Université de Tek-Up

DatteAI Valor

Auteur :

Aymen Tangara

Mehrez Orabi

Saif Bahri

Eya saied

Nour el houda Ben ghayadha

Superviseur :

Mm. Sawssen Jalel

Mm. Zeineb SADRAOUI

20 décembre 2024

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Dans le cadre de ce projet, nous avons cherché à répondre à un besoin crucial dans le secteur de l'agriculture, plus spécifiquement dans la culture des palmiers dattiers, qui constitue une richesse essentielle pour la Tunisie, notamment dans les oasis du sud telles que Tozeur, Kebili, Gafsa et Douz. En dépit de cette richesse, la détection précoce des maladies des palmiers reste un défi majeur pour les agriculteurs. L'intégration de la technologie dans ce domaine, à travers l'utilisation du machine learning et des outils modernes de développement logiciel, représente une opportunité pour améliorer la gestion de ces cultures.

Le projet a pour objectif de concevoir et de développer une application mobile dédiée à la détection des maladies des palmiers, en utilisant des techniques avancées de traitement d'image et d'intelligence artificielle. L'application permet aux utilisateurs, principalement les agriculteurs et les experts en palmiers, de détecter rapidement les maladies spécifiques affectant leurs cultures et de recevoir des recommandations de traitement adaptées. Cette solution repose sur une architecture robuste et moderne, composée d'un front-end développé en Flutter, d'un back-end utilisant NestJS, d'une base de données MongoDB, ainsi que de l'intégration du machine learning pour l'analyse des images.

Le processus de développement de ce projet a impliqué plusieurs étapes clés : un brainstorming initial pour définir les besoins et les objectifs du projet, suivi d'une étude approfondie du terrain et de contacts avec les acteurs locaux afin de recueillir des informations pertinentes. Ce processus a également permis de prendre des décisions techniques sur les outils à utiliser pour garantir la meilleure qualité du projet. L'application

a été développée avec une attention particulière portée à la qualité à chaque étape, en intégrant des outils de test et d'intégration continue tels que GitHub et Kubernetes pour le déploiement DevOps, ainsi que des réunions quotidiennes via Google Meet pour assurer une communication fluide entre les membres de l'équipe.

En termes de tests, une série de vérifications fonctionnelles et non fonctionnelles ont été mises en place, notamment des tests de performance, de sécurité et de compatibilité multiplateforme. L'utilisation d'outils modernes de test et la mise en œuvre d'un panel d'administration développé en NestJS et Next.js permettent une gestion optimale de l'application et de ses utilisateurs.

Ce rapport présente l'ensemble des processus suivis, des choix techniques opérés, ainsi que des résultats obtenus lors des différentes phases de test et de validation. L'objectif étant de garantir la performance, la sécurité et l'expérience utilisateur de cette application mobile innovante dans le domaine de l'agriculture et de la santé des palmiers dattiers.

Chapitre 1

Étude préliminaire

Introduction

Ce chapitre a pour objectif de situer le projet dans son contexte général. Nous y présentons l'organisme d'accueil, puis nous exposons la problématique liée à notre projet. Enfin, nous détaillons la méthodologie adoptée pour mener à bien ce travail.

1.1 Contexte et définition du projet

1.1.1 Contexte du projet

Le secteur des dattes en Tunisie, particulièrement dans les régions des oasis, est un pilier majeur de l'économie agricole. Les dattes sont perçues par les habitants du sud tunisien comme la principale, voire l'unique, source de revenus pour de nombreuses familles. En plus de la vente des fruits, les palmiers servent à divers usages, renforçant l'importance vitale de cette ressource. Cependant, les producteurs locaux sont souvent confrontés à des difficultés liées à la méconnaissance de la qualité et de la valeur marchande de leurs produits. Ils vendent souvent leurs récoltes à des prix sous-évalués, entraînant des pertes financières. Le manque de transparence et de données favorise également des pratiques où les investisseurs profitent de cette situation en achetant à bas prix. Ce projet a pour objectif de résoudre ces problématiques en utilisant des technologies modernes telles que l'intelligence artificielle et le machine learning pour classifier les dattes en fonction de leurs caractéristiques et anomalies. L'ambition est d'offrir un outil de suivi intelligent aux producteurs, leur permettant ainsi de mieux valoriser leurs produits et d'assurer un revenu plus juste et stable pour les familles qui dépendent de cette ressource.

1.2 Problématique

Les principaux problèmes rencontrés dans le secteur des dattes en Tunisie sont :

— La méconnaissance des producteurs quant à la qualité et à la valeur réelle de leurs dattes.

— Les anomalies non détectées affectant la qualité des dattes et des palmiers, entraînant des pertes non anticipées.

— Un manque de transparence sur les prix et la qualité, profitant davantage aux investisseurs qu'aux fermiers.

— L'absence d'un système de suivi intelligent pour aider les fermiers à mieux gérer leurs oasis et à anticiper les problèmes liés à la culture des palmiers.

1.3 Étude Préliminaire

Le secteur des dattes en Tunisie, particulièrement dans le sud, est crucial pour l'économie locale. Cependant, les producteurs font face à des difficultés liées à la méconnaissance de la qualité de leurs produits, la détection tardive des anomalies et un manque de transparence sur les prix, ce qui entraîne des pertes financières.

1.3.1 Problématiques rencontrées

Les principaux problèmes identifiés dans le secteur sont :

— **Méconnaissance de la qualité des dattes** : Les producteurs ne savent pas évaluer correctement leurs récoltes.

