

# Cahier des charges

**Projet Cloud of things :**

SMART GARDEN



**Réalisé par :**

Nouha ben Hamada

Nour Laabidi

**Encadrées par :**

M. mohamed Bécha Kaâniche

**Année universitaire : 2023/2024**

## Table des matières

1	Concept . . . . .	2
1.1	Contexte . . . . .	2
1.2	Problématique . . . . .	2
1.3	Solution . . . . .	3
2	Segment Clientèle . . . . .	3
3	Besoins . . . . .	3
3.1	Fonctionnel . . . . .	3
3.2	Non fonctionnel . . . . .	3
4	Technologies Choisies . . . . .	4
4.1	Partie Software . . . . .	4
4.1.1	Middleware . . . . .	4
4.1.2	Backend . . . . .	4
4.1.3	Frontend . . . . .	4
4.1.4	Intégration IoT . . . . .	5
4.1.5	Intégration MLOps . . . . .	5
4.2	Partie Hardware . . . . .	5
5	Architecture . . . . .	6
6	Limites . . . . .	6
7	Modèle Commercial . . . . .	7
7.1	Marketing Mix . . . . .	7
7.2	Analyse SWOT . . . . .	8

## Table des figures

1	Architecture du projet . . . . .	6
---	----------------------------------	---

# 1 Concept

## 1.1 Contexte

L'irrigation est une technique qui permet d'apporter de l'eau aux plantes pour les aider à se développer dans de bonnes conditions, notamment lorsque les précipitations naturelles sont insuffisantes. Elle est essentielle pour garder les plantes en bonne santé, surtout pendant les périodes de sécheresse ou dans les régions où il pleut peu. Certaines plantes ont besoin d'un apport régulier en eau pour bien pousser et peuvent souffrir si elles n'en reçoivent pas assez, ce qui ralentit leur croissance et peut même les faire dépérir. D'autres plantes sont encore plus sensibles à la sécheresse et ne survivent pas en cas de manque d'eau.

Dans ce contexte, il devient nécessaire de disposer d'un système qui aide à mieux gérer l'irrigation en avertissant et prédire lorsqu'un arrosage est nécessaire garantissant ainsi un arrosage adapté aux besoins de la plante tout en limitant le gaspillage d'eau.

Dans ce travail, nous proposons de concevoir un modèle réduit de jardin intelligent, ou "smart garden".

## 1.2 Problématique

L'irrigation traditionnelle présente plusieurs défis majeurs, notamment l'adaptation aux conditions climatiques variables, la gestion de la consommation d'eau et la préservation de la santé des plantes.

- Adaptation aux conditions environnementales : Comment adapter les cycles d'arrosage aux variations climatiques et environnementales pour garantir un arrosage efficace et éviter la sur-irrigation ou la sous-irrigation ?
- Adaptation aux conditions environnementales : Comment adapter les cycles d'arrosage aux variations climatiques et environnementales pour garantir un arrosage efficace et éviter la sur-irrigation ou la sous-irrigation ?
- Gaspillage d'eau : Comment réduire le gaspillage d'eau dans les systèmes d'irrigation traditionnels, tout en maintenant la santé et la vitalité des plantes, que ce soit dans un cadre domestique ou agricole ?
- Survie des plantes : Comment assurer un approvisionnement en eau optimal pour les plantes afin d'éviter le stress hydrique, qui peut nuire à leur croissance et parfois même entraîner leur dépérissement ?

## 1.3 Solution

Pour répondre aux défis liés à l'irrigation, nous allons développer un système d'irrigation intelligent qui utilise un modèle de machine learning. Ce système sera capable de déterminer quand arroser les plantes en fonction des données recueillies par des capteurs d'humidité, de température et d'humidité de l'air.

En utilisant ces données, le modèle de machine learning apprendra à prédire les besoins en eau des plantes, permettant ainsi d'alerter l'utilisateur lorsqu'il est temps d'arroser et de contrôler la pompe pour un arrosage à distance. De plus, ce jardin intelligent avertira l'utilisateur lorsque le sol devient sec et, en cas d'oubli d'arrosage, activera automatiquement l'arrosage. Cette approche contribuera à réduire le gaspillage d'eau tout en maintenant la santé des plantes.

