reliée d'un côté au réseau d'eau (arrivée) et disposant de 4 sorties commandables électroniquement. Chaque sortie sera reliée à un tuyau goutte à goutte et pourra être ouverte indépendamment des autres. (En réalité représenté par 4 leds pour le P.O.C)
- Capteur d'humidité de l'air et de température : module DHT11.
- Quatre capteurs d'humidité du sol : chaque capteur d'humidité du sol contribuera ou non à l'ouverture d'une des 4 sorties de l'électrovanne (En réalité représenté par un capteur à des fins de démonstration et 3 potentiomètre à des fins de simulations pour le P.O.C).
- Système de goutte à goutte (Non présent pour le P.O.C).

La Partie Électronique du Système Embarqué

L'IOT est composé d'une carte électronique (proposée : ESP32) qui permet de communiquer vers l'extérieur en Wifi. Ce composant :

- Lit régulièrement :
 - La température ambiante
 - L'humidité de l'air
 - L'humidité du sol des 4 capteurs
 - La date et l'heure
- Réalise les actions suivantes :
 - Ouvrir et fermer de façon autonome chacune des 4 vannes de façon indépendante selon un algorithme décrit ci-après.
 - Ouvrir et fermer chacune des 4 vannes selon un ordre envoyé depuis l'application Web ou mobile. La sortie passe alors en mode « manuel ». La reprise de cette sortie en fonctionnement autonome, le mode « automatique



» ne se fera que sur la réception d'un nouvel ordre reçu. Une remise sous tension de l'IOT doit tenir compte des sorties en mode manuel ou automatique précédents.

Algorithme de l'IOT en Mode Automatique

Le système distingue 3 niveaux d'hygrométrie :

- Faible : sol trop sec, une action est à prévoir. La sortie de l'électrovanne doit être ouverte par tranche de 5 minutes toutes les 15 minutes jusqu'à constater une hygrométrie normale.
- Normale : pas d'action immédiate requise. Le système attend le matin, à une heure configurable par l'appli mobile (par défaut 6h), pour ouvrir la sortie de l'électrovanne par tranche de 5 minutes toutes les 15 minutes jusqu'à obtenir une hygrométrie forte.
- Forte : période très pluvieuse ou suite à un arrosage, aucune action à prévoir.

Lorsque le Wifi est opérationnel, l'IOT communique vers l'extérieur pour mémoriser les informations suivantes toutes les 30 minutes :

- La température ambiante
- L'humidité de l'air
- L'humidité du sol des 4 capteurs

Ces informations seront affichées sur des écrans des applications Web et mobile. Lors du premier emploi de l'IOT, celui-ci découvre les bornes Wifi disponibles et propose, via un téléphone portable, la borne à laquelle se connecter. Un bouton reset permet de réinitialiser cette procédure.

Les Applications Web et Mobile

Notre client souhaite se procurer plusieurs systèmes d'arrosage pour un même endroit ou pour plusieurs endroits différents. Les applications Web et mobile doivent permettre :

- De s'authentifier
- De gérer des systèmes d'arrosage : ajouter/modifier/supprimer
- De montrer la liste des systèmes d'arrosage pour cet utilisateur
- De montrer le détail d'un système d'arrosage, permettant :
 - De modifier le mode auto/manu de chaque sortie
 - D'ouvrir/fermer une sortie en mode manu
 - Pour l'application Web, de montrer un historique (courbe) des valeurs mesurées des 2 dernières semaines



- Pour l'application mobile, de montrer un historique (texte) des valeurs mesurées de la dernière journée. Un historique sous forme de courbe serait un plus, selon le temps de développement disponible.
- Créer un nouveau jardinier qui sera caractérisé par :
 - Nom, prénom
 - Adresse
 - Adresse email
 - Mot de passe

Idéalement, cela se fait depuis l'application Web par un compte administrateur. Alternativement, la création d'un nouveau jardinier peut se faire par un moyen externe (comme la Console Firebase).

III. Planning prévisionnel

Diagramme de gantt pour la période du 18 avril au 25 mai.