— **Anomalies et maladies non détectées** : Les problèmes des palmiers et des dattes ne sont pas identifiés à temps.

— **Manque de transparence des prix** : Les producteurs vendent souvent à des prix trop bas.

— **Absence d'outils de suivi** : Aucun système pour aider à la gestion des cultures.

1.3.2 Besoins identifiés

Les entretiens avec Madame Monjia Charef (responsable du contrôle et de la qualité) et Monsieur Rabii Ezzine (responsable agricole) ont révélé plusieurs besoins :

— **Outil de classification de la qualité des dattes** : Estimer la valeur des récoltes.

— **Détection des anomalies et des maladies** : Identifier les problèmes rapidement.

— **Recommandations de traitements** : Proposer des solutions pour améliorer la qualité.

— **Suivi des cultures** : Enregistrer et suivre les traitements appliqués.

L'étude préliminaire confirme les besoins des producteurs en matière de suivi intelligent et d'amélioration de la gestion des cultures. L'application proposée pourrait répondre efficacement à ces besoins, en utilisant des technologies comme l'intelligence artificielle.

1.4 Objectif du Projet

L'objectif principal de ce projet est de développer une solution technologique basée sur l'intelligence artificielle (IA) et le machine learning pour améliorer la gestion et la valorisation des dattes et des palmiers en Tunisie, en particulier dans les régions du sud. Plus précisément, ce projet vise à :

— Classifier les dattes selon leur qualité en détectant les anomalies liées aux fruits et aux palmiers, afin de permettre aux producteurs de mieux comprendre et gérer leurs récoltes.

— Fournir aux fermiers un outil de suivi intelligent qui leur permet d'anticiper les problèmes et d'optimiser la production, tout en conservant un historique des anomalies détectées et des traitements appliqués.

— Assurer une meilleure valorisation des dattes en fournissant des estimations précises

de leur valeur marchande, afin d'éviter les ventes à perte et d'instaurer plus de transparence dans la chaîne de vente, profitant ainsi aux producteurs plutôt qu'aux investisseurs.

— Proposer des recommandations personnalisées pour le traitement des anomalies et l'optimisation des cultures de palmiers, améliorant ainsi la durabilité et la productivité des oasis.

— Maximiser l'utilisation des palmiers en permettant aux fermiers d'exploiter toutes les potentialités de cette ressource, véritable trésor pour les habitants du sud tunisien, et ce, de façon rentable et durable.

1.5 Étude de l'existant

L'étude de l'existant est une phase primordiale lors de l'élaboration du projet. Elle permet de comprendre et analyser le système de travail actuel. La gestion manuelle présente plusieurs problèmes.

1.5.1 Analyse de l'existant

Tout au long de l'étude préalable du projet, nous avons détecté certaines défaillances non résolues par la méthode de travail actuelle, notamment :

— Absence d'un système d'intelligence artificielle et de détection des anomalies sur les dattes et les oasis spécifiquement.

1.5.2 Exemples d'applications

Quelques exemples d'applications existantes sont :

— **AgriTech Tunisia** : AgriTech Tunisia est une plateforme locale qui propose des solutions numériques pour l'agriculture en Tunisie, incluant des outils de gestion des récoltes, la détection des maladies, et des conseils personnalisés pour optimiser

la production agricole. Elle utilise des technologies comme l'Internet des objets (IoT) et l'analyse des données agricoles pour aider les agriculteurs à mieux gérer leurs fermes.

— **Ag Leader** : Ag Leader propose des solutions d'agriculture de précision, y compris des systèmes pour l'analyse des sols, la gestion des cultures, et l'optimisation de l'irrigation. Leur technologie permet de suivre l'état des cultures et de fournir des recommandations personnalisées sur les traitements à appliquer.

— **FarmLogs** : FarmLogs est une application de gestion agricole qui aide les fermiers à suivre leurs cultures, prévoir les rendements, gérer les finances, et optimiser les ressources. L'application permet aussi d'enregistrer l'historique des interventions, des anomalies et de suivre les performances des cultures.

1.5.3 Présentation de la solution

Le projet consiste à développer une application intelligente pour classer les palmiers selon leur état et identifier les anomalies présentes dans les palmiers. Elle offrira des recommandations adaptées pour traiter ces anomalies et réalisera une analyse descriptive des données historiques et des anomalies observées. L'application permettra également un suivi intelligent des cultures de palmiers, avec un enregistrement automatique des données et un historique détaillé des anomalies. Elle proposera des estimations de la valeur des palmiers en fonction de leur état et des traitements appliqués, afin d'aider à optimiser la gestion des ressources. En outre, elle servira d'outil de transparence pour les investisseurs, leur permettant d'évaluer l'état des palmiers avant l'achat. Cette solution pourra être étendue à d'autres cultures agricoles, tout en intégrant des technologies comme la réalité augmentée et virtuelle pour enrichir l'expérience utilisateur.

