

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'Informatique



MÉMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Réseaux et Multimédia et Informatique Décisionnelle

THÈME

**Conception et mise en place d'un système
mobile pour l'agriculture intelligente :
détection automatisée et recommandations
par modèles d'intelligence artificielle.**

Présenté par :

Memoud Abdellahi

Belkheir Mohamed Salem

Soutenu publiquement le : 19/06/2025

Devant le jury composé de :

Président : Dr Attia Abdelouaheb

Examineur : Dr Belhadj Foudil

Encadreur : Dr Adoui Ibtissem

Co-Encadreur : Dr Mostefai Messaoud

Année Universitaire 2024/2025

REMERCIEMENTS

Avant tout, nous remercions Dieu, le Tout-Puissant, de nous avoir donné la force, la patience et l'opportunité de mener à bien ce travail.

Nous exprimons notre profonde gratitude à nos encadreurs, M. Mostefai Messoaud et Mme Adoui Ibtissem, pour leur encadrement de qualité, leur disponibilité constante, leurs conseils avisés et leur soutien indéfectible tout au long de ce projet. Leur générosité, leur rigueur et leur accompagnement bienveillant ont été essentiels à la réalisation de ce mémoire.

Nous remercions également l'ensemble de nos enseignants pour les compétences et les connaissances qu'ils nous ont transmises durant notre formation. Leur dévouement et leur implication ont largement contribué à notre réussite.

Enfin, nous adressons nos sincères remerciements à toutes les personnes qui, de près ou de loin, ont soutenu notre travail et nous ont encouragés tout au long de notre parcours.

DÉDICACE

Nous tenons à dédier ce travail à toutes les personnes qui ont été à notre côté durant cette étape importante de notre vie. Tout d'abord, nous remercions nos chers parents pour leur amour et leur soutien constant, ainsi que pour leur encouragement à poursuivre nos rêves. Nous tenons également à remercier les professeurs et conseillers académiques pour leur mentorat et leur expertise qui ont été précieux tout au long de ce parcours professionnel. Leurs enseignements ont été une source d'inspiration qui nous ont permis de réaliser nos pleins potentiels.

Enfin, nous dédions ce travail à nos amis et camarades de classe qui ont partagé ce parcours avec nous. Leurs encouragements, leur aide et leur amitié ont été une source de motivation constante. Cette réussite est celle de toute personne qui en a contribué et à lui nous sommes reconnaissants.

LISTE DES ACRONYMES

IA	Intelligence Artificielle
IoT	Internet des Objets (Internet of Things)
Wi-Fi	Wireless Fidelity
GSM	Global System for Mobile Communications
3G	Troisième Génération de téléphonie mobile
4G	Quatrième Génération de téléphonie mobile
Agritech	Technologies appliquées à l'agriculture
MIT	Massachusetts Institute of Technology
RFID	Radio Frequency Identification
BLE	Bluetooth Low Energy
CNN	Convolutional Neural Network
TFLite	TensorFlow Lite
NB-IoT	Narrowband Internet of Things

RÉSUMÉ

Ce mémoire présente la conception et la mise en œuvre d'un système intelligent de surveillance agricole et météorologique, basé sur l'intégration de l'Internet des objets (IoT). Le système se compose d'un module émetteur (Arduino, capteurs, et actionneurs) et d'un module récepteur (ESP32), ainsi que d'applications mobiles Android et iOS pour la surveillance des données via Firebase. Le système utilise des données météorologiques issues des capteurs IoT, en s'appuyant également sur des APIs telles que OpenWeather et Tomorrow.io pour bénéficier de données supplémentaires plus précises, essentielles à la complexité de ce projet. Deux modèles d'intelligence artificielle ont été intégrés : l'un pour la détection des maladies des plantes à partir d'images, et l'autre pour fournir des recommandations agricoles intelligentes basées sur les prévisions météorologiques.

Mots-clés : IoT, capteurs sans fil, Firebase, OpenWeather, Tomorrow.io, intelligence artificielle, agriculture intelligente.

ABSTRACT

This thesis presents the design and implementation of an intelligent agricultural and weather monitoring system, based on the integration of the Internet of Things (IoT). The system consists of a transmitter module (Arduino, sensors, and actuators) and a receiver module (ESP32), along with Android and iOS mobile applications for data monitoring via Firebase. It uses weather data collected from IoT sensors, combined with additional data from APIs such as OpenWeather and Tomorrow.io to enhance precision and support the system's complexity. Two AI models are integrated: one for detecting plant diseases from images, and another for generating intelligent agricultural recommendations based on weather forecasts.

Keywords: IoT, wireless sensors, Firebase, OpenWeather, Tomorrow.io, artificial intelligence, smart farming.

TABLE DES MATIÈRES

Remerciements	i
Dedicaces	ii
Liste des acronymes	iii
Résumé	iv
Abstract	v
Introduction Générale	1
1 Contexte Générale du projet	3
1.1 Introduction	4
1.2 Présentation du projet	4
1.2.1 Problématique	4
1.2.2 Objectifs visés	4
1.2.3 Solutions proposées	5
1.3 L’agriculture intelligente	6
1.3.1 Définition	6

1.3.2	Objectifs et enjeux	6
1.3.3	Cas d'usage réels	7
1.4	L'Internet des Objets (IoT)	7
1.4.1	Principes de fonctionnement	8
1.4.2	Architecture d'un réseau IoT appliqué au monde agricole	10
1.5	Les technologies de communication sans fil	11
1.5.1	LPWAN (Low Power Wide Area Network)	11
1.5.2	Wi-Fi	12
1.5.3	Bluetooth Low Energy (BLE)	12
1.5.4	Zigbee	12
1.6	Conclusion	13
2	Analyse et Conception	14
2.1	Introduction	15
2.2	Spécification des besoins	15
2.3	Les besoins fonctionnels	15
2.3.1	Le Fonctionnement	16
2.4	Besoins non fonctionnels	16
2.5	Conception des modèles IA	17
2.5.1	Modèle de détection de maladies (CNN)	17
2.5.2	Description du jeu de données	17
2.5.3	Conception et choix du modèle	18
2.5.4	Entraînement du modèle	19
2.5.5	Évaluation du modèle	19
2.5.6	Modèle de recommandation	20
2.5.7	Intégration de modèles dans l'application mobile	21

2.6	Conception de l'infrastructure IoT	21
2.6.1	Les capteurs IoT utilisés	21
2.7	Conception de l'application mobile	23
2.7.1	Méthodologie de modélisation adoptée	23
2.7.2	Le langage de modélisation UML	23
2.7.3	Pourquoi choisir UML ?	23
2.7.4	Diagrammes UML	24
2.7.5	Diagramme de cas d'utilisation	25
2.7.6	Digramme de classe	26
2.7.7	Diagrammes de séquence	27
2.7.8	Diagramme de séquence : Authentification	27
2.7.9	Diagramme de séquence :Détection des maladies des plantes	28
2.7.10	Diagramme d'activités	29
2.8	Conclusion	29
3	Implémentation	30
3.1	Introduction	31
3.2	Environnement de travail	31
3.2.1	Environnement Matériel	31
3.2.2	Environnement Logiciel	31
3.3	Implémentation	36
3.3.1	Inscription et Authentification	36
3.3.2	Page d'accueil	37
3.3.3	Détection des maladies	37
3.3.4	Météo et Recommendations IA	38
3.4	Conclusion	40

Conclusion Générale	41
Bibliographie	41

TABLE DES FIGURES

1.1	IoT en agriculture	9
1.2	Architecture du réseau IoT	11
1.3	Différentes technologies de communication sans fil	13
2.1	Capteur d'humidité du sol	22
2.2	Capteur DHTT22	22
2.3	Table ESP-WROOM-32	23
2.4	Diagramme de cas d'utilisation	26
2.5	Diagramme de classe	27
2.6	Diagramme de séquence : Authentification	28
2.7	Détection des maladies des plantes	28
2.8	Diagramme d'activités pour la connexion	29
2.9	Diagramme d'activités pour la détection des maladies des plantes	29
3.1	Logo Kaggle	31
3.2	Logo Google Colab	32
3.3	Logo Flutter	32
3.4	Logo Fritzing	32

3.5	Logo Arduino IDE	33
3.6	Logo Vs Code IDE	33
3.7	Logo Android Studio IDE	33
3.8	Logo Figma	34
3.9	Logo GitHub	34
3.10	Logo SDK	34
3.11	Logo UML	35
3.12	Logo OpenWeather	35
3.13	Logo Tomorrow.io	35
3.14	Interface d'authentification	36
3.15	Interface d'inscription	36
3.16	Page d'accueil	37
3.17	Examiner les plantes	38
3.18	Resultat de détection	38
3.19	Examiner les plantes	39
3.20	Resultat de détection	39

LISTE DES TABLEAUX

2.1	Paramètres d'entraînement du modèle	19
2.2	Résultats obtenus	20
2.3	Diagrammes UML	25

INTRODUCTION GÉNÉRALE

L'agriculture représente l'un des piliers fondamentaux de l'économie mondiale, particulièrement dans les pays en développement où elle constitue la principale source de revenu et d'alimentation pour une grande partie de la population.

D'autre part, les agriculteurs sont confrontés à de nombreux défis majeurs qui compromettent à la fois la productivité agricole et la gestion durable des ressources naturelles. Parmi ces défis figurent le manque d'informations en temps réel sur l'état des cultures, la difficulté à anticiper les besoins en irrigation, l'incapacité à détecter précocement les maladies des plantes, ainsi qu'une gestion inefficace des ressources telles que l'eau, les engrais ou les pesticides. Ces contraintes freinent considérablement l'efficacité des exploitations agricoles, en particulier dans les zones rurales peu mécanisées et mal desservies en solutions technologiques adaptées.

Face à ces problématiques, nous avons entrepris de développer une application mobile intelligente, intégrant des technologies avancées telles que l'Internet des Objets (IoT) et l'intelligence artificielle (IA). L'objectif de notre projet est de proposer une solution innovante et accessible permettant aux agriculteurs de surveiller leurs cultures en temps réel, de recevoir des recommandations précises, et de prendre des décisions éclairées en vue d'améliorer leur productivité tout en préservant les ressources naturelles.

Notre application permet d'exploiter les données collectées automatiquement à partir de capteurs IoT (tels que ceux mesurant l'humidité du sol ou la température ambiante), combinées à des modèles prédictifs d'intelligence artificielle. Ces modèles sont capables, d'une part, de détecter les maladies des plantes à partir d'images fournies par l'utilisateur, et d'autre part, de générer des recommandations d'irrigation intelligentes en analysant les conditions environnementales et les besoins spécifiques des cultures.