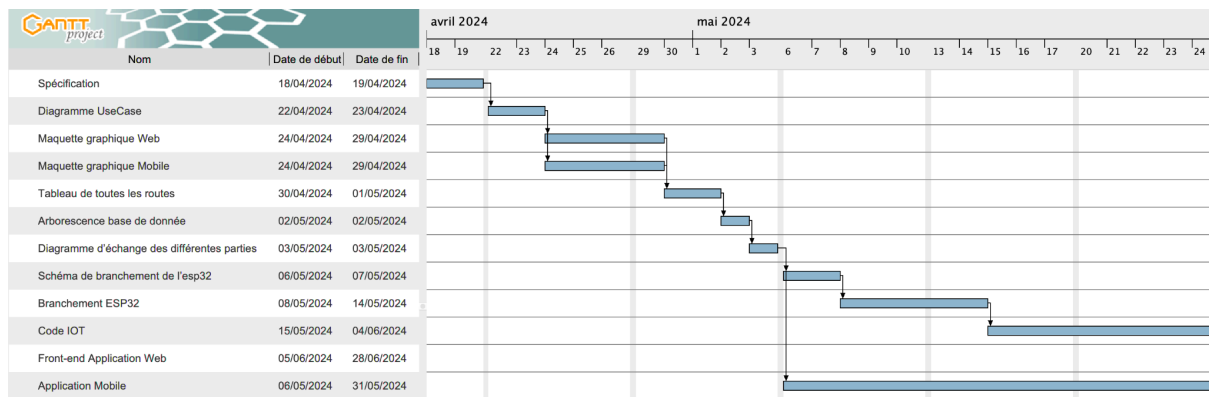
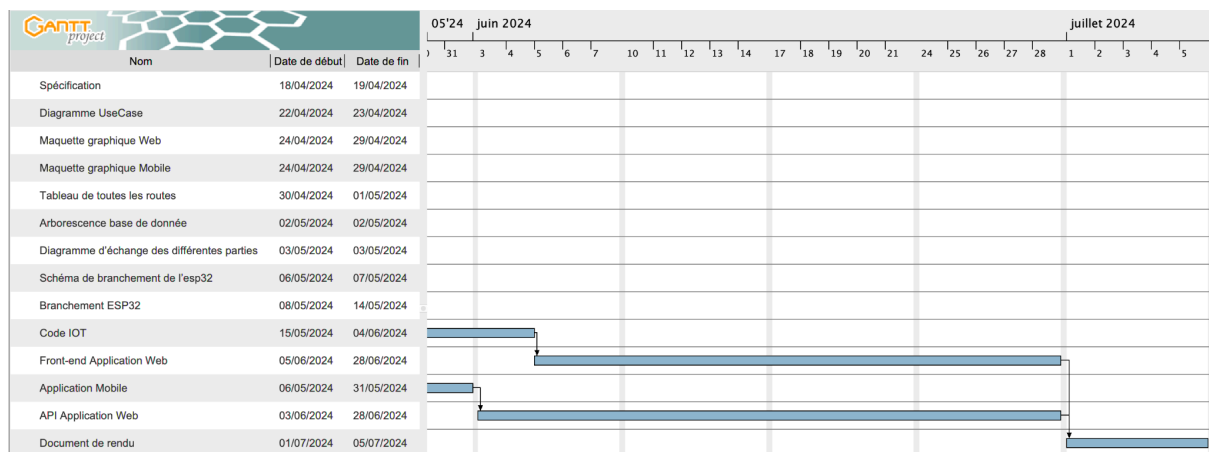


Diagramme de gantt pour la période du 30 mai au 07 juillet.





IV. Base de données Realtime

A. Règle de sécurité

Pour assurer la sécurité des données dans notre base de données Realtime sur Firebase, nous avons utilisé un système d'authentification basé sur des identifiants uniques (UID) pour chaque utilisateur. Voici les principales mesures de sécurité que nous avons mises en place :

- **Authentification des utilisateurs** : Chaque utilisateur se connecte avec des informations d'identification uniques (UID) générées par Firebase Authentication. Cela garantit que chaque utilisateur accède uniquement à ses propres données.
- **Règles de sécurité de la base de données** : Nous avons défini des règles de sécurité dans Firebase pour contrôler qui peut lire et écrire des données. Par exemple, il faut être authentifié et fournir un UID pour requêter la base de données.
- **Communication sécurisée** : Toutes les données transmises entre les utilisateurs et la base de données sont cryptées via le protocole HTTPS, garantissant que les informations ne peuvent pas être interceptées par des tiers.