1.5.4 Tableau comparatif

Critères	AgriTech Tunisia	Ag Leader	FarmLogs	DatteAI Valor
Type de technologie utilisée	IoT, analyse des données agricoles	Agriculture de précision (analyse des sols, gestion des cultures)	Gestion des cultures, prévision des rendements	Machine Learning pour la détection des maladies des palmiers, suivi intelligent avec analyse d'images
Objectif principal	Gestion des récoltes et des maladies	Optimisation de l'irrigation et gestion des cultures	Suivi des rendements et gestion des finances	Détection des anomalies sur les palmiers, recommandations de traitement adaptées
Précision de la détection des anomalies	Détection des maladies agricoles de manière générique	Analyse des sols et des cultures	Suivi des performances et anomalies agricoles	Détection précise des anomalies sur les palmiers, basée sur des images scannées et l'intelligence artificielle

Critères	**AgriTech Tunisia**	**Ag Leader**	**FarmLogs**	**DatteAI Valor**
Utilisation de la technologie de pointe	Pas d'intelligence artificielle	Pas d'intelligence artificielle	Pas d'intelligence artificielle	Utilisation de l'intelligence artificielle pour analyser des images et détecter des anomalies spécifiques aux palmiers
Valeur ajoutée	Solution locale avec outils de gestion	Spécialisation dans l'agriculture de précision	Gestion globale de l'agriculture	Application dédiée à la détection des maladies des palmiers, utilisation du machine learning, interface conviviale, et possibilité d'intégrer des technologies futures comme la réalité augmentée

Tableau 1.1 : Tableau comparatif des applications

1.6 Méthodologie adoptée

Pour différents types de projets, l'utilisation d'une méthodologie agile est l'option la plus populaire. La base de la méthodologie agile est un cycle de développement qui met l'accent sur la communication avec le client. L'implication du client dans le projet lors de sa réalisation permet d'avoir une vision plus claire et de valoriser le travail au fur et à mesure de sa réalisation.

1.6.1 Présentation du cadre du projet

En raison de son applicabilité à nos besoins et de notre volonté de satisfaire les demandes des clients tout en maximisant les chances de réussite du projet, nous avons décidé d'utiliser la méthode Scrum comme méthodologie de travail. Nous allons également utiliser le framework Jira.

1.6.2 Présentation de Scrum

Ce modèle est le plus adapté aux sociétés ; il repose sur des réunions appelées « Daily Scrum Meeting (DSM) » qui durent 15 minutes chaque jour, dans le but d'identifier les problèmes rencontrés et de revoir ce qui s'est passé hier et ce qui reste à faire. Le cycle de vie Scrum est rythmé par des itérations de deux à quatre semaines. À chaque début du sprint, des réunions appelées « Sprint Planning » sont organisées. En effet, le sprint planning consiste à préciser une liste des tâches pour le prochain sprint, cette liste représentant le Product Backlog. À la fin de chaque sprint, toute l'équipe se réunit et prend en général 15 à 30 minutes pour discuter des réussites et des points à améliorer, c'est la rétrospective.

Les acteurs participants dans le cycle SCRUM sont :

— **Le propriétaire du produit (Product Owner)** : Il représente le client et est

en contact avec le Scrum master.

— **Le Scrum master** : Il est responsable de l'application quotidienne des directives Scrum et de la création d'un environnement de travail agréable pour l'équipe.

— **L'équipe de travail** : Se compose de deux à neuf personnes. Chaque membre doit identifier, pendant le « Sprint Planning », les tâches qu'il a achevées et celles qui restent à faire, dans un « Product Backlog » spécifique à chaque sprint.

1.6.3 Les rôles Scrum

Dans le cadre de notre projet, nous présentons l'équipe de travail selon les rôles Scrum dans le tableau suivant :

Rôle	**Description**
Product Owner	Aymen TANGARA
Scrum master	mEHREZ OURABI
Development Team	Eya SAIED / Saif BAHRI / Nour elhouda BEN GHAYADHA

Tableau 1.2 : L'équipe de travail selon le Framework SCRUM

Conclusion

En conclusion, ce premier chapitre a établi le contexte du projet, identifié la problématique, proposé des solutions novatrices, et défini la méthodologie Scrum. Ces éléments constituent une base solide pour la suite du rapport, orientant notre analyse et la mise en œuvre du projet.

Chapitre 2

Analyse des besoins

Introduction

Nous consacrons ce deuxième chapitre à la phase de spécification et d'analyse des besoins. C'est la première étape de la conception qui consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent, afin d'assurer un ouvrage ou un processus répondant aux besoins.

2.1 Analyse des besoins

La spécification des besoins est nécessaire pour le développement d'un produit. La phase de spécification est une traduction des besoins utilisateurs en documentations conceptuelles et techniques. Elle contient l'analyse des besoins fonctionnels et des besoins non fonctionnels.

2.1.1 Besoins fonctionnels

— **Classification automatique des palmiers** : Utilisation de l'IA pour classer les palmiers en différentes catégories selon leur état et leurs caractéristiques.

— **Détection d'anomalies** : Détection des anomalies affectant les palmiers (problèmes liés à la santé des palmiers, maladies, irrégularités de croissance, etc.).

— **Recommandations de traitement** : Propositions de solutions spécifiques pour remédier aux anomalies détectées sur les palmiers.

— **Suivi intelligent** : Historique détaillé des anomalies, traitements appliqués et résultats obtenus pour chaque palmier.

— **Analyse descriptive** : Génération de rapports sur l'état des palmiers, les traitements appliqués et l'évolution des anomalies.

— **Recommandations basées sur des données historiques** : Utilisation de la NLP et des CNN pour proposer des stratégies et recommandations personnalisées

pour améliorer la gestion des palmiers.

— **Authentification sécurisée** : Mise en place d'un système d'authentification pour protéger l'accès à l'application, garantir la sécurité des données et permettre un accès personnalisé pour les utilisateurs (producteurs, investisseurs, etc.).