## 2 Segment Clientèle

- Jardiniers et Amateurs de Plantes.
- Marché d'agriculture Tunisien.
- Entreprises de Paysagisme et Gestion d'Espaces Verts.

## 3 Besoins

### 3.1 Fonctionnel

- Suivi en Temps Réel : Le système doit pouvoir mesurer et afficher en temps réel les niveaux d'humidité du sol, de température et d'humidité de l'air.
- Prédiction des Besoins en Eau : Grâce au modèle de machine learning, le système doit prédire les moments optimaux pour arroser les plantes.
- Alertes Utilisateur : Le système doit alerter l'utilisateur via une notification lorsqu'il est temps d'arroser.
- Interface Utilisateur : Fournir une interface simple permettant de visualiser les données des capteurs, les alertes d'arrosage et le contrôle de la pompe à distance.

### 3.2 Non fonctionnel

- Efficacité Énergétique : Le système doit consommer le moins d'énergie possible, en particulier pour les capteurs et les communications sans fil.
- Fiabilité : Assurer un fonctionnement continu et des relevés de capteurs précis pour éviter les erreurs d'arrosage.

- Sécurité des Données : Protéger les données de mesure et d'utilisateur lors de leur stockage ou de leur envoi dans le cloud.
- Facilité d'utilisation : L'interface utilisateur doit être intuitive, même pour un utilisateur non technique.

## 4 Technologies Choisies

Les technologies choisies impliquent les technologies utilisées dans la partie Software et la partie Hardware.

### 4.1 Partie Software

Les différentes technologies logicielles que nous allons utiliser dans le développement de notre application sont les suivantes :

#### 4.1.1 Middleware

- **WildFly** : Anciennement connu sous le nom de JBoss Application Server, WildFly est un serveur d'applications Java EE open-source. Il est écrit en Java et distribué sous la licence GNU LGPL. Il peut être utilisé sur tout système d'exploitation disposant d'une machine virtuelle Java.
- **Mosquitto MQTT Broker** : Mosquitto est un courtier MQTT largement utilisé, servant d'intermédiaire pour un messaging efficace et fiable entre les dispositifs IoT et le système back-end.
- **JAX-RS** : L'API Java pour les services web RESTful est une interface de programmation Java permettant de créer des services web selon l'architecture REST.

#### 4.1.2 Backend

- **MongoDB** : Base de données orientée documents NoSQL utilisée pour stocker les données des utilisateurs. Elle est pratique et facile à utiliser avec Jakarta EE.
- **MQTT** : Un protocole léger de type publish-subscribe utilisé pour communiquer les données des capteurs à un courtier MQTT dans le cloud.

#### 4.1.3 Frontend

- **PWA (Progressive Web App)** : Une PWA est une application web multiplateforme qui offre une expérience similaire à celle d'une application native. Elle permet

de développer des applications pour plusieurs plateformes en utilisant des technologies web telles que HTML, CSS et JavaScript. Les PWAs sont faciles et rapides à coder, sans nécessiter de connaissances préalables en développement web.

#### 4.1.4 Intégration IoT

- **Node-RED** : Node-RED est un outil de programmation qui permet de connecter des dispositifs matériels, des API et des services en ligne.