Grâce à cette plateforme mobile, les agriculteurs disposent d'un outil centralisé, intuitif et

performant pour surveiller l'état de leurs champs, recevoir des alertes pertinentes, et optimiser la gestion de leur exploitation. L'application contribue ainsi à réduire les pertes, à économiser les ressources hydriques, et à renforcer la résilience des pratiques agricoles face aux aléas climatiques.

CHAPITRE 1

CONTEXTE GÉNÉRALE DU PROJET

1.1 Introduction

Ce chapitre présente le cadre théorique et technologique dans lequel s'inscrit notre projet. Nous y décrivons la problématique rencontrée dans le domaine agricole, les objectifs poursuivis, ainsi que les solutions que nous proposons. Ensuite, nous introduisons les notions fondamentales liées à l'agriculture intelligente, à l'Internet des Objets, aux technologies de communication sans fil, et à l'intelligence artificielle, afin de mieux comprendre les fondements de notre application.

1.2 Présentation du projet

1.2.1 Problématique

Des milliers d'agriculteurs à travers le monde, en particulier dans les zones rurales, ne disposent pas de moyens technologiques avancés leur permettant de surveiller en temps réel l'état de leurs cultures ou d'interagir avec des systèmes intelligents capables de les assister dans la gestion de leurs exploitations. Ce manque d'outils d'aide à la décision empêche une réactivité adéquate face à des facteurs critiques comme la sécheresse, les maladies des plantes ou les changements climatiques soudains.

Les données essentielles (telles que l'humidité du sol, la température ou l'état sanitaire des cultures) ne sont souvent ni collectées, ni analysées à temps, ce qui entraîne des pertes de rendement notables. De plus, la difficulté à identifier avec précision les besoins spécifiques des cultures ou à détecter précocement les maladies limite fortement la capacité des agriculteurs à intervenir efficacement. Ce constat met en évidence la nécessité d'un système intelligent, autonome et accessible, permettant une gestion agricole plus réactive, durable et basée sur des données fiables.

1.2.2 Objectifs visés

Ce projet a pour objectif principal de renforcer la productivité agricole et d'optimiser la gestion des ressources naturelles grâce à l'intégration de technologies intelligentes accessibles aux agriculteurs.

Cette application que nous avons conçue, se veut simple, intuitive et adaptée au contexte rural. Elle permet aux agriculteurs de suivre en temps réel l'état de leurs cultures, d'accéder à des analyses intelligentes basées sur des données environnementales collectées par des capteurs

IoT, et de recevoir des recommandations personnalisées, notamment en matière d'irrigation ou de détection de maladies.

Un accent particulier a été mis sur la fiabilité des données, la précision des prédictions générées par les modèles d'intelligence artificielle, ainsi que sur la facilité d'usage de l'interface mobile, même pour des utilisateurs non technophiles. De plus, l'application a été conçue pour fonctionner hors ligne, avec des modèles embarqués, assurant ainsi une accessibilité optimale même dans les zones à faible connectivité.

En somme, ce projet vise à mettre en place un outil intelligent, pratique et durable, capable de transformer les pratiques agricoles traditionnelles en offrant aux agriculteurs une aide à la décision efficace, automatisée et en temps réel, tout en contribuant à la réduction des pertes, à l'économie d'eau, et à l'amélioration globale du rendement des exploitations.

1.2.3 Solutions proposées

Ce projet vise à concevoir et développer une application mobile intelligente dédiée à l'optimisation de la gestion agricole. L'application est destinée à accompagner les agriculteurs dans leur travail quotidien en leur offrant un accès facilité à des données environnementales critiques, à des analyses intelligentes et à des recommandations personnalisées pour une prise de décision rapide et efficace.

L'application permet notamment de collecter automatiquement les données issues de capteurs IoT placés dans les champs (tels que les capteurs d'humidité du sol ou de température ambiante), et de les analyser localement à l'aide de modèles d'intelligence artificielle embarqués. Ces modèles ont pour mission de détecter les maladies des plantes à partir d'images prises par l'utilisateur et de recommander des actions d'irrigation intelligentes en fonction des conditions mesurées et du type de culture.

En complément de ces fonctionnalités principales, l'application offre également à l'agriculteur :

- Une interface de suivi en temps réel des données collectées par les capteurs.
- Un système de notifications intelligentes pour l'alerter des anomalies ou des besoins spécifiques (maladie détectée, manque d'eau...).
- Un historique des actions et recommandations pour améliorer le suivi de l'exploitation sur le long terme.
- Elle doit se fonctionner même en mode hors ligne, cette approche permet d'atteindre des exploitants agricoles évoluant dans des environnements à faible connectivité tout en leur offrant des outils de pointe pour améliorer la productivité, économiser les ressources et

renforcer la durabilité de leurs pratiques.

1.3 L'agriculture intelligente

1.3.1 Définition

L'agriculture intelligente (ou smart agriculture) désigne une approche moderne de la production agricole qui intègre des technologies avancées telles que l'Internet des Objets (IoT), l'intelligence artificielle (IA), la robotique, les drones et le cloud computing. Cette approche vise à optimiser la gestion des exploitations agricoles en collectant et en analysant des données en temps réel, afin d'améliorer la productivité, la durabilité et la résilience des systèmes agricoles face aux défis environnementaux et climatiques.

1.3.2 Objectifs et enjeux

L'agriculture intelligente vise à transformer les pratiques agricoles traditionnelles en intégrant des technologies avancées telles que l'Internet des Objets (IoT), l'intelligence artificielle (IA), et le cloud computing. Cette approche répond à des objectifs multiples et soulève des enjeux majeurs pour le secteur agricole.

→ Objectifs principaux

1. Augmenter durablement la productivité agricole

En optimisant l'utilisation des ressources et en améliorant les rendements, l'agriculture intelligente contribue à assurer la sécurité alimentaire mondiale.

2. Renforcer la résilience face aux changements climatiques

Grâce à une meilleure gestion des risques climatiques et à l'adaptation des pratiques agricoles, elle permet de maintenir la production malgré les aléas climatiques.

3. Réduire l'impact environnemental de l'agriculture

En minimisant les émissions de gaz à effet de serre et en favorisant une utilisation efficace des intrants, elle participe à la protection de l'environnement.

→ Enjeux majeurs

1. Sécurité alimentaire mondiale

Avec une population mondiale en croissance, il est impératif d'augmenter la production alimentaire de manière durable pour répondre à la demande future.

2. Gestion durable des ressources naturelles

L'agriculture intelligente permet une utilisation plus efficace de l'eau, des sols et des intrants, réduisant ainsi la pression sur les ressources naturelles.

3. Intégration technologique et accessibilité

Le déploiement de technologies avancées dans les zones rurales nécessite des infrastructures adaptées et une formation des agriculteurs pour assurer une adoption efficace.

4. Protection des données et cybersécurité

La collecte et l'analyse de données agricoles sensibles soulèvent des questions de confidentialité et de sécurité des informations.

1.3.3 Cas d'usage réels

L'agriculture intelligente transforme les pratiques agricoles en intégrant des technologies avancées telles que l'Internet des Objets (IoT), l'intelligence artificielle (IA), et le cloud computing. Voici quelques exemples concrets de son application :

1. Irrigation intelligente basée sur l'IoT

Des capteurs connectés mesurent en temps réel l'humidité du sol, la température et d'autres paramètres environnementaux. Ces données permettent d'optimiser l'irrigation, en fournissant la quantité d'eau nécessaire au moment opportun, réduisant ainsi le gaspillage et améliorant les rendements.

2. Automatisation de la récolte et tri des produits

Des robots agricoles, équipés de bras articulés et de systèmes de vision par ordinateur, sont capables de récolter des fruits et légumes avec précision, réduisant la dépendance à la main-d'œuvre et augmentant l'efficacité. De plus, des systèmes automatisés peuvent trier les produits selon leur taille, couleur ou maturité.

3. Gestion intelligente des serres

Des systèmes automatisés contrôlent les conditions environnementales à l'intérieur des serres, tels que la température, l'humidité et la luminosité, en fonction des besoins spécifiques des cultures. Cela permet d'optimiser la croissance des plantes et de réduire la consommation d'énergie.

1.4 L'Internet des Objets (IoT)

L'Internet des Objets (IoT) désigne un réseau interconnecté d'objets physiques — tels que des capteurs, des appareils, des véhicules ou des équipements agricoles — capables de collecter, transmettre et traiter des données via des réseaux de communication, sans intervention humaine

directe. Ces objets sont dotés de capteurs, de logiciels et d'autres technologies leur permettant d'interagir entre eux et avec leur environnement, transformant ainsi des éléments physiques en sources actives d'information.

Le terme « Internet des Objets » a été introduit en 1999 par Kevin Ashton, chercheur au Massachusetts Institute of Technology (MIT), dans le cadre de ses travaux sur l'identification par radiofréquence (RFID) pour améliorer la gestion des chaînes d'approvisionnement. Il a envisagé un système où les objets physiques pourraient être connectés à Internet pour transmettre des données, permettant ainsi une gestion plus efficace des ressources et des processus.

En 2005, l'Union Internationale des Télécommunications (UIT) a officiellement reconnu l'IoT comme une composante essentielle de l'infrastructure technologique moderne, soulignant son potentiel à transformer divers secteurs, y compris l'agriculture, la santé, l'industrie et les transports.

Dans le domaine de l'agriculture intelligente, l'IoT joue un rôle crucial en permettant la surveillance en temps réel des conditions environnementales, l'automatisation des systèmes d'irrigation, la détection précoce des maladies des cultures et l'optimisation de l'utilisation des ressources. Par exemple, des capteurs peuvent mesurer l'humidité du sol, la température de l'air et d'autres paramètres essentiels, fournissant ainsi aux agriculteurs des données précises pour prendre des décisions éclairées et améliorer la productivité tout en réduisant l'impact environnemental.

1.4.1 Principes de fonctionnement

L'Internet des Objets (IoT) en agriculture repose sur l'intégration de capteurs, d'appareils connectés et de systèmes de communication pour collecter, transmettre et analyser des données en temps réel. Ces technologies permettent aux agriculteurs de surveiller et de gérer efficacement leurs exploitations, en optimisant l'utilisation des ressources et en améliorant la productivité.