Ces mesures permettent de protéger les données des utilisateurs et d'assurer que seules les personnes autorisées peuvent accéder aux informations stockées dans la base de données Realtime.

B. Arborescence

Arborescence des systèmes :



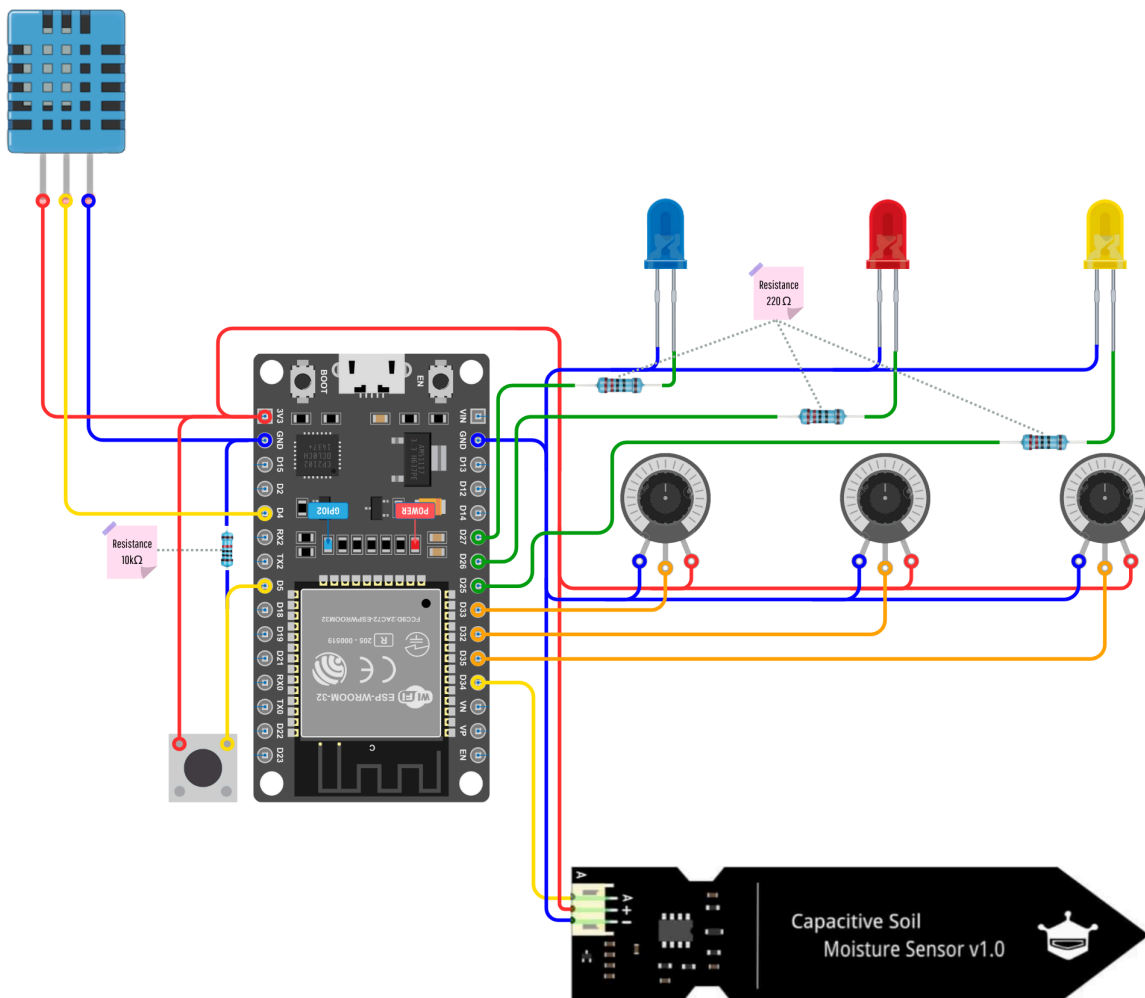


Arborescence des jardiniers :