#### 4.1.5 Intégration MLOps

- **Pipeline CI/CD** : Notre pipeline d'intégration continue et de déploiement continu (CI/CD), propulsé par Jenkins, est au cœur de notre stratégie MLOps.
- **Technologies de test** : Pour la validation des modèles et l'assurance qualité des données, nous utilisons PyTest et Selenium. Ces frameworks de test aident à vérifier la fonctionnalité et la fiabilité de notre système de surveillance de la santé.
- **Technologies de développement** : Nous utilisons Python et Jupyter Notebook pour le développement et l'expérimentation des modèles. Ces technologies offrent un environnement polyvalent et axé sur les données pour créer et affiner nos modèles de machine learning.
- **Surveillance et journalisation** : Notre système repose sur Prometheus et Grafana pour la surveillance et la journalisation. Ces outils nous permettent de suivre les performances de nos modèles de machine learning et de détecter toute anomalie.
- **Containerisation** : Les conteneurs Docker sont utilisés pour un déploiement et une mise à l'échelle efficaces, garantissant la cohérence et la facilité de gestion de notre système.
- **Traitement des flux** : Apache Kafka est utilisé pour gérer les flux de données en temps réel provenant des capteurs, permettant des mises à jour et des notifications opportunes dans notre système de surveillance de la santé.

## 4.2 Partie Hardware

- **Raspberry Pi 4** pour le contrôle du système.
- **DHT22** : capteur de température et d'humidité.
- **YL-69** : capteur d'humidité du sol.
- **Pompe** : pour l'irrigation.
- **Une LED** : pour les alertes.
- **Fils de connexion** .

- **Résistances de 220 Ohm** pour les LED.
- **Batterie** pour la carte Raspberry Pi.

## 5 Architecture

Cette architecture montre l'intégration entre différents composants pour la gestion d'un jardin intelligent. Une application PWA (Progressive Web App) sert d'interface utilisateur et communique avec un serveur basé sur Jakarta EE déployé sur WildFly via des appels API REST. Ce serveur effectue des requêtes vers une base de données MongoDB pour la gestion des données. Il interagit aussi avec un courtier MQTT via WebSocket pour envoyer et recevoir des messages. Le Raspberry Pi, connecté aux capteurs (comme la température et l'humidité) et aux actionneurs (comme la LED et la pompe), reçoit les instructions MQTT pour exécuter des actions d'arrosage automatique. Un pipeline d'apprentissage automatique hébergé sur le cloud permet de collecter, évaluer et entraîner des modèles pour optimiser l'arrosage et informer l'utilisateur.

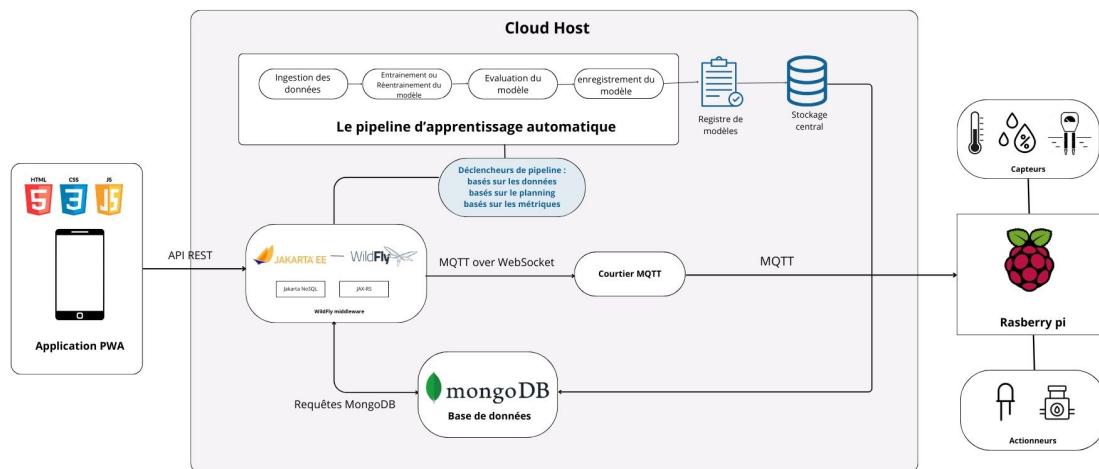


FIGURE 1 – Architecture du projet

## 6 Limites

En embarquant dans ce projet, il est nécessaire de prendre en compte les limitations qui pourraient survenir lors de la phase d'implémentation.

- **Fiabilité de la connexion** : Dans les zones rurales ou éloignées, l'absence de Wi-Fi stable peut limiter la connectivité entre les capteurs IoT, l'application mobile, et le cloud.