1. Collecte de données via des capteurs

Des capteurs sont déployés sur le terrain pour mesurer divers paramètres essentiels à la culture, tels que :

- ★ Humidité du sol
- ★ Température et humidité de l'air
- ★ etc

2. Transmission des données

Les données recueillies par les capteurs sont transmises via des réseaux de communication sans fil, tels que :

- ★ LoRaWAN
- ★ GSM/3G/4G
- ★ Wi-Fi

Ces réseaux assurent la transmission fiable des données vers des plateformes de traitement, même dans des zones rurales éloignées.

3. Traitement et analyse des données

Les données transmises sont stockées et analysées sur des plateformes cloud ou des serveurs locaux. Des algorithmes d'intelligence artificielle et de machine learning peuvent être appliqués pour :

- ★ Détecter des anomalies ou des tendances
- ★ Prédire les besoins en irrigation ou en fertilisation
- ★ Identifier les signes précoces de maladies

Cette analyse permet une prise de décision rapide et éclairée.

4. Prise de décision et action

Les informations analysées sont présentées aux agriculteurs via des interfaces conviviales, telles que des applications mobiles ou des tableaux de bord. Cela leur permet de :

- ★ Surveiller l'état de leurs cultures en temps réel
- ★ Recevoir des alertes et des recommandations personnalisées
- ★ Automatiser certaines actions, comme l'irrigation ou la fertilisation

En intégrant ces technologies, l'IoT transforme l'agriculture en un système intelligent, réactif et durable.

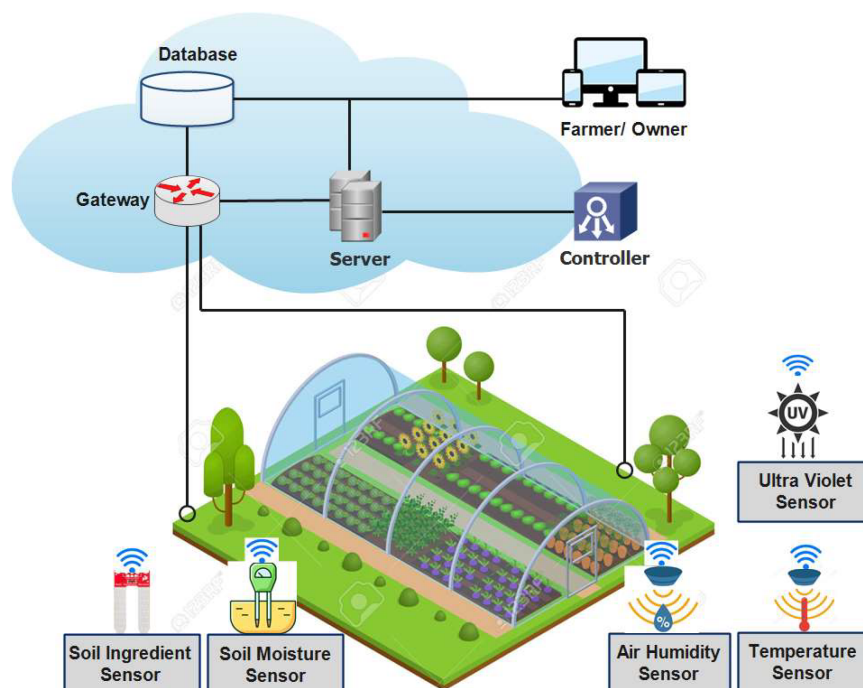


FIGURE 1.1 – IoT en agriculture

1.4.2 Architecture d'un réseau IoT appliqué au monde agricole

L'architecture d'un réseau IoT en agriculture repose sur une structure en couches, permettant une collecte, une transmission et une analyse efficaces des données issues des exploitations agricoles. Cette architecture facilite la prise de décisions éclairées et l'automatisation de certaines tâches, contribuant ainsi à une agriculture plus durable et productive.

1. Couche de perception (capteurs et actionneurs)

Cette couche comprend les dispositifs physiques qui interagissent directement avec l'environnement agricole :

- ★ Capteurs : mesurent des paramètres tels que l'humidité du sol, la température, la luminosité, le pH, la salinité, etc.
- ★ Actionneurs : dispositifs capables d'agir sur l'environnement, comme les systèmes d'irrigation automatisés ou les distributeurs d'engrais.

Ces dispositifs collectent des données en temps réel, fournissant une vision précise des conditions de l'exploitation.

2. Couche de réseau (transmission des données)

Les données recueillies par les capteurs sont transmises via des réseaux de communication sans fil vers des centres de traitement. Les technologies couramment utilisées incluent :

- ★ LoRaWAN : réseau longue portée et basse consommation, idéal pour les zones rurales étendues.
- ★ GSM/3G/4G/5G : réseaux cellulaires offrant une large couverture et des débits élevés.
- ★ Wi-Fi : adapté aux zones avec une infrastructure existante.

Ces réseaux assurent une transmission fiable des données, même dans des environnements éloignés ou difficiles d'accès.

3. Couche de traitement (serveurs et plateformes cloud)

Une fois les données transmises, elles sont stockées et analysées sur des serveurs locaux ou des plateformes cloud. Cette couche permet :

- ★ Stockage des données : conservation sécurisée des informations collectées.
- ★ Analyse des données : utilisation d'algorithmes d'intelligence artificielle et de machine learning pour extraire des informations pertinentes.
- ★ Visualisation : présentation des données sous forme de tableaux de bord ou de graphiques pour une interprétation aisée.

Ce traitement permet d'identifier des tendances, de détecter des anomalies et de prévoir des besoins futurs.

4. Couche d'application (interfaces utilisateur)

Les résultats de l'analyse sont accessibles aux agriculteurs via des interfaces conviviales, telles que :

- ★ Applications mobiles : permettent une consultation en temps réel des données et la réception d'alertes.
- ★ Tableaux de bord web : offrent une vue d'ensemble de l'exploitation et des recommandations personnalisées.



FIGURE 1.2 – Architecture du réseau IoT

1.5 Les technologies de communication sans fil

Dans le contexte de l'agriculture intelligente, la transmission efficace des données collectées par les capteurs est essentielle. Plusieurs technologies de communication sans fil sont utilisées pour assurer cette transmission, chacune ayant ses avantages et inconvénients en fonction des besoins spécifiques des exploitations agricoles.

1.5.1 LPWAN (Low Power Wide Area Network)

Les réseaux LPWAN sont conçus pour permettre la communication à longue portée avec une faible consommation d'énergie, ce qui les rend particulièrement adaptés aux environnements

agricoles étendus.

- ★ LoRaWAN (Long Range Wide Area Network) : Cette technologie permet une communication sur de longues distances (jusqu'à 15 km en zone rurale) avec une faible consommation d'énergie. Elle est idéale pour les capteurs déployés dans des zones éloignées sans accès facile à l'électricité. LoRaWAN est souvent utilisée pour la surveillance de l'humidité du sol, la température et d'autres paramètres environnementaux.
- ★ NB-IoT (Narrowband Internet of Things) : Utilisant les réseaux cellulaires existants, il offre une bonne couverture et une pénétration efficace dans les bâtiments. Elle est adaptée aux applications nécessitant une transmission de données peu fréquente mais fiable, comme la surveillance des silos ou des réservoirs d'eau.

1.5.2 Wi-Fi

Le Wi-Fi offre des débits de données élevés sur de courtes distances. Il est principalement utilisé dans les environnements agricoles où une infrastructure réseau est déjà en place, comme dans les serres ou les bâtiments d'élevage. Cependant, sa portée limitée et sa consommation d'énergie relativement élevée peuvent restreindre son utilisation en plein champ.

1.5.3 Bluetooth Low Energy (BLE)

BLE est une technologie de communication à courte portée avec une faible consommation d'énergie. Elle est adaptée aux applications nécessitant une communication entre des appareils proches, comme la surveillance du bétail ou la collecte de données à partir de capteurs portables.

1.5.4 Zigbee

Zigbee est une technologie de communication sans fil à faible consommation d'énergie, adaptée aux réseaux maillés. Elle est utilisée dans les applications agricoles nécessitant une communication entre plusieurs capteurs répartis sur une zone relativement restreinte, comme dans les serres.

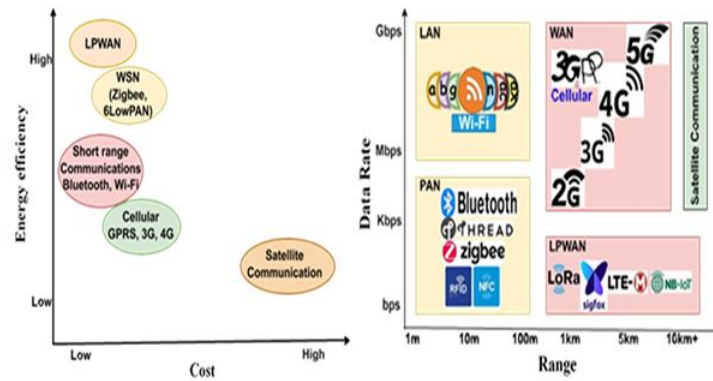


FIGURE 1.3 – Différentes technologies de communication sans fil

En résumé, le choix de la technologie de communication sans fil dépend des besoins spécifiques de l'exploitation agricole, notamment en termes de portée, de consommation d'énergie, de débit de données et de coût. Une combinaison de plusieurs technologies peut être nécessaire pour répondre aux différentes exigences des applications agricoles intelligentes

1.6 Conclusion

Ce premier chapitre a permis d'établir le cadre général de notre projet en mettant en lumière les défis majeurs auxquels sont confrontés les agriculteurs, notamment le manque d'outils technologiques adaptés pour une gestion efficace et durable de leurs exploitations. Nous avons identifié les problématiques spécifiques liées à la surveillance des cultures, à la détection précoce des maladies et à l'optimisation de l'irrigation. Pour répondre à ces enjeux, nous avons proposé une solution innovante : une application mobile intelligente intégrant des technologies avancées telles que l'Internet des Objets (IoT) et l'intelligence artificielle (IA).

Le chapitre suivant sera consacré à l'analyse détaillée et à la spécification de notre application, afin de traduire les besoins identifiés en exigences fonctionnelles et techniques concrètes.

CHAPITRE 2

ANALYSE ET CONCEPTION

2.1 Introduction

Dans le chapitre précédent, nous avons défini notre champ d'étude et ses limites, nous avons aussi présenté la solution préconisée. Nous présentons dans ce chapitre l'étape importante qui consiste à concevoir notre projet « Conception et mise en place d'une application pour la gestion des cas d'urgences ». Nous commencerons par la présentation d'UML, Par la suite nous présenterons une vue qui exprime les modules et les exécutable physiques sans aller à la réalisation concrète du système. Nous allons mettre l'accent sur les diagrammes de cas d'utilisation, classe, séquence et activités.

