

Cahier des charges pour le projet de serre agricole IoT

1. Introduction

1.1 Contexte

Le projet consiste en la création d'une serre agricole intelligente utilisant des technologies IoT et IA pour optimiser la croissance des plantes et détecter les maladies. L'application web (backend en Spring Boot et frontend en Angular) permettra de gérer les données collectées et d'envoyer des notifications en cas d'alerte. Une application mobile sera également développée pour fournir un accès en temps réel aux informations de la serre.

1.2 Objectifs

- Automatiser la gestion de la serre pour améliorer la production agricole.
- Intégrer un système de détection de maladies et de suivi de la croissance des plantes via l'IA.
- Fournir une application web pour la gestion et la visualisation des données.
- Développer une application mobile pour un accès facile et en temps réel aux informations.

2. Description du projet

2.1 Architecture générale

- Capteurs IoT : Mesurer les conditions de la serre (température, humidité, lumière, etc.).
- Module IA : Analyser les images des plantes pour détecter les maladies et suivre leur croissance.
- Backend (Spring Boot) : Gérer les données collectées, les analyser et les stocker.
- Frontend (Angular) : Interface utilisateur pour visualiser les données et gérer la serre.
- Application mobile : Accès en temps réel aux informations de la serre, notifications.

3. Spécifications fonctionnelles

3.1 Capteurs IoT

- Mesurer les conditions environnementales (température, humidité, lumière).

- Transmettre les données en temps réel au serveur backend.

3.2 Module IA

- Analyser les images des plantes pour détecter les maladies.
- Suivre la croissance des plantes.
- Générer des alertes en cas de détection de maladies.

3.3 Backend (Spring Boot)

- Collecter et stocker les données des capteurs.
- Traiter les données pour générer des statistiques et des rapports.
- Gérer les alertes et les notifications.

3.4 Frontend (Angular)

- Dashboard pour visualiser les données de la serre.
- Graphiques et rapports sur les conditions environnementales et la croissance des plantes.
- Interface de gestion des alertes et des notifications.

3.5 Application mobile

- Afficher les données en temps réel.
- Recevoir des notifications en cas d'alerte ou d'erreur.
- Interface utilisateur simplifiée pour un accès rapide aux informations essentielles.

4. Spécifications techniques

4.1 Capteurs IoT

- Utilisation de capteurs compatibles avec les microcontrôleurs (Arduino, Raspberry Pi).
- Communication via protocoles standard (MQTT, HTTP).

4.2 Module IA

- Utilisation de modèles de machine learning pour la détection des maladies.
- Entraînement des modèles sur un jeu de données approprié.

4.3 Backend (Spring Boot)

- Développement en Java avec Spring Boot.
- Base de données pour le stockage des données.
- API REST pour la communication avec le frontend et les capteurs IoT.

4.4 Frontend (Angular)

- Développement en TypeScript avec Angular.
- Utilisation de frameworks graphiques pour les visualisations de données (Chart.js, D3.js).

4.5 Application mobile

- Développement en Flutter pour une application cross-platform .
- Intégration avec le backend via API REST.

5. Gestion des alertes et des notifications

- Système de notification en temps réel pour les alertes (email, application web, application mobile).
- Gestion des seuils d'alerte pour les conditions environnementales et la détection des maladies.