- **Précision des données collectées** : La fiabilité du système dépend de la précision des capteurs d'humidité et de température, qui peuvent varier avec l'usure ou selon les conditions environnementales.

- **Maintenance du matériel** : Les capteurs exposés aux conditions extérieures peuvent se dégrader rapidement en fonction de l'humidité, de la poussière, ou des variations climatiques, nécessitant des remplacements réguliers.

**Conditions climatiques** : Dans des environnements extrêmes (très chaud, très froid, forte humidité), les capteurs et les équipements peuvent rencontrer des défaillances ou des dysfonctionnements.

## 7 Modèle Commercial

### 7.1 Marketing Mix

#### 1/ Produit :

- Capteurs IoT pour mesurer l'humidité et la température du sol.
- Suivi en temps réel de l'état d'hydratation et de température des plantes via une application mobile.
- Prédiction des besoins d'irrigation.
- Amélioration continue de l'algorithme de prédiction à l'aide de MLOps pour optimiser la précision.
- Notifications envoyées à l'utilisateur en cas de besoin d'irrigation. Affichage de la distance entre l'utilisateur et les plantes.
- Services basés sur la localisation pour visualiser les emplacements des utilisateurs et des plantes.

#### 2/ Prix :

- Tarification à partir de la vente directe des kits IoT et d'un abonnement pour l'application mobile.
- Tarification compétitive adaptée aux jardiniers amateurs, petites exploitations agricoles et professionnels.
- Remises offertes for the clients achetant in large quantities.

#### 3/ Place (Distribution) :

- Application mobile accessible sur les plateformes web et les stores d'applications, facilement téléchargeable.
- Partenariats avec des jardinerie et des magasins d'équipements agricoles pour la vente des kits IoT.



### **3/ Promotion :**

- Utiliser une approche multi-canaux pour la promotion.
- Campagnes publicitaires ciblées sur les réseaux sociaux (Facebook, Instagram, LinkedIn) pour atteindre les jardiniers, agriculteurs et professionnels.
- Création de contenu éducatif (articles de blog, vidéos, tutoriels) pour sensibiliser aux avantages de l'irrigation intelligente.

## **7.2 Analyse SWOT**

### **1/ Forces :**

- Utilisation de l'IoT, de l'edge computing et de l'IA pour optimiser l'irrigation, ce qui réduit le gaspillage d'eau et améliore l'efficacité.
- Accès aux données d'humidité et de température via une application mobile, permettant une gestion instantanée.
- Application mobile conviviale avec notifications et contrôle à distance, rendant le système accessible même aux non-experts.
- Notifications et assistance en fonction des besoins spécifiques des plantes et de l'utilisateur.
- Contrôle à distance d'irrigation.

### **2/ Faiblesses :**

- Dépendance à une connexion internet pour les fonctionnalités cloud et à l'énergie pour les capteurs.
- Installation et configuration des capteurs et de l'application nécessitant potentiellement une assistance technique.
- Risque de panne ou d'usure des capteurs et de la connectivité IoT, nécessitant une régulière maintenance.

### **3/ Opportunités :**

- Répondre à la demande croissante de solutions économes en eau et respectueuses de l'environnement.
- Développement rapide de l'IoT dans l'agriculture et le jardinage, permettant de capter une clientèle nouvelle.
- Possibilité de collaborer avec des acteurs de l'agriculture, des jardineries et des organisations écologiques pour atteindre plus de clients.
- Évolutions dans le domaine du MLOps permettant d'améliorer les modèles de prédiction pour une irrigation encore plus précise.

#### **4/ Menaces :**

- Présence de solutions concurrentes d'irrigation intelligente sur le marché qui pourraient capter une part de la clientèle cible.
- Nécessité d'une connexion stable et d'un accès à l'énergie, pouvant poser problème dans certaines zones rurales.
- Réticence de certains agriculteurs et jardiniers à adopter de nouvelles technologies ou à investir dans des solutions innovantes.