2.2 Spécification des besoins

Exigences anticipées de l'application : L'application envisagée doit satisfaire aux exigences fonctionnelles à exécuter par le système et aux exigences non fonctionnelles qui améliorent la qualité du logiciel système.

2.3 Les besoins fonctionnels

Les besoins fonctionnels, ou exigences fonctionnelles, représentent les actions qu'un système doit effectuer et ne fonctionne que si ces actions sont satisfaites. Cette application doit répondre aux exigences fonctionnelles suivantes :

- ★ Collecte et affichage des données environnementales(Intégrer des capteurs IoT pour mesurer des paramètres tels que l'humidité du sol, la température ambiante, la luminosité, etc).
- ★ Afficher en temps réel les données collectées.
- ★ Détecter des maladies des plantes.
- ★ Permettre à l'agriculteur de capturer des images des cultures
- ★ Utiliser des modèles d'intelligence artificielle pour analyser les images et détecter la présence éventuelle de maladies.
- ★ Analyser les données environnementales pour estimer les besoins en eau des cultures.
- ★ Fournir des recommandations personnalisées sur les volumes et les fréquences d'irrigation.
- ★ Envoyer des notifications à l'utilisateur en cas d'anomalies détectées (par exemple, humidité du sol insuffisante, température excessive, détection de maladies).

- ★ Conserver un historique des données collectées, des actions entreprises et des recommandations fournies.

2.3.1 Le Fonctionnement

L'agriculteur accède à l'application via son smartphone et une authentification sécurisée lui est requise pour garantir la confidentialité des données et personnaliser les services offerts. De l'autre part, des capteurs IoT installés dans les champs mesurent en continu des paramètres essentiels tels que l'humidité du sol, la température ambiante, la luminosité et la qualité de l'air. Ces données sont transmises à l'application pour une analyse approfondie. L'agriculteur, peut prendre des images des plantes pour détecter des signes précoces de maladies en se basant sur des modèles IA intégrés dans l'application qui peuvent aussi recommander des stratégies d'irrigation optimales en fonction des conditions environnementales.

En cas de détection d'anomalies ou de besoins spécifiques (par exemple, stress hydrique, risque de maladie), l'application envoie des notifications instantanées à l'agriculteur, lui permettant de prendre des mesures correctives rapidement.

2.4 Besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient ses contraintes internes et externes. Les besoins non fonctionnels que notre application doit y répondre sont :

- ★ La fiabilité : l'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.
- ★ Les erreurs : les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre application.
- ★ La convivialité : le système doit avoir une interface conviviale permettant un dialogue simple entre l'utilisateur et l'application.
- ★ La sécurité : le système doit être fiable et sécurisé, les utilisateurs seront impérativement routés vers la partie d'authentification afin d'empêcher les accès non permis.
- ★ La maintenabilité : le code doit être compréhensible par simple lecture, notamment en respectant les règles de gestion et les normes de développement.
- ★ L'exploitabilité : les impacts, en termes de performances, doivent être pris en compte lors de développement, ainsi que la consommation des ressources (CPU, mémoire, etc.) qui doit être minimisée.

2.5 Conception des modèles IA

2.5.1 Modèle de détection de maladies (CNN)

La détection des maladies des plantes à partir d'images est un enjeu crucial dans l'agriculture intelligente. Dans notre projet, nous avons conçu et entraîné un modèle CNN (de réseau de neurones convolutif) pour identifier des maladies sur les feuilles des plantes, en se basant sur le dataset PlantVillage, une base de données open-source largement utilisée pour la recherche en phytopathologie assistée par l'IA.

2.5.2 Description du jeu de données

2.5.2.1 Source des données

Le jeu de données utilisé pour ce travail est PlantVillage, une base de données publique bien connue dans la recherche agronomique et l'apprentissage automatique. Il est disponible sur Kaggle et contient :

- Plus de 54 000 images annotées de feuilles de plantes
- Couvre 38 classes : combinaisons de 14 espèces de plantes et de leurs maladies (ou état sain)
- Résolution variable, format JPEG, fond neutre dans la majorité des cas

2.5.2.2 Structure du dataset

Chaque image est étiquetée selon la plante et la maladie correspondante, par exemple :

- Tomato___ / Late / _ / blight
- Potato___ / Healthy
- Apple___ / Black / _ / rot

Ces étiquettes sont converties en entiers pour l'entraînement (via encodage one-hot pour les sorties multi-classes).

2.5.2.3 Prétraitement des données

Avant l'entraînement, plusieurs étapes de prétraitement ont été effectuées :

- Redimensionnement des images : Toutes les images ont été converties au format 224x224 pixels pour correspondre aux dimensions attendues par les réseaux pré-entraînés.
- Normalisation : Les valeurs des pixels ont été ramenées dans l'intervalle [0, 1] en divisant chaque pixel par 255.
- Encodage des labels : Encodage one-hot des 38 classes.

2.5.2.4 Augmentation des données

L'augmentation des données a été utilisée pour améliorer la robustesse du modèle et limiter le surapprentissage. Nous avons appliqué, avec des probabilités aléatoires :

- Rotations ($\pm 25^\circ$)
- Zoom (jusqu'à 20)
- Inversions horizontales
- Décalages verticaux et horizontaux
- Légères modifications de luminosité

Cette étape a été réalisée en temps réel pendant l'entraînement avec ImageDataGenerator de Keras.

2.5.3 Conception et choix du modèle

2.5.3.1 Apprentissage par transfert (Transfer Learning)

Nous avons opté pour un modèle basé sur le transfer learning, en utilisant une architecture pré-entraînée sur ImageNet, ce qui permet d'exploiter des caractéristiques visuelles génériques déjà apprises. Le modèle choisi est :

ResNet50, un réseau convolutionnel profond de 50 couches, connu pour ses bonnes performances et sa gestion efficace du gradient via les skip connections.

2.5.3.2 Architecture du modèle final

L'architecture complète est la suivante :

1. Base convolutionnelle : ResNet50 sans sa couche de classification finale (`include_top=False`)
2. Couches supplémentaires ajoutées :
 - GlobalAveragePooling2D
 - Dense(512, activation='relu')

- Dropout(0.5)
- Dense(38, activation='softmax') → pour la classification multi-classes

L'ensemble du modèle a été compilé dans Keras.

2.5.4 Entraînement du modèle

2.5.4.1 Configuration technique

- Langage : Python 3.10
- Framework : TensorFlow 2.x + Keras
- Environnement : Google Colab Pro (GPU NVIDIA Tesla T4)
- Durée de l'entraînement : Environ 45 minutes pour 30 époques

2.5.4.2 Paramètres d'entraînement

TABLE 2.1 – Paramètres d'entraînement du modèle

Paramètre	Valeur
Optimiseur	Adam (learning rate = 0.0001)
Fonction de perte	Categorical Crossentropy
Taille de batch	32
Nombre d'époques	30
Stratégie de validation	15% des données d'entraînement

2.5.4.3 Callbacks utilisés

- EarlyStopping : arrêt précoce si la validation n'évolue plus après 5 époques
- ModelCheckpoint : sauvegarde du modèle avec la meilleure précision de validation
- ReduceLRonPlateau : réduction du taux d'apprentissage en cas de stagnation

2.5.5 Évaluation du modèle

2.5.5.1 Séparation des données

Le jeu de données a été divisé comme suit :

- 70 % pour l'entraînement
- 15 % pour la validation

- 15 % pour les tests

La division a été réalisée de manière stratifiée pour maintenir la distribution des classes.

2.5.5.2 Résultats obtenus

TABLE 2.2 – Résultats obtenus

Indicateur	Valeur
Précision (accuracy)	96.3 % (test)
Précision moyenne	95.8 %
Rappel moyen	96.1 %
F1-score moyen	96.0 %

La matrice de confusion montre que les erreurs sont rares mais concernent surtout les maladies visuellement similaires comme Tomato___Leaf_Mold et Tomato___ / Late / _ / blight.

2.5.5.3 Visualisation des résultats

Des courbes d'évolution de la précision et de la perte (train/val) ont été générées. Les courbes montrent une bonne convergence sans surapprentissage apparent grâce au dropout et à l'augmentation.

2.5.6 Modèle de recommandation

Pour garder en bon état ses cultures, l'agriculteur doit être au courant de ce qui se déroule dans son champ. C'est qui requiert un système d'alerte intelligent qui propose à l'agriculteur des recommandations qui doit prendre en considération.

Dans ce cadre, nous avons conçu et entraîné un modèle IA pour les recommandations intelligentes. Ce modèle détermine automatiquement les besoins des cultures à partir des données environnementales (humidité du sol, température de l'air ambiant, humidité relative de l'air, rayonnement solaire ...etc) collectées en temps réel par les capteurs IOT installés sur le terrain.

Grâce à ce modèle, l'agriculteur est capable à prévenir les situations de stress hybride, à éviter les pertes et maladies que les plantes peuvent subir et optimiser la consommation de l'eau et des engrès.

2.5.7 Intégration de modèles dans l'application mobile

Après avoir entraîné et testé les modèles, et afin d'assurer une exécution rapide et autonome, ils doivent être convertis au format TFLite. Ce qui permet de générer des prédictions et des recommandations sans appel à un serveur distant et permet aussi aux agriculteurs d'utiliser l'application hors ligne en l'absence de connexion Internet.

2.6 Conception de l'infrastructure IoT

Les capteurs IoT représentent la colonne vertébrale de tout système d'agriculture intelligente. Ils permettent la collecte en temps réel de données agro-environnementales essentielles qui alimentent les modèles d'IA intégrés.

Dans cette partie, nous décrivons l'architecture mise en place ainsi que le mode de transmission et de traitement des données.

2.6.1 Les capteurs IoT utilisés

L'infrastructure de notre système repose principalement sur une carte ESP32 et intégrant des capteurs environnementaux essentiels au suivi agricole intelligent.

1. Capteur de pH du sol

Le capteur d'humidité du sol utilisé avec Arduino permet d'évaluer le niveau d'humidité du sol en se basant sur la variation de la conductivité électrique : plus le sol est sec, plus sa résistance augmente. Ce principe permet de détecter précisément le besoin en irrigation.