- ▼ — `jardinier` *Racine de tous les jardiniers*
 - ▼ — `vKEWd0hrdtSazfBI9mY20pawc8h2` *Identifiant unique d'un jardinier correspondant à son UID Firebase*
 - `admin`: Booléen déterminant s'il est admin ou non
 - `adresse`: Adresse postale du jardinier
 - `img`: Image de profil
 - `nom`: Nom du jardinier
 - `prenom`: Prénom du jardinier

V. IOT

A. Branchement de l'ESP32





Pour réaliser ce montage vous aurez besoin de :

- un ESP32,
- un module DHT11 (capteur de température et d'humidité de l'air),
- un bouton poussoir,
- une résistance de 10.000 ohm,
- 3 résistance de 220 ohm,
- 3 leds,
- 3 potentiomètres,
- un capteur d'humidité de sol,
- 2 (recommandé) planches de branchement,
- de nombreux câbles de diverses tailles.

B. Liste des bibliothèques du projet

Nom de la bibliothèque	Développeur	Version
DHT sensor librar	Adafruit	1.4.6
Adafruit Unified Sensor	Adafruit	1.1.14
Firebase esp client	Mobizt	4.3.6
Firebase Arduino Client Library for ESP8266 and ESP32	Mobizt	4.3.6
WifiManger	Tzapu	2.0.16
AsyncTCP	dvarrel	1.1.4



VI. Application Web et mobile

C. Diagramme Use Case

Le diagramme de cas d'utilisation représente la structure des principales fonctions requises par les utilisateurs du système. C'est le premier diagramme du modèle UML, assurant la relation entre l'utilisateur et les objets mis en œuvre par le système.

L'objectif du diagramme de cas d'utilisation est de fournir une vue globale des interactions entre les acteurs et le système afin de comprendre les fonctionnalités du système et les rôles des différents acteurs.