2.1.2 Besoins non fonctionnels

— **Performance** : L'application doit être capable de traiter un grand volume de données en temps réel ou quasi-réel pour garantir une détection rapide des anomalies.

— **Sécurité** : Les données relatives aux fermiers et à leurs produits doivent être protégées contre les accès non autorisés.

— **Compatibilité** : L'application doit être disponible sur plusieurs plateformes (mobile, web) et fonctionner aussi bien en ligne qu'hors ligne.

— **Fiabilité** : L'application doit être capable de fournir des résultats précis et de garantir un suivi continu des données.

— **Ergonomie** : L'interface utilisateur doit être simple et intuitive, adaptée aux utilisateurs peu familiers avec la technologie.

2.2 Backlog

Avant même de démarrer le premier sprint, nous avons besoin d'un backlog du produit, c'est-à-dire une liste priorisée de caractéristiques orientées client.

Le backlog du produit existe tout au long de la vie du produit. C'est la feuille de route du produit. Tout au long du projet, le backlog du produit centralise la liste de « tout ce qui peut être fait par l'équipe, par ordre de priorité ». Il n'existe qu'un seul backlog pour un produit.

Pour estimer la complexité de nos user stories, nous avons utilisé les valeurs suivantes :

ID	Fonctionnalité	ID	User Story	Priorité	Complexité
1	Gestion utilisateur	1	En tant qu'utilisateur, je souhaite m'enregistrer et me connecter de manière sécurisée afin d'accéder aux fonctionnalités de l'application.	Moyenne	8
		2	En tant qu'administrateur, je souhaite m'enregistrer et me connecter de manière sécurisée .	Faible	3
2	Identification d'anomalie	3	En tant qu'utilisateur, je souhaite télécharger une image d'une feuille de palmier pour détecter d'éventuelles maladies.	Moyenne	5
		4	En tant qu'utilisateur, je souhaite Scanner une image.	Forte	21
		5	En tant qu'utilisateur, je souhaite Recevoir la résultat du détéction avec une récommandation des traitements.	Forte	13
3	Gestion du traitement	6	En tant qu'utilisateur, je souhaite lancer le traitment du l'anomalie traiter	Forte	5
		7	En tant qu'utilisateur, je souhaite suivre le progresse du traitement lancer	Moyenne	13

ID	Fonctionnalité	ID	User Story	Priorité	Complexité
4	Statistique	8	En tant qu'utilisateur, je souhaite recevoir des statistiques sur les anomalies détectées.	Moyenne	8
5	Tableau de board	9	En tant qu'administrateur, je souhaite ajouter une anomalie	Moyenne	5
		10	En tant qu'administrateur, je souhaite gérer les utilisateurs	Faible	3

2.2.1 Les sprints

Sprint	User IDs	Complexité (calculée)
Sprint 1	1, 2, 3, 9, 10	Complexité : 24
Sprint 2	4, 5	Complexité : 34
Sprint 3	6, 7, 8	Complexité : 26

Tableau 2.2 : Répartition des sprints avec les User IDs et la complexité calculée

Où :

— **ID** : représente l'identifiant de l'user story.

— **Feature** : permet de mieux ordonner les user stories.

— **User Story** : comporte la description des user stories suivant la forme « En tant que ... Je veux ... ».

— **Priorité** : de l'user story selon la valeur métier et l'ordre de réalisation.

— **Complexité** : de la réalisation de chaque user story selon la suite de Fibonacci. Cette dernière est une suite mathématique dans laquelle chaque terme est la somme des deux termes précédents, en considérant les deux termes initiaux 0 et 1. Le début de cette suite est : 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 jusqu'à l'infini.

2.3 Diagramme de cas d'utilisation général

2.3.1 Identification des acteurs

Un acteur est une entité externe (une personne, un matériel ou un logiciel) qui interagit avec le système dans le but de réaliser une valeur de plus. Dans le cas de notre projet, les acteurs sollicitant le système sont :

— **L'administrateur** : une personne en charge de la gestion globale de l'application mobile.

— **les agricoles)** : une personne qui visite l'application mobile pour visualiser toute l'information disponible.