Le module comprend une sortie numérique, dont le seuil de déclenchement peut être ajusté grâce à un potentiomètre intégré, et une sortie analogique qui fournit une mesure continue du niveau d'humidité.

L'ensemble se connecte simplement via deux fils à la platine de détection (GND et signal), tandis que l'alimentation en 5V est fournie directement par la carte Arduino ou ESP32. Ce capteur offre ainsi une solution simple et efficace pour surveiller en temps réel l'état hydrique du sol.



FIGURE 2.1 – Capteur d’humidité du sol

2. Capteur de température et d’humidité (DHT22)

Le DHT22 est utilisé pour mesurer la température ambiante et l’humidité de l’air. Il est apprécié pour sa précision, sa fiabilité et sa faible consommation d’énergie. Il fournit sous forme numérique les mesures, ce qui facilite son intégration avec des microcontrôleurs comme l’ESP32.

Le DHT22 est capable de mesurer :

- ★ la température de 40°C à +80°C avec une précision d’environ $\pm 0.5^\circ\text{C}$,
- ★ et l’humidité relative de 0 % à 100 % avec une précision d’environ $\pm 2-5 \%$.

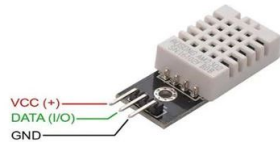


FIGURE 2.2 – Capteur DHT22

3. ESP-WROOM-32

Le module ESP-WROOM-32, représenté à la Figure 2.4, est une solution sans fil multifonction développée par Espressif Systems, combinant les technologies Wi-Fi et Bluetooth dans un seul composant. Il repose sur le microcontrôleur ESP32, doté d’un processeur Xtensa LX6 double cœur, capable d’atteindre une fréquence allant jusqu’à 240 MHz.

Conçu pour répondre aux besoins variés des projets liés à l’Internet des Objets (IoT), ce module prend en charge la connectivité Wi-Fi 2,4 GHz (normes 802.11 b/g/n) ainsi que le Bluetooth 4.2, incluant le mode Low Energy (BLE), ce qui le rend particulièrement adapté aux applications mobiles et autonomes.

L’ESP-WROOM-32 offre une richesse d’interfaces d’entrées/sorties, facilitant la connexion de composants externes :

- ★ GPIOs pour les signaux numériques,

- ★ UART pour les échanges série,
- ★ SPI et I2C pour la communication entre circuits,
- ★ ADC et DAC pour le traitement de signaux analogiques,
- ★ ainsi que le PWM pour le contrôle de moteurs ou d'éclairage.

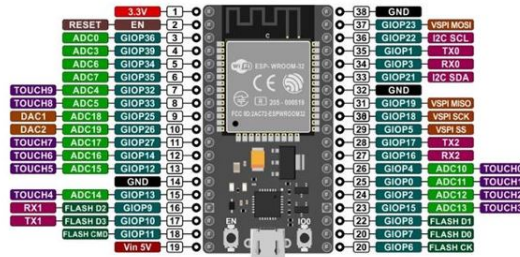


FIGURE 2.3 – Table ESP-WROOM-32

2.7 Conception de l'application mobile

2.7.1 Méthodologie de modélisation adoptée

Dans le cadre de notre projet, nous avons opté le langage UML comme une approche de modélisation. Ci-dessous, nous présentons ce langage puis nous justifions notre choix.

2.7.2 Le langage de modélisation UML

UML (Unified Modeling Language) est un langage de modélisation objet né de la fusion de trois langages de modélisation objet (Booch, OMT, OOSE) qui ont révolutionné la modélisation dans les années 90.

UML est dit universel car il est indépendant des langages de programmation, des domaines d'application et aussi du processus de développement adopté. Il permet de représenter un système sous forme de schémas. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de la solution. Sa véritable force repose sur un méta modèle qui normalise la sémantique des concepts qu'il véhicule, UML est open source.

2.7.3 Pourquoi choisir UML ?

UML est un langage formel et normalisé, il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet pour la garantie de stabilité. Il est un moyen puissant de

communication performant car il cadre l'analyse et facilite la compréhension de représentations, abstraites complexes. Sa diversité et sa flexibilité en font une langue universelle.

2.7.4 Diagrammes UML

Diagramme	Objectifs	Type
Diagramme de cas d'utilisation	Décrire la manière dont une organisation ou un système externe doit interagir avec le système.	Fonctionnel
Diagramme de séquence	<ul style="list-style-type: none"> ★ Décrire ce que doit faire le système. ★ Mettre en évidence les services rendus par le système. ★ Validation des cas d'utilisation pour comprendre la logique de l'application. ★ Complète le diagramme de cas d'utilisation en mettant en évidence les objets et leurs interactions d'un point de vue temporel. 	Dynamique

Diagramme de classe	<ul style="list-style-type: none"> ★ Point central de la modélisation du système pour décrire ce que le système doit faire (analyse) et avec quoi il va le faire (conception). ★ Représentation de la structure statique du système d'information. ★ Modélisation des classes et de leurs relations. 	Statique
---------------------	---	----------

TABLE 2.3 – Diagrammes UML

2.7.5 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de modéliser les interactions entre les utilisateurs (ou "acteurs") et le système. Il offre une vue d'ensemble des fonctionnalités principales du système du point de vue de l'utilisateur final. Dans le cadre de ce projet, ce diagramme illustre les différentes actions que peut effectuer un agriculteur via l'application mobile, telles que la consultation des données environnementales, la réception de recommandations, ou encore l'envoi d'images pour la détection de maladies. Il met également en évidence les interactions internes entre les modules, notamment le traitement des données par les capteurs IoT et le modèle d'intelligence artificielle.

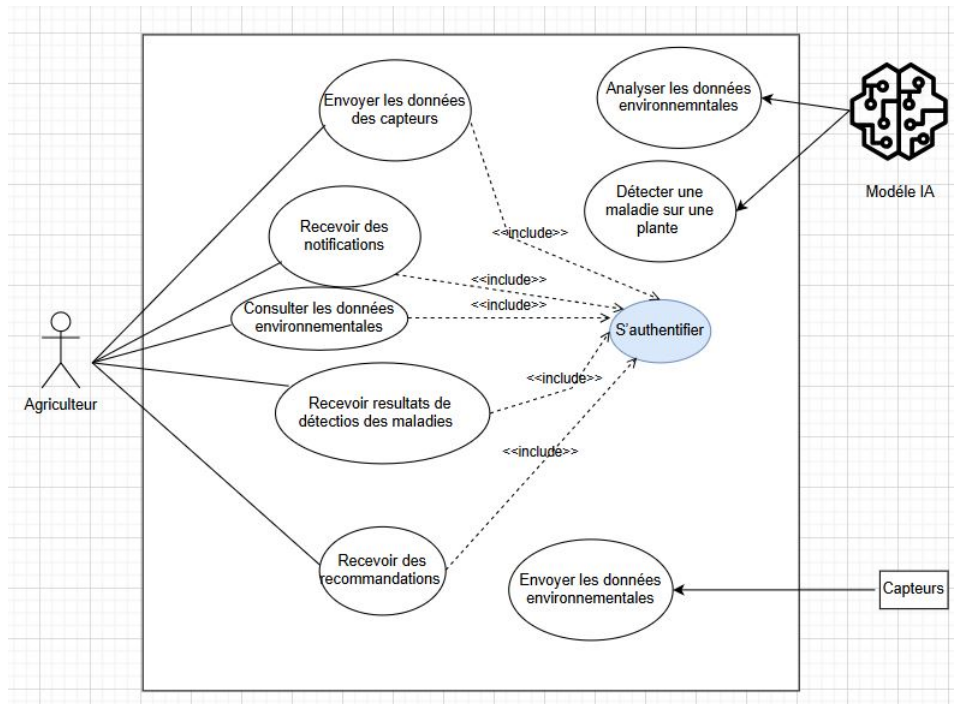


FIGURE 2.4 – Diagramme de cas d'utilisation

2.7.6 Diagramme de classe

Les diagrammes de classes sont sans doute les diagrammes les plus importants représentant les méthodes d'analyse orientées objet. Cela permet de préciser qui intervient à l'intérieur du système.

Un diagramme de classe consiste à représenter la vue statique d'un système d'information. Il s'agit plutôt des relations entre les classes, des services que chaque classe fournit et utilise, et des connexions dans leur ensemble.

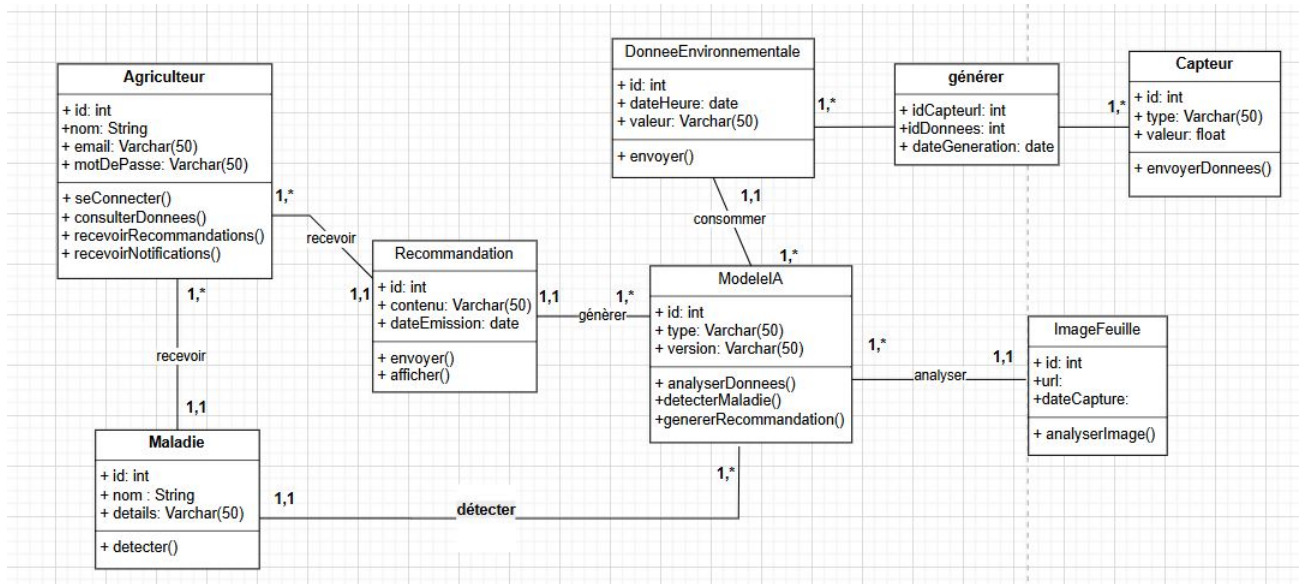


FIGURE 2.5 – Diagramme de classe

2.7.7 Diagrammes de séquence

Les diagrammes de séquence montrent les interactions entre objets selon un point de vue temporel. Ils permettent de mieux visualiser la séquence des messages par une lecture de bas en haut. L'axe vertical représente le temps et l'axe horizontal représente les objets qui collaborent, une verticale en pointillé est attachée à chaque objet qui représente sa ligne de vie.