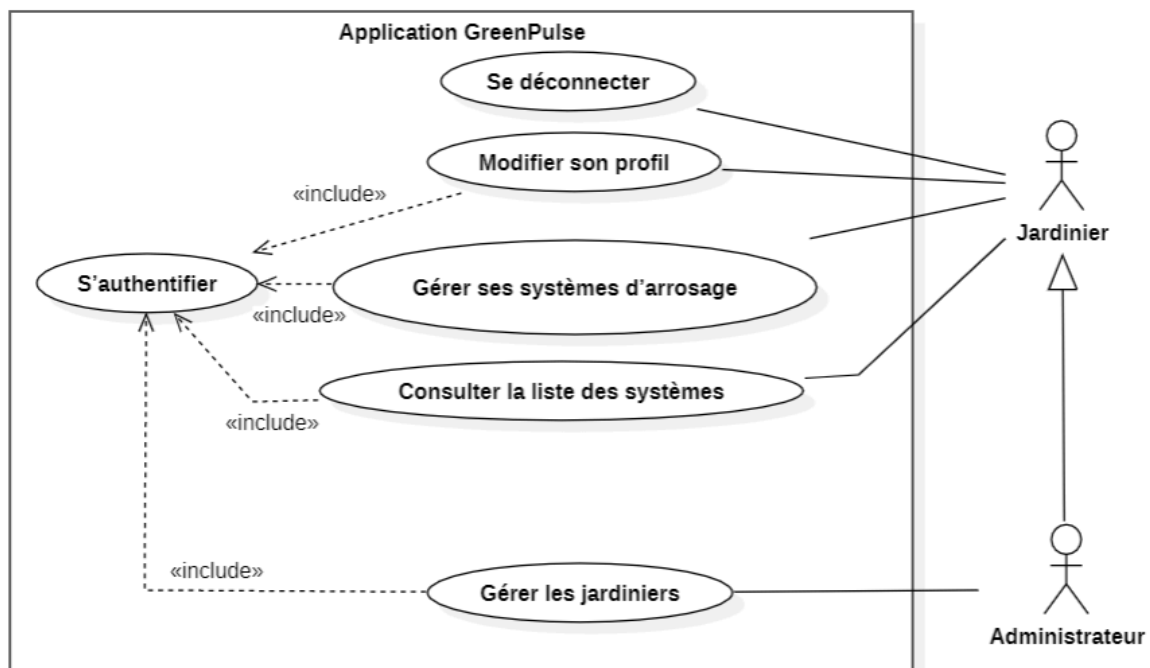
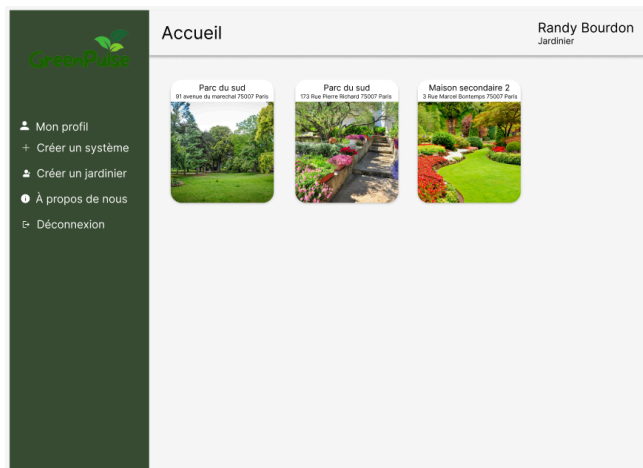
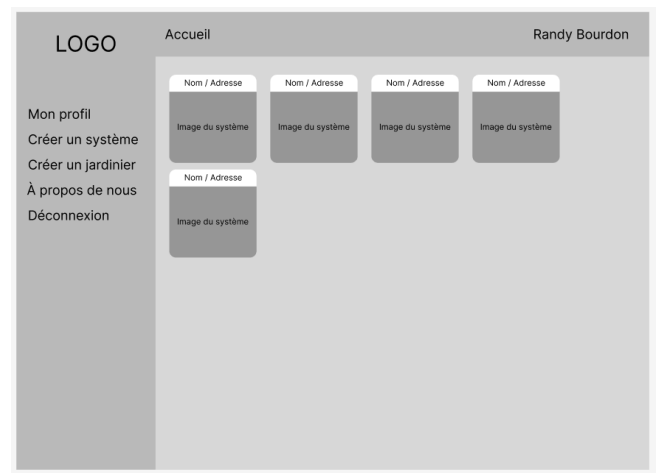
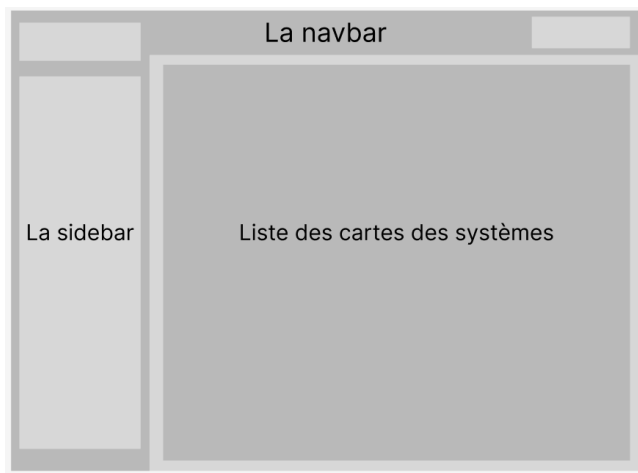


Diagramme de cas d'utilisation de l'application GreenPulse



D. Conception Graphique



Le zoning, wireframe et la maquette graphique de l'application web



Le zoning, wireframe et la maquette graphique de l'application mobile

E. Spécificité de l'application web

1. Objectif et Fonctionnalités

L'objectif de cette application est de fournir une plateforme pour gérer un système d'arrosage automatique connecté, permettant aux utilisateurs de contrôler et surveiller à distance leurs systèmes d'arrosage.

2. Technologies Utilisées

- Backend : Node.js
- Frontend : Spring Boot
- Base de données : Firebase
- API REST : Pour la communication entre le frontend et le backend
- Sécurité : HTTPS pour sécuriser les communications

3. Architecture technique

- Backend : API REST développée avec Node.js pour la gestion des utilisateurs, des systèmes d'arrosage et des données.
- Frontend : Interface utilisateur développée avec Spring Boot, consommant l'API REST du backend.



4. Fonctionnalités Implémentées

1. Authentification des utilisateurs

- Backend :
 - API pour l'inscription, la connexion et la déconnexion des utilisateurs.
 - Utilisation de token fourni par Firebase pour sécuriser les sessions.
- Frontend :
 - Formulaires de connexion et d'inscription.
 - Gestion des sessions utilisateurs.