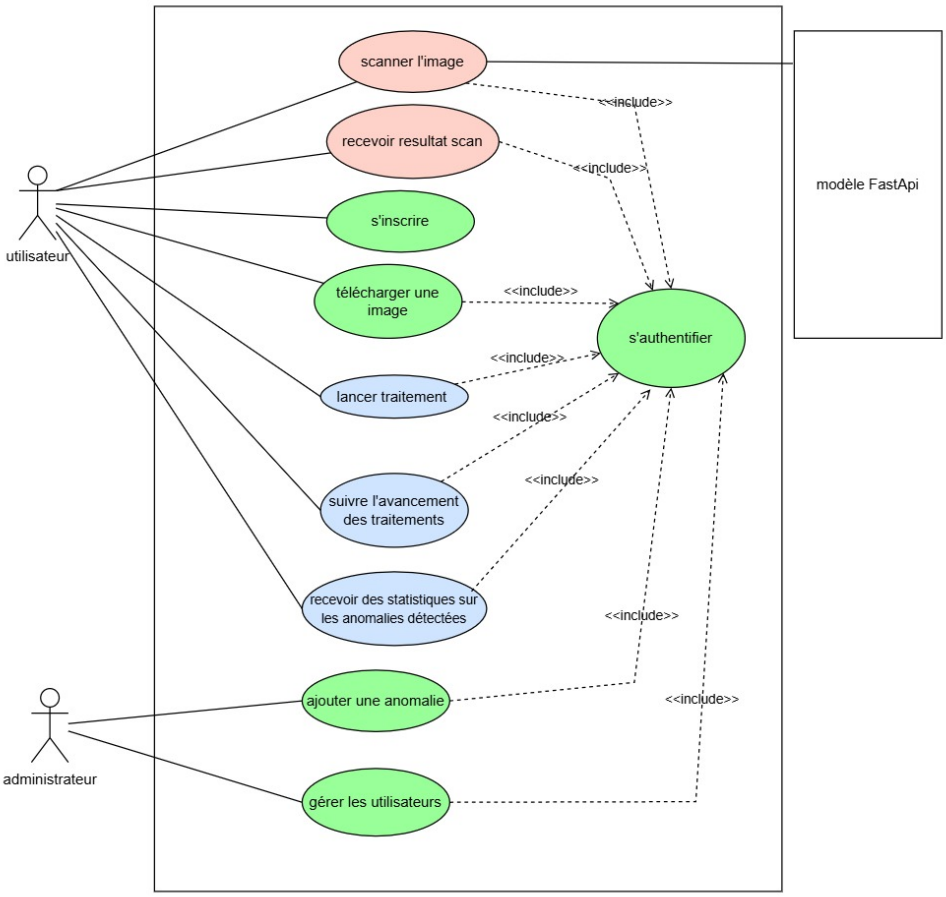


Figure 2.1 : Diagramme de cas d'utilisation

2.4 Diagramme de Classe général

Le diagramme de classes est un diagramme statique d'UML, il décrit les structures des objets et des informations utilisées par notre application. Il décrit également les informations sans référence à une implémentation particulière.

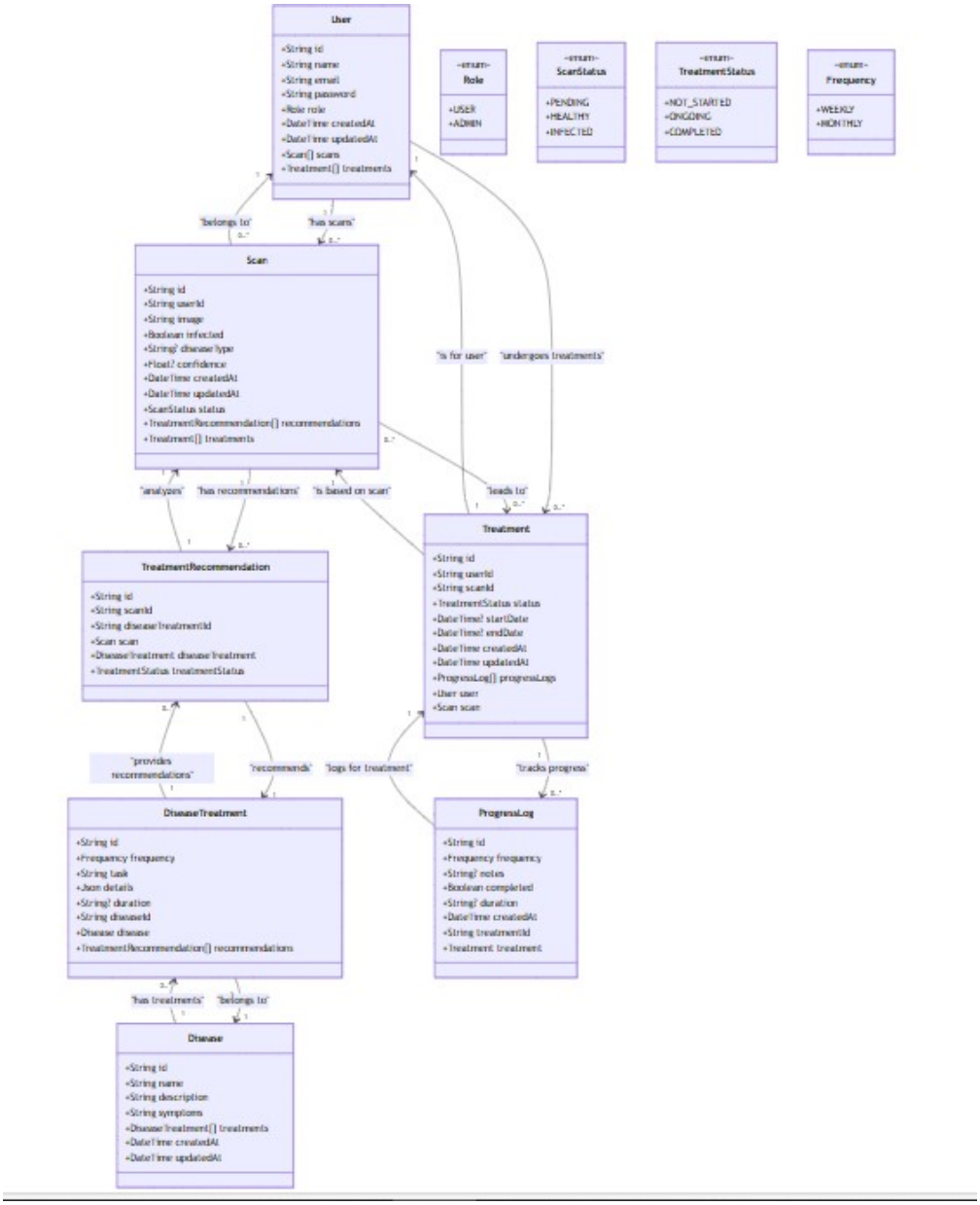


Figure 2.2 : Diagramme de classe

2.5 Architecture Physique

L'architecture à n niveaux, basée sur l'environnement client-serveur, est un modèle couramment utilisé pour organiser les différentes couches d'une application. Dans notre

projet, cette architecture est mise en œuvre de la manière suivante :

2.5.1 Couche de Présentation

La Couche de Présentation représente l'interface utilisateur et la gestion des interactions avec l'utilisateur final. Elle est responsable de la présentation des données et de la communication avec les autres couches de l'architecture.