2.7.8 Diagramme de séquence : Authentification

Lorsque l'utilisateur(agriculteur) souhaite accéder à notre système, il sera obligé de s'authentifier avant d'y accéder en saisissant son email et mot de passe. Après la saisie, le système vérifie les informations envoyées, si les informations sont correctes (compatibles avec la base de données), l'utilisateur accèdera au système, sinon un message d'erreur lui sera affiché et redirigé à la page d'authentification.

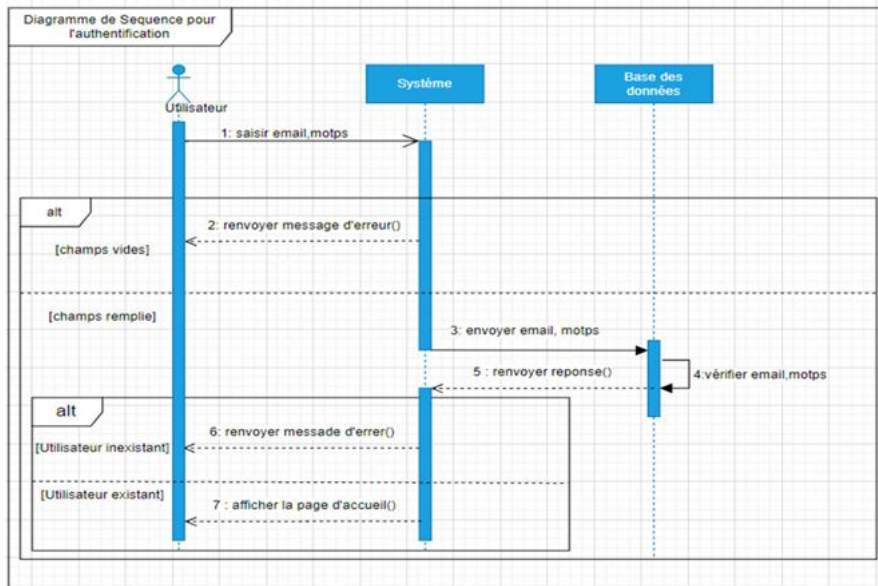


FIGURE 2.6 – Diagramme de séquence : Authentification

2.7.9 Diagramme de séquence :Détection des maladies des plantes

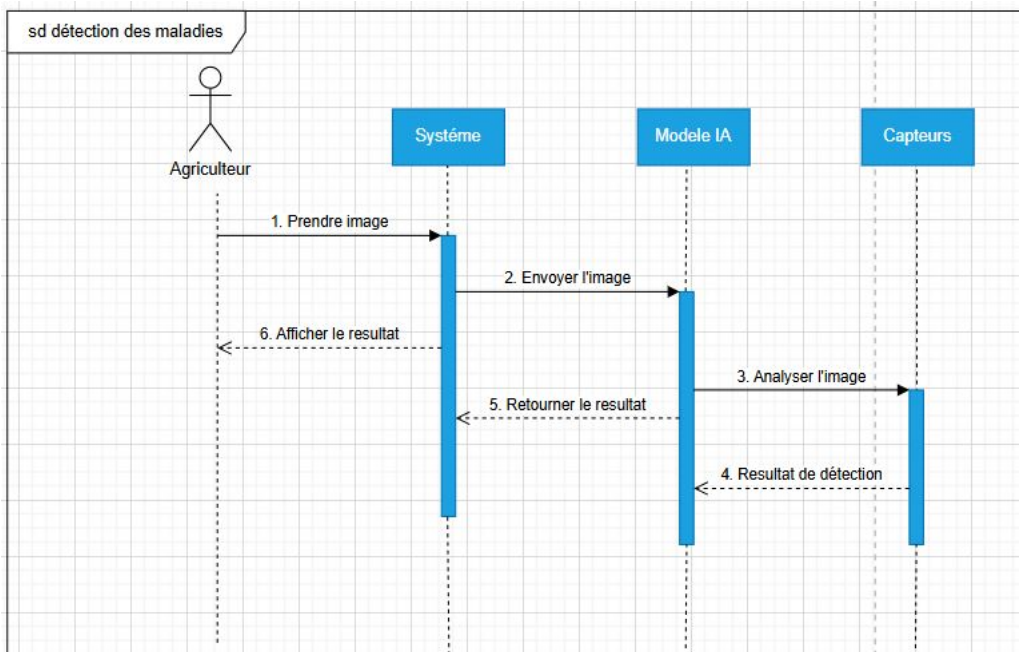


FIGURE 2.7 – Détection des maladies des plantes

2.7.10 Diagramme d'activités

Le diagramme d'activités est un outil de modélisation comportementale permet de représenter le déroulement d'un processus donné sous forme de flux de travail. Il décrit les différentes étapes, décisions et actions réalisées lors de l'exécution d'une activité.

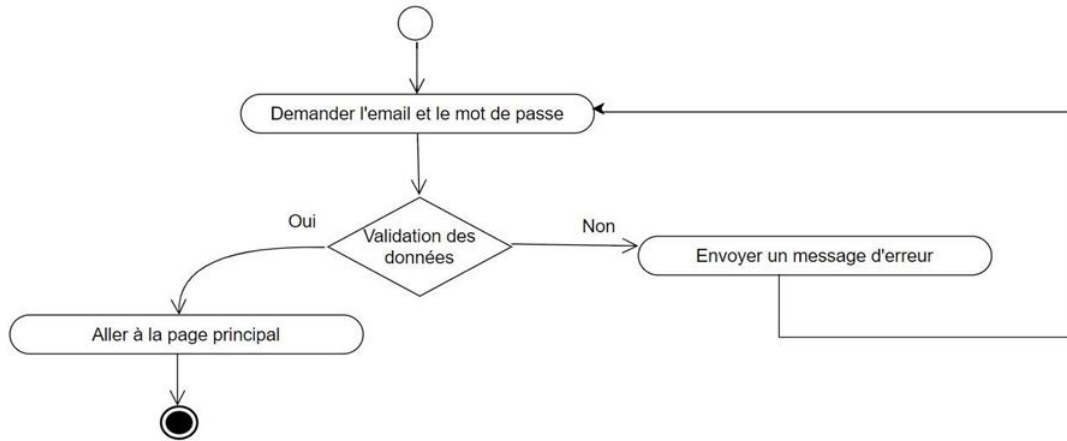


FIGURE 2.8 – Diagramme d'activités pour la connexion

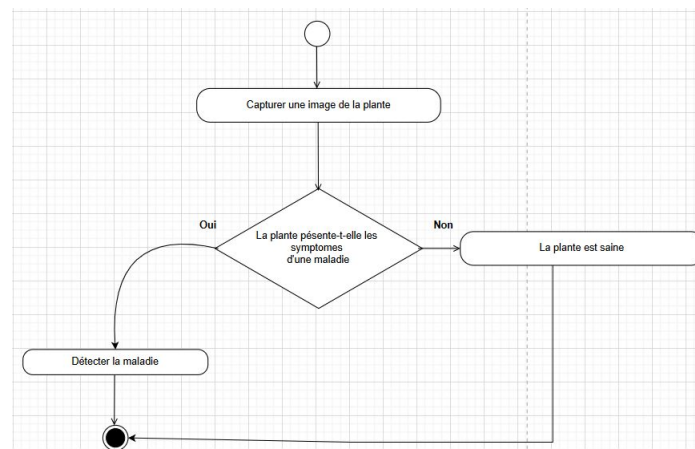


FIGURE 2.9 – Diagramme d'activités pour la détection des maladies des plantes

2.8 Conclusion

Dans ce chapitre, nous avons présenté les exigences attendues, les utilisateurs de l'application et les diagrammes qui modélisent notre système, ce qui va aboutir immédiatement à l'implémentation.

Dans le chapitre suivant, nous présenterons, l'environnement, la technologie et les outils de développement ainsi que l'architecture et les interfaces du système.

CHAPITRE 3

IMPLÉMENTATION

3.1 Introduction

Après avoir finalisé l'étape d'analyse et conception, nous passons à l'implémentation de notre application . Nous commençons tous d'abord par la présentation de l'architecture de l'application, l'environnement matériel et logiciel ainsi que les différents outils utilisés pour l'implémentation. Puis nous présentons quelques interfaces de notre application.

3.2 Environnement de travail

3.2.1 Environnement Matériel

Nous avons élaboré ce travail sur deux PC dont la configuration est la suivante :

- Pc portable : Dell – RAM : 16,00 Go
- Disque dur : 1000 GO
- Microprocesseur : Intel(R) Core (TM) i7-5005U CPU @ 2.00GHz 2.00 GHz
- Pc portable : acer – RAM : 16,00 Go
- Disque dur : 1000 GO
- Microprocesseur : Intel(R) Core (TM) i7-5005U CPU @ 2.00GHz 2.00 GHz

3.2.2 Environnement Logiciel

- Kaggle



FIGURE 3.1 – Logo Kaggle

Kaggle est une plateforme en ligne dédiée à la science des données et à l'apprentissage automatique. Elle permet aux utilisateurs de participer à des compétitions, d'explorer des ensembles de données variés, de publier des notebooks interactifs et de collaborer avec une vaste communauté de data scientists. Utilisée aussi bien par les débutants que par les professionnels, Kaggle constitue un environnement riche pour expérimenter, tester des modèles et accéder à des re

— Google Colab

**FIGURE 3.2** – Logo Google Colab

Est un environnement de développement développé par Google, permettant d'écrire et d'exécuter du code Python directement dans un navigateur. Google colab est basé sur les notebooks Jupyter, en offrant l'accès gratuit à des ressources matérielles puissantes, à titre d'exemple les GPU et TPU, ce qui le rend adapté aux projets de machine learning, de deep learning et d'analyse de données.