2. Gestion des systèmes d'arrosage

- Backend :
 - API pour ajouter, modifier et supprimer des systèmes d'arrosage.
 - Modèle de données pour les systèmes d'arrosage.
- Frontend :
 - Interface pour gérer les systèmes d'arrosage.
 - Formulaire pour ajouter et modifier des systèmes.

3. Surveillance et contrôle

- Backend :
 - API pour récupérer les données en temps réel des systèmes d'arrosage.
 - Gestion des commandes manuelles et automatiques.
- Frontend :
 - Tableau de bord affichant la liste des systèmes.
 - Interface pour afficher les détails et contrôler les systèmes.

4. Historique des mesures

- Backend :
 - API pour récupérer l'historique des mesures.
- Frontend :
 - Graphiques pour visualiser les données historiques des systèmes d'arrosage.

Cette application d'arrosage utilise une architecture moderne avec un backend développé en Node.js et un frontend en Spring Boot. La structure du code est organisée en modules clairs pour une gestion facile et une maintenance efficace. L'application répond aux besoins des utilisateurs en offrant une interface intuitive pour la gestion et le contrôle des systèmes d'arrosage, tout en assurant la sécurité et la scalabilité nécessaires.



F. Spécificité de l'application mobile

1. Objectif et Fonctionnalités

L'objectif de cette application mobile est de fournir une plateforme pour gérer un système d'arrosage automatique connecté, permettant aux utilisateurs de contrôler et surveiller à distance leurs systèmes d'arrosage depuis leur smartphone.

2. Technologies Utilisées

- Langage : Dart
- Framework : Flutter
- Backend : Firebase (Firebase Authentication, Firebase real time)
- Base de données : Firebase

3. Fonctionnalités Implémentées

1. Authentification des utilisateurs

- Firebase Authentication :
 - Inscription, connexion et déconnexion des utilisateurs.
 - Gestion des sessions utilisateur.
- Flutter :
 - Formulaires de connexion et d'inscription.
 - Gestion des états d'authentification.

2. Gestion des systèmes d'arrosage

- Firebase Realtime Database :
 - Stockage des données des systèmes d'arrosage.
 - CRUD (Créer, Lire, Mettre à jour, Supprimer) des systèmes.
- Flutter :
 - Écran pour l'ajout, la modification et la suppression des systèmes.
 - Affichage de la liste des systèmes d'arrosage.

3. Surveillance et contrôle

- Firebase Realtime Database :
 - Récupération des données en temps réel des systèmes d'arrosage.
 - Gestion des commandes manuelles et automatiques.
- Flutter :
 - Tableau de bord affichant les systèmes et leurs détails.
 - Interface utilisateur pour contrôler les systèmes.

4. Historique des mesures

- Firebase Realtime Database :
 - Stockage et récupération des mesures historiques.
- Flutter :
 - Graphiques interactifs pour visualiser les données historiques des systèmes d'arrosage.



Cette application mobile d'arrosage utilise Dart et Flutter pour une interface utilisateur réactive et moderne, avec Firebase en backend pour la gestion des utilisateurs, des systèmes d'arrosage et des mesures. La structure du code est organisée de manière modulaire pour faciliter la maintenance et l'extensibilité. L'application offre une expérience utilisateur fluide et sécurisée, répondant aux besoins des jardiniers pour gérer et surveiller leurs systèmes d'arrosage depuis leur smartphone.

G. Tableau de toutes les routes

Routes Backend (Node.js)