Technologie utilisée : Flutter

Composants :

— **Application Mobile :** L'application Flutter qui s'exécute sur les appareils Android et iOS. Elle permet à l'utilisateur de prendre des photos des palmiers et de recevoir des recommandations.

— **UI/UX :** La couche de présentation comprend l'interface utilisateur qui permet de naviguer, d'afficher les anomalies détectées et d'interagir avec les fonctionnalités de l'application.

— **Appels API :** Elle envoie des requêtes HTTP au backend (NestJS) pour récupérer les anomalies et les recommandations de traitement.

2.5.2 Couche Applicative

La Couche Applicative est responsable de la logique métier, des services et de l'orchestration des opérations entre les différentes couches. Elle traite les requêtes, applique des règles métier et dirige les flux de données.

Technologie utilisée : NestJS (Backend Node.js)

Composants :

— **API REST :** Permet de gérer les interactions entre le frontend (Flutter) et le backend. Cette API expose les différents services permettant la gestion des utilisateurs, des anomalies des palmiers et des recommandations de traitement.

— **Contrôleurs :** Reçoivent les requêtes des clients et renvoient les réponses appropriées. Chaque contrôleur gère un type de fonctionnalité (par exemple, la gestion des palmiers ou la gestion des utilisateurs).

— **Services :** Contiennent la logique métier qui traite les données et effectue des opérations comme la détection des anomalies ou la génération des recommandations.

— **Sécurisation :** Mise en place de mécanismes d'authentification et d'autorisation (par exemple JWT) pour garantir la sécurité des communications entre le frontend et le backend.

2.5.3 Couche d'Objets Métier

La Couche d'Objets Métier définit les objets et les règles métiers qui modélisent les données et les entités du système.

Technologie utilisée : Node.js (avec NestJS)

Composants :

— **Modèles d'objets métier :** Représentation des données de l'application sous forme d'objets. Par exemple, un objet Palmier, Anomalie, ou Recommandation. Ces objets encapsulent les informations sur un palmier, les anomalies détectées et les recommandations de traitement.

— **Logique métier :** Contient la logique spécifique à l'application, par exemple, comment évaluer les anomalies ou comment générer des recommandations en fonction des données détectées.

— **Services de détection d'anomalies :** Mise en place d'algorithmes avancés de Machine Learning et d'Intelligence Artificielle pour analyser les images des palmiers et détecter de manière précise diverses anomalies, telles que les maladies, les parasites ou d'autres troubles affectant leur santé. Cette analyse est réalisée à l'aide de la bibliothèque FastAI basée sur PyTorch, permettant d'optimiser les performances

du modèle en termes de rapidité et de précision, tout en offrant une solution scalable et adaptable aux différents types d'images capturées par les utilisateurs.

2.5.4 Couche d'Accès aux Données

La Couche d'Accès aux Données est responsable de l'interaction avec les bases de données et les systèmes de stockage de données. Elle permet au système de persister et de récupérer les données.

Technologie utilisée : MongoDB,

Composants :

- **Base de données :** Utilisation d'une base de données NoSQL (MongoDB) pour stocker des entités telles que les utilisateurs, les palmiers, les anomalies détectées et les recommandations de traitement.
- **Système de stockage d'images :** Les images sont stockées localement dans un dossier, et les URL publiques sont enregistrées dans la base de données MongoDB.
- **ORM / ODM :** Utilisation de bibliothèques telles que Mongoose pour interagir avec la base de données depuis le backend.

2.6 Modélisation de l'Architecture Physique

Pour illustrer le déploiement de l'application, un diagramme de déploiement est utilisé pour montrer la disposition physique des différents matériels (ou nœuds) et la répartition des composants au sein des nœuds.

2.6.1 Diagramme de Déploiement

Le diagramme de déploiement représente comment les différents composants du système sont déployés dans l'infrastructure physique.

2.6.1.1 Composants

2.6.1.2 Nœuds Principaux

- **Appareil Utilisateur (Mobile) :**
  - Appareil Android qui exécute l'application Flutter.
  - Il prend des photos des palmiers, envoie les images au backend, reçoit les recommandations et suivre le progresse .
- **Serveur Backend (NestJS) :**
  - Expose des API REST pour la communication avec l'application mobile.
  - Gère la logique métier, la détection d'anomalies et la génération de recommandations.
- **Base de Données (Cloud) :**
  - Base de données NoSQL (MongoDB) pour stocker les données structurées (utilisateurs, palmiers, anomalies).
- **Service de Machine Learning / IA :**
  - Un service spécialisé pour l'analyse des images
  - Utilise des modèles de détection d'anomalies pour identifier des problèmes comme des maladies ou des parasites.
- **Admin Panel (NestJS + Next.js) :**
  - Un tableau de bord d'administration développé avec NestJS pour la gestion du backend et Next.js pour la partie frontend.
  - Permet de gérer les utilisateurs , visualiser les anomalies détectées et d'ajouter de nouvelles anomalies si nécessaire.
  - L'interface offre également des options pour consulter les recommandations de traitement et suivre l'historique des actions effectuées.
  - Le panneau d'administration est déployé sur le même serveur cloud que le backend ou sur un serveur distinct, selon l'architecture choisie.