— Flutter

**FIGURE 3.3** – Logo Flutter

Flutter est un framework open source développé par Google qui permet de créer des applications multiplateformes (Android, iOS, Web, Windows, macOS, Linux) à partir d'un seul code source écrit en Dart. Il se distingue par son moteur de rendu performant, sa capacité à créer des interfaces utilisateur modernes, fluides et personnalisées, et son approche déclarative basée sur des widgets.s apportées au code.

— Fritzing

**FIGURE 3.4** – Logo Fritzing

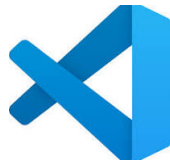
Fritzing est un logiciel open source de conception électronique sert à faciliter la création, la documentation et le prototypage de circuits électroniques. Il permet de représenter graphiquement les montages réalisés sur une plaque d'essai (breadboard), et de concevoir des circuits imprimés (PCB). Fritzing est fortement utilisé dans le prototypage rapide et des projets DIY liés à l'IoT.

— Arduino IDE

**FIGURE 3.5** – Logo Arduino IDE

L'Arduino IDE est une plateforme électronique qui propose des solutions logicielles et matérielles conviviales, adaptées aussi bien aux débutants qu'aux experts. Arduino prend en charge l'ensemble des cartes Arduino ainsi que d'autres extensions comme l'ESP32, LoRa, et bien d'autres. De plus, son interface se distingue par un design simple, intégrant un éditeur de code, la coloration syntaxique, une barre d'outils, un menu de fonctions avancées, ainsi qu'une console fournissant un retour en temps réel.

— Visual Studio Code (VS Code)

**FIGURE 3.6** – Logo Vs Code IDE

Visual Studio Code est un éditeur de code multiplateforme développé par Microsoft. Apprécié pour sa rapidité, sa flexibilité et sa richesse fonctionnelle, il prend en charge une large gamme de langages de programmation grâce à une vaste bibliothèque d'extensions. Doté d'un terminal intégré, d'une autocomplétion intelligente (IntelliSense), d'un débogueur performant et d'un système de gestion de projet efficace..

— Android Studio IDE

**FIGURE 3.7** – Logo Android Studio IDE

Un éditeur de code et environnement développement pour créer des application Android. Il est basé sur IntelliJ IDEA, et permet de concevoir, tester et déployer des applications mobiles.

— Figma

**FIGURE 3.8** – Logo Figma

Figma est un outil de conception d'interface UI et UX basé sur le cloud. Il permet aux designers de créer des designs innovants interactifs en ligne depuis un navigateur. Figma est fortement utilisé en vue de concevoir des prototypes interactifs grâce à son interface intuitive et il est ainsi connu par sa flexibilité à faciliter la collaboration en temps réel entre plusieurs utilisateurs car il est idéal pour les équipes de design.

— GitHub

**FIGURE 3.9** – Logo GitHub

GitHub est une plateforme en ligne de gestion de projets et de collaboration, basée sur le système de contrôle de version Git. Elle permet aux développeurs de stocker, suivre, et gérer leurs projets de manière efficace, tout en facilitant le travail en équipe grâce à des fonctionnalités telles que les dépôts (repositories), les branches

— SDK

**FIGURE 3.10** – Logo SDK

SDK, également appelé trousse de développement Il s'agit d'un ensemble d'outils conçus afin d'aider les développeurs dans la création de leurs projets sur une plateforme spécifique. Il regroupe différents composants tels que des bibliothèques, ainsi que des utilitaires facilitant le processus de programmation.

— UML

**FIGURE 3.11** – Logo UML

UML est un langage de modélisation standardisé utilisé pour représenter visuellement la structure, le comportement et l'architecture d'un système. Il permet de documenter, concevoir et comprendre plus facilement les différentes composantes d'une application à travers des diagrammes comme les cas d'utilisation, les classes, les séquences ou encore les activités.

— OpenWeather

**FIGURE 3.12** – Logo OpenWeather

OpenWeather (anciennement OpenWeatherMap) est une plateforme de services météo en ligne qui fournit des données météorologiques en temps réel, prévisions à long terme, et d'autres informations climatiques à l'échelle mondiale.

— Tomorrow.io

**FIGURE 3.13** – Logo Tomorrow.io

Est une plateforme météorologique avancée fournissant des prévisions hyperlocales et personnalisées grâce à l'intelligence artificielle et aux données satellitaires. Elle est conçue pour les entreprises et développeurs qui ont besoin de données météo précises, en temps réel et orientées vers l'action.

3.3 Implémentation

Dans cette section, notre application est implémentée et ses interfaces sont organisées d'une manière claire et bien structurée.

3.3.1 Inscription et Authentification

Pour accéder à l'application, l'utilisateur passe par l'interface d'authentification s'il est doté d'un compte, sinon il lui nécessite de créer un compte l'interface d'inscription. Ces interfaces sont représentées dans Figure 3.14 et Figure 3.15

HasadAPP

Entrez votre email

Entrez votre mot de passe

se connecter

Mot de passe oublié

ou

Continuer avec Google

Vous avez déjà un compte? [s'inscrire](#)

FIGURE 3.14 – Interface d'authentification

HasadAPP

Entrez votre nom

Entrez votre email

Entrez votre mot de passe

Confirmer le mot de passe

s'inscrire

Vous avez déjà un compte? [se connecter](#)

FIGURE 3.15 – Interface d'inscription

3.3.2 Page d'accueil

Après authentifié et la sauvegarde de ses données dans le back-end firebse, l'utilisateur, accède à la page d'accueil ou principale de l'application(FIGURE 3.16) qui contient les cartes personnalisées de support intelligent (dont parlerons de chacune plus tard).

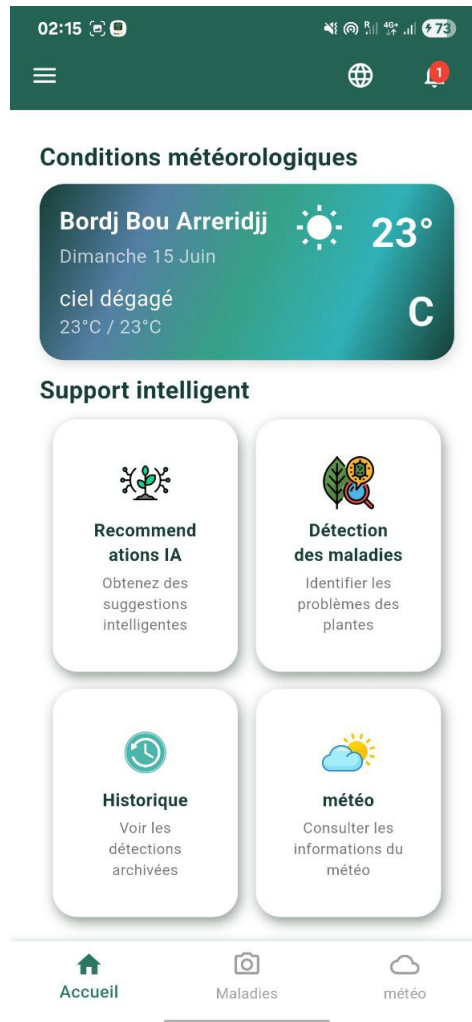


FIGURE 3.16 – Page d'accueil

3.3.3 Détection des maladies

En cliquant sur la carte personnalisée "Détection des maladies", l'agriculteur peut identifier des manière précise et très efficaces les problèmes des plantes à partir des sytomes qu'ils présentent grace à un modèle IA bien entrainé.

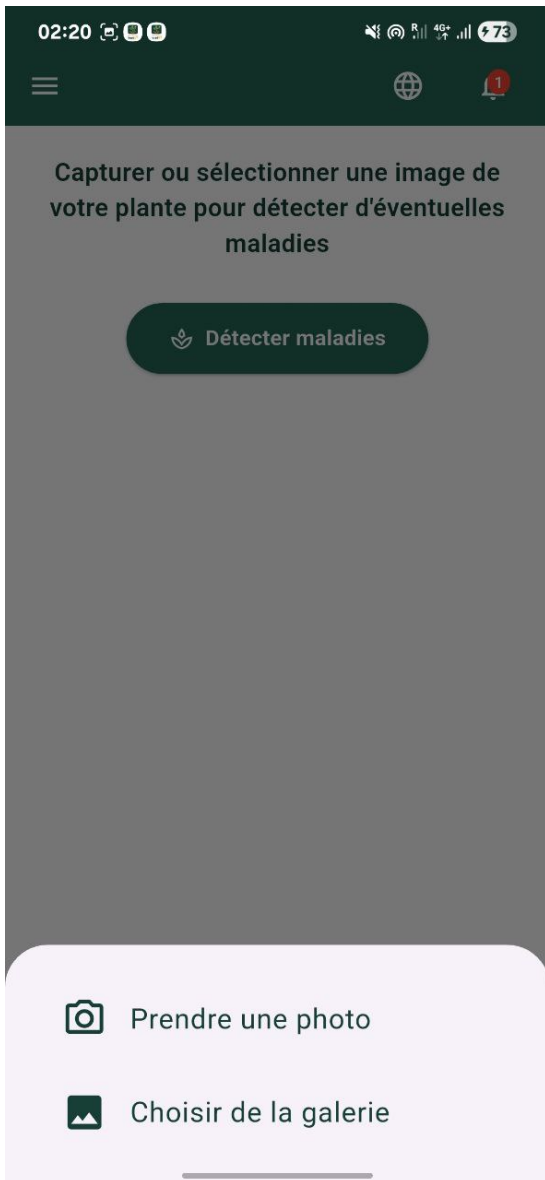


FIGURE 3.17 – Examiner les plantes

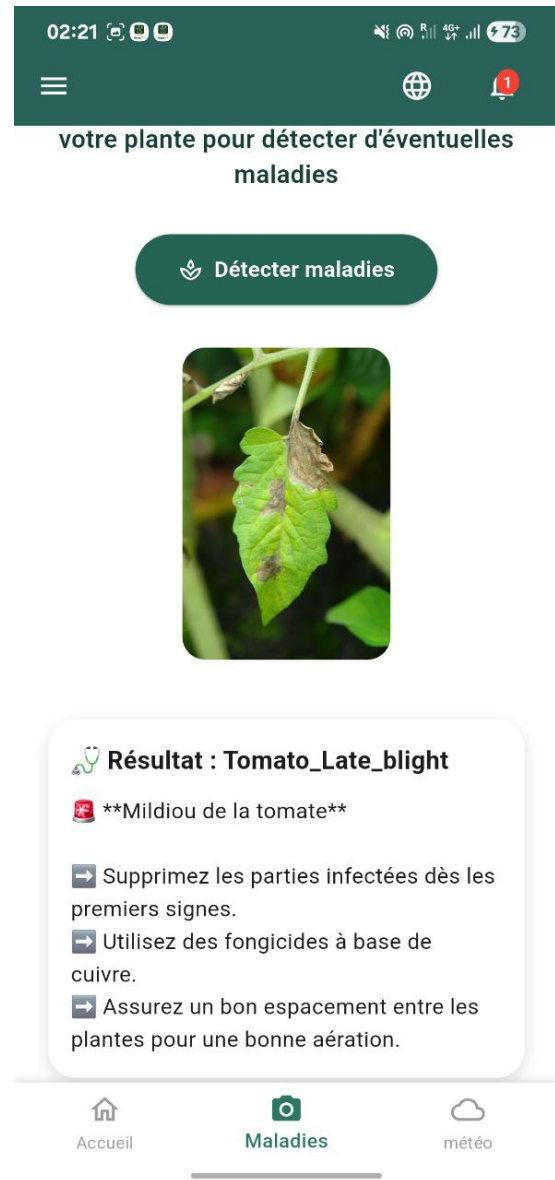


FIGURE 3.18 – Resultat de détection

3.3.4 Météo et Recommendations IA

Les données météorologique sont collectées en temps réel via les capteurs IoT (FIGURE 3.19), puis transmises à un modèle IA conçu spécifiquement pour en analyser et suggère de manière prise des recommandations intelligentes(FIGURE 3.20).