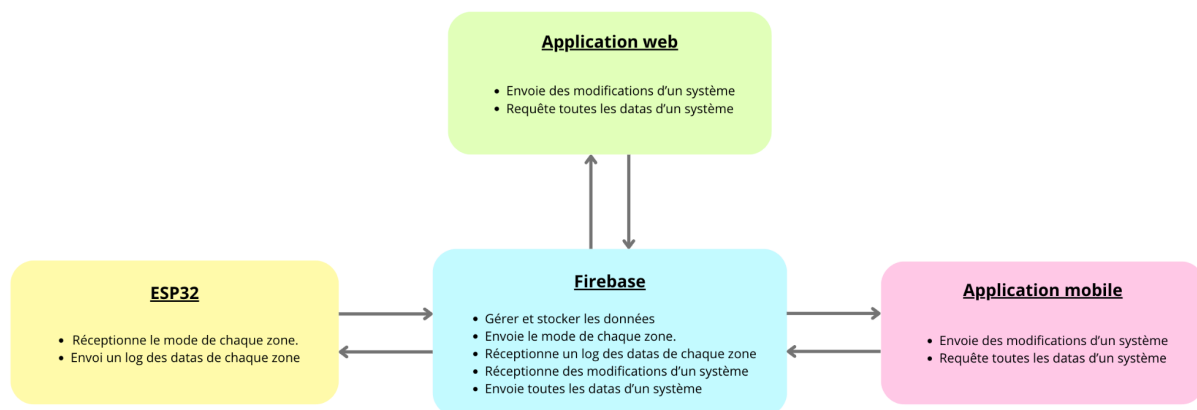
Méthode HTTP	Route	Description	Contrôleur	Autorisation
POST	/api/register	Inscription d'un nouvel utilisateur	firebaseAuthController	Public
POST	/api/login	Connexion d'un utilisateur	firebaseAuthController	Public
POST	/api/logout	Déconnexion d'un utilisateur	firebaseAuthController	Public
GET	/api/jardiniers	Liste de tous les jardiniers	jardinierController	Authentifié
GET	/api/jardinier/:id	Détails d'un jardinier	jardinierController	Authentifié
POST	/api/jardiniers	Ajouter un nouveau jardinier	jardinierController	Authentifié
PUT	/api/jardinier/:id	Mettre à jour un jardinier	jardinierController	Authentifié
DELETE	/api/jardinier/:id	Supprimer un jardinier	jardinierController	Authentifié



Routes Frontend (Spring Boot)

GET	/login	Afficher la page de connexion	AuthController	Public
POST	/login	Traiter la connexion de l'utilisateur	AuthController	Public
GET	/systemes	Afficher la liste des systèmes d'arrosage	SystemController	Authentifié
GET	/systems/{id}	Afficher les détails d'un système	SystemController	Authentifié

VII. Échange entre les différentes parties



Ce diagramme illustre les échanges de flux entre un ESP32, une application web, une application mobile et Firebase, qui centralise toutes les communications. L'ESP32 envoie des logs de données à Firebase et requête les modes de chaque zones. Les applications web et mobile envoient des modifications d'un système et requête des données via Firebase, qui gère et stocke toutes les informations, assurant la synchronisation et la mise à jour des différents systèmes.



VIII. Versionning du Projet

Le versioning de l'application d'arrosage intelligent suit les meilleures pratiques pour garantir une gestion efficace et organisée des différentes versions du projet. Chaque étape de développement est minutieusement documentée et enregistrée pour assurer une traçabilité complète des modifications et des mises à jour.

Pour ce projet, un système de gestion de version comme Git est utilisé. Git permet de créer des branches pour chaque nouvelle fonctionnalité ou correctif, ce qui facilite le développement en parallèle et la collaboration entre les membres de l'équipe. Les branches sont nommées de manière descriptive pour refléter la nature des modifications apportées.

IX. Conclusion

Le projet GreenPulse représente une solution complète et innovante pour l'arrosage intelligent des jardins urbains et périurbains. En intégrant des technologies avancées comme les capteurs IoT, Firebase pour la gestion de données en temps réel, et des interfaces utilisateur développées avec des frameworks modernes comme Node.js et Spring Boot, ce système permet une gestion optimisée de l'eau tout en assurant une surveillance et un contrôle à distance.

L'approche méthodique utilisée, incluant une répartition claire des tâches, un planning prévisionnel détaillé et une attention particulière à la sécurité des données, garantit non seulement la faisabilité technique du prototype mais aussi sa robustesse et sa scalabilité pour une future mise en production. Le recours à des technologies éprouvées assure également une maintenabilité et une évolutivité du système.

En conclusion, GreenPulse ne se contente pas de répondre aux besoins immédiats de gestion de l'arrosage, mais pose les bases d'une solution durable et adaptable, répondant aux défis futurs de la gestion intelligente de l'eau pour les jardiniers urbains.