2.7 Partie machine learning

2.7.1 Model choisi

2.7.1.1 Approches et Techniques Utilisées

Approche	Description	Points Positifs	Points Négatifs
Data Augmentation Avancée	Utilisation de transformations avancées pour augmenter la variété des données d'entraînement. Cela inclut des rotations, des zooms, des changements de luminosité, etc.	- Améliore la robustesse du modèle./ Réduit le surapprentissage. /Améliore la généralisation.	Peut augmenter le temps d'entraînement./ Peut rendre l'entraînement plus complexe.
Multi-class Classification	Classification d'images en plusieurs catégories en utilisant une architecture de Deep Learning.	- Convient bien pour des ensembles de données avec plusieurs catégories./Améliore la précision pour des classes multiples.	Peut nécessiter un ajustement fin du modèle pour des classes déséquilibrées.

Weighted Loss	Utilisation de poids ajustés pour gérer l'équilibre des classes dans le cas d'un déséquilibre des classes (par exemple, augmenter le poids de la classe "Leaf Spots").	- Aide à compenser l'imbalance des classes. / Améliore la performance sur des classes sous-représentées.	- Peut introduire un biais si les poids sont mal ajustés.
MixUp Regularization	Technique de régularisation consistant à mélanger deux images et leurs labels pour améliorer la robustesse du modèle.	- Réduit le surapprentissage. / Améliore la généralisation du modèle	Peut compliquer l'interprétation des résultats./ Peut nécessiter un ajustement fin des hyperparamètres.

Tableau 2.3 : Approches et techniques utilisées avec leurs avantages et inconvénients.

2.7.1.2 Architectures Utilisées

Architecture	Description	Points Positifs	Points Négatifs
EfficientNet-B3	Utilisation d'EfficientNet-B3, un modèle pré-entrainé optimisé pour les ressources tout en conservant une grande précision dans les tâches de classification.	- Très efficace en termes de précision par rapport à la taille du modèle./ Moins de calculs nécessaires pour une précision élevée.	- Nécessite des ajustements pour des cas d'usage spécifiques./Peut être plus complexe à optimiser par rapport à des architectures plus simples.

Tableau 2.4 : Architecture EfficientNet-B3 et ses avantages et inconvénients.

2.7.1.3 Modèles Similaires et Comparaison

Modèle	Points Positifs	Points Négatifs	Similarités avec notre modèle
ResNet-50	Excellente performance sur des jeux de données complexes. Très robuste contre le surapprentissage.	Plus lourd en termes de calcul. Peut être trop complexe pour des tâches simples.	Utilisation de CNN pour la classification. Convient à des datasets complexes.
VGG-16	Simple et bien compris. Facile à entraîner et à optimiser	Moins performant que les architectures modernes sur des tâches complexes. Modèle plus grand	Convient aux tâches de classification multi-classes. Structure CNN classique.

Xception	Très performant dans des tâches complexes./Moins de paramètres que ResNet.	Nécessite une grande quantité de données pour être efficace. Plus complexe que VGG.	Utilisation de convolutions pour extraire des caractéristiques. Classifications multi-classes.

Tableau 2.5 : Comparaison des modèles de classification d'images et leurs similarités avec notre modèle.

2.7.1.4 Résultats Attendus et Comparaison

Critère de Performance	**Modèle FastAI (EfficientNet-B3)**	**ResNet-50**	**VGG-16**	**Xception**
Précision	Élevée (grâce à l'augmentation des données et MixUp).	Très élevée, surtout pour les tâches complexes.	Moyenne à élevée.	Très élevée, surtout pour les données complexes.
Temps d'entraînement	Modéré (optimisé pour de bons résultats avec moins de calculs).	Longue durée d'entraînement.	Longue durée d'entraînement.	Modéré à long (dépend du dataset).
Facilité d'implémentation	Facile à implémenter avec FastAI.	Modérément difficile (plus de paramètres à ajuster).	Relativement facile.	Plus complexe, nécessite plus de paramètres.

Tableau 2.6 : Critères de performance des modèles de classification d'images.

Conclusion

Ce chapitre a détaillé l'analyse des besoins. En définissant clairement les besoins fonctionnels et non fonctionnels, nous avons établi une base solide pour la conception et le développement futurs. L'élaboration du backlog, tandis que les diagrammes de cas d'utilisation et de classe. Enfin, l'architecture physique proposée assure une organisation cohérente et maintenable de l'application.