FIGURE 3.19 – Examiner les plantes

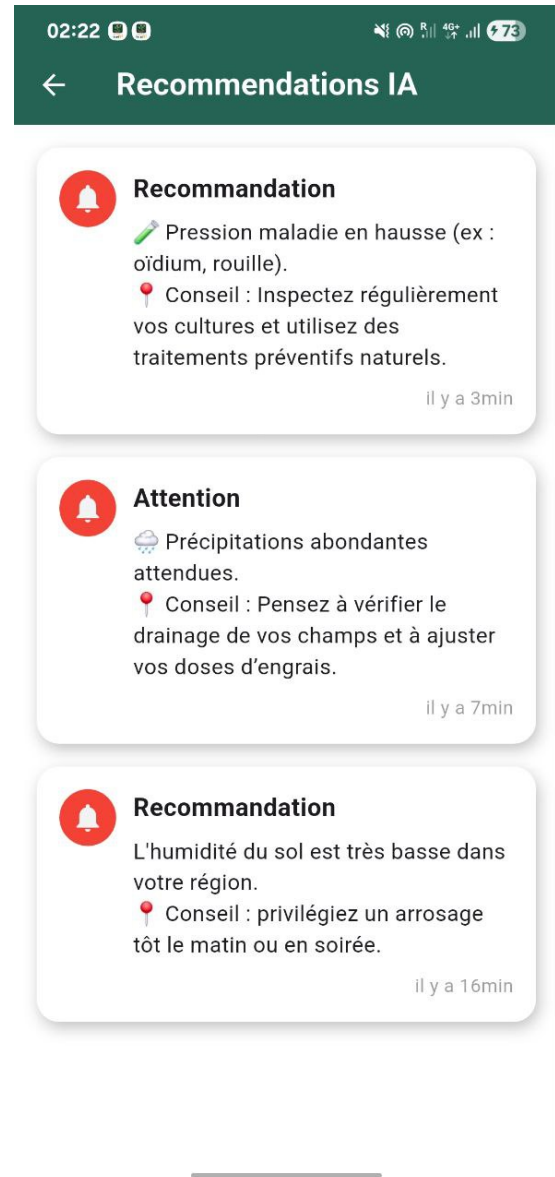


FIGURE 3.20 – Resultat de détection

3.4 Conclusion

Dans ce chapitre, un système intelligent et complet a été mis en œuvre pour la surveillance agro-météorologique. Le système intègre plusieurs fonctionnalités, notamment la mesure locale de paramètres environnementaux (température, humidité, qualité de l'air, etc.) grâce aux capteurs, l'utilisation de données météorologiques externes issues d'APIs comme OpenWeather et Tomorrow.io, ainsi que le contrôle automatique de certains équipements agricoles. Deux modèles d'intelligence artificielle ont également été intégrés : l'un pour la détection de maladies des plantes à partir d'images, et l'autre pour la génération de recommandations agricoles intelligentes basées sur les prévisions météorologiques. Les résultats expérimentaux ont démontré la fiabilité du système à transmettre des données sur de longues distances et à fournir une assistance intelligente et contextuelle aux agriculteurs. Cette étape conclut ainsi la phase de conception, de développement et de mise en œuvre du système, affirmant son efficacité et sa pertinence dans le cadre d'une agriculture moderne, connectée et durable

CONCLUSION GÉNÉRALE

Ce travail s'inscrit dans le cadre du développement de systèmes intelligents appliqués au domaine de l'agriculture, en mettant en œuvre une solution complète de surveillance agrométéorologique intégrant les technologies de l'Internet des objets (IoT), la communication sans fil (WIFI), l'intelligence artificielle (IA) et des services cloud.

Nous avons conçu et réalisé une architecture distribuée capable de collecter des données environnementales (température, humidité, qualité de l'air, etc.) à partir de capteurs IoT, de les transmettre puis de les synchroniser en temps réel dans une base de données via Firebase. En complément, des API météo telles qu'OpenWeather et Tomorrow.io ont été intégrées pour enrichir les analyses avec des prévisions météorologiques précises.

Deux modèles d'intelligence artificielle ont été développés : le premier permet la détection automatique des maladies des plantes à partir d'images capturées, tandis que le second analyse les données météo pour générer des recommandations agricoles intelligentes, personnalisées et contextualisées. Une application mobile multiplateforme (Android/iOS) a également été développée pour permettre à l'utilisateur de visualiser l'ensemble des données et recommandations de manière intuitive.

Les résultats expérimentaux ont démontré la fiabilité et la portée du système dans des conditions réelles, confirmant la pertinence de l'approche adoptée pour moderniser les pratiques agricoles. Ce projet ouvre la voie à une agriculture plus connectée, proactive et durable, où les décisions sont prises sur la base de données précises, en temps réel, et assistées par l'IA.

Plusieurs perspectives sont envisageables pour les travaux futurs, telles que l'intégration de nouvelles fonctionnalités basées sur la vision par ordinateur, l'analyse prédictive à long terme, ou encore l'extension du système à d'autres types de cultures ou environnements agricoles.

BIBLIOGRAPHIE

- [1] S. G. H. Soumyalatha, “Study of IoT : understanding IoT architecture, applications, issues and challenges,” in *1st Int. Conf. on Innovations in Computing & Networking (ICICN16)*, Int. J. of Advanced Networking & Applications, 2016.
- [2] K. K. Patel, S. M. Patel, and P. Scholar, “Internet of Things-IOT : Definition, characteristics, architecture, enabling technologies, application & future challenges,” *Int. J. Eng. Sci. Comput.*, vol. 6, no. 5, 2016.
- [3] S. Devalal and A. Karthikeyan, “LoRa technology—An overview,” in *Proc. 2nd Int. Conf. on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 2018, pp. 284–290.
- [4] A. Diané, “Modèle de déploiement optimal des nœuds pour un réseau LoRa multilinéaire de bout en bout sans fil,” Master Thesis, 2022.
- [5] I. K. A. Enriko, F. N. Gustiyana, K. Kurnianingsih, and E. L. I. P. Sari, “LoRaWAN for Smart Street Lighting Solution in Pangandaran Regency,” *JOIV Int. J. Inform. Vis.*, vol. 7, no. 4, pp. 2452–2459, 2023.
- [6] A. Zourmand, A. L. K. Hing, C. W. Hung, and M. AbdulRehman, “Internet of Things (IoT) using LoRa technology,” in *IEEE Int. Conf. on Automatic Control and Intelligent Systems (I2CACIS)*, 2019, pp. 324–330.
- [7] D. M. Ibrahim, “Internet of Things technology based on LoRaWAN revolution,” in *10th Int. Conf. on Information and Communication Systems (ICICS)*, IEEE, 2019, pp. 234–237.
- [8] N. S. Chilamkurthy, O. J. Pandey, A. Ghosh, L. R. Cenkeramaddi, and H.-N. Dai, “Low-power wide-area networks : A broad overview of its different aspects,” *IEEE Access*, vol. 10, pp. 81926–81959, 2022.
- [9] J. Č. Gambiroža, T. Mastelić, P. Šolić, and M. Čagalj, “Capacity in LoRaWAN networks : Challenges and opportunities,” in *4th Int. Conf. on Smart and Sustainable Technologies (SpliTech)*, IEEE, 2019, pp. 1–6.

-
- [10] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Šolić, and A. L. Aquino, “LoRa-WAN—A low power WAN protocol for Internet of Things : A review and opportunities,” in *2nd Int. Multidisciplinary Conf. on Computer and Energy Science (SpliTech)*, IEEE, 2017, pp. 1–6.
- [11] S. Sendra Compte, L. García, J. Lloret, and R. Vega Rodríguez, “LoRaWAN Network for Fire Monitoring in Rural Environments,” 2020.
- [12] M. Hammache, “Performances et gestion de l’itinérance dans les réseaux LoRaWAN,” PhD Thesis, Institut National Polytechnique de Toulouse-INPT, 2022.
- [13] Heltec Automation, “HTCC-AB01 LoRa Development Board.” [Online]. Available : <https://heltec.org/project/htcc-ab01>
- [14] S. K. Natarajan, P. Shanmurthy, D. Arockiam, B. Balusamy, and S. Selvarajan, “Optimized machine learning model for air quality index prediction in major cities in India,” *Scientific Reports*, vol. 14, no. 1, p. 6795, 2024.
- [15] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
- [16] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [17] C. E. Pereira and A. D. Silva, “Using Firebase as a Backend Service for IoT Applications,” *International Journal of Computer Applications*, vol. 975, p. 8887, 2020.
- [18] H. Zhang and Y. Zhao, “Enhancing Agricultural Forecasting with Weather APIs : An IoT-based Framework,” *Sensors*, vol. 22, no. 4, p. 1492, 2022.
- [19] P. P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed, and A. Zaslavsky, “Internet of Things platform for smart farming : Experiences and lessons learnt,” *Sensors*, vol. 16, no. 11, p. 1884, 2016.