Chapitre 3

Sprint 0 Installation d'environnement logiciel et matériel

Introduction

Dans ce chapitre nous allons donner un aperçu sur les environnements matériels et logiciels utilisés pour la réalisation de ce projet

3.1 Environnement matériel

En cours de réalisation de notre projet, nous avons eu à notre disposition deux ordinateurs portables qui disposent des configurations suivantes :

Tableau 3.1 : Spécifications des ordinateurs utilisés pour le projet

Environnement matériel				
1er ordinateur	**2ème ordinateur**	**3ème ordinateur**	**4ème ordinateur**	**5ème ordinateur**
				
Processeur : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz	**Processeur :** Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz	**Processeur :** Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz	**Processeur :** Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz	**Processeur :** Intel(R) Core(TM) i3 7020u GHZ CPU @ 2.50GHz 2.50 GHz
Mémoire installée (RAM) : 8.00 Go	**Mémoire installée (RAM) :** 8.00 Go	**Mémoire installée (RAM) :** 16.00 Go	**Mémoire installée (RAM) :** 16.00 Go	**Mémoire installée (RAM) :** 12.00 Go
Type de système : Windows 10 64 bits	**Type de système :** Windows 10 64 bits	**Type de système :** Windows 11-64 bits	**Type de système :** Windows 11-64 bits	**Type de système :** Windows 11-64 bits

3.2 Environnement logiciel

L'environnement logiciel pour la réalisation de notre projet comprend une série d'outils et de technologies essentiels à la conception, au développement, à la gestion et à la communication entre les membres de l'équipe. Voici une description détaillée des logiciels et plateformes utilisés :

3.2.1 Développement Backend

Pour le développement du serveur backend, nous avons utilisé **NestJS** est un framework de développement web open-source pour Node.js qui facilite la création d'applications côté serveur efficaces, évolutives et bien structurées. Il est conçu autour des concepts de la programmation orientée objet (POO), de la programmation fonctionnelle et des principes de conception modulaire. NestJS est particulièrement adapté pour créer des applications backend robustes et de grande envergure

3.2.2 Base de données

La gestion de nos données est assurée par **MongoDB**, une base de données orientée documents. MongoDB est choisie pour sa flexibilité, sa capacité à gérer de grandes quantités de données et sa facilité d'intégration avec Node.js via la bibliothèque Mongoose, qui offre des fonctionnalités de modélisation des données pour assurer une structure cohérente et sécurisée.

3.2.3 Développement Frontend

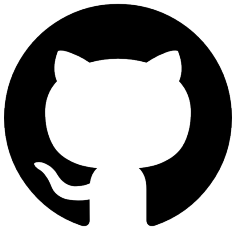
Le développement de l'application mobile est réalisé avec **Flutter** est un framework open-source développé par Google pour la création d'applications mobiles, web et de bureau (desktop) à partir d'une seule base de code. Il permet aux développeurs de créer des interfaces utilisateur (UI) modernes et performantes, tout en offrant une expérience native sur différentes plateformes. Flutter se distingue par sa capacité à offrir une grande fluidité d'animation et des performances élevées, tout en permettant de maintenir un code partagé entre les différentes plateformes.

Next.js est un framework open-source pour le développement d'applications web modernes, basé sur React. Il permet de créer des applications web performantes et optimisées en offrant des fonctionnalités puissantes pour le rendu côté serveur (SSR - Server-Side Rendering), le rendu statique (SSG - Static Site Generation) et la génération de pages au moment de la demande (ISR - Incremental Static Regeneration).

3.2.4 Tests API

Pour tester les API développées, nous avons utilisé **Swagger**(maintenant appelé OpenAPI Specification) est un ensemble d'outils et un framework qui permet de concevoir, documenter, tester et consommer des API RESTful. Il fournit une manière standardisée de définir les API web en utilisant un format lisible par machine, ce qui facilite la communication entre les développeurs, les équipes et les utilisateurs finaux. Swagger est largement utilisé pour créer de la documentation interactive et générer des tests pour les API, offrant ainsi une meilleure expérience de développement et d'intégration.

3.2.5 Intégration et gestion de versions

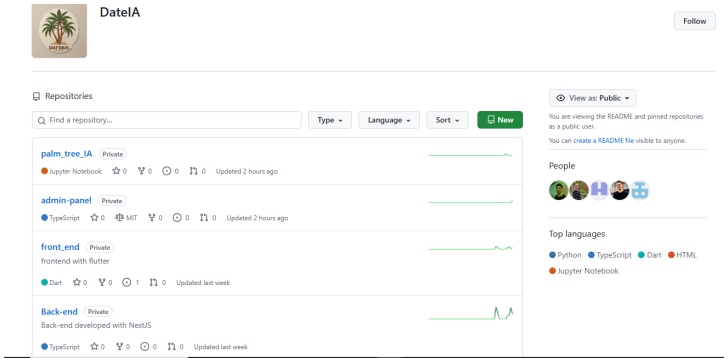


Tout notre code source est géré et versionné sur **GitHub**, ce qui facilite la collaboration entre les membres de l'équipe. GitHub sert également de plateforme centrale pour la révision de code, la gestion des branches et le suivi des modifications.

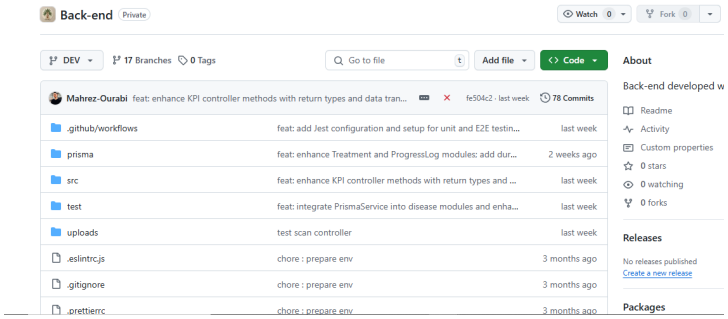
3.3 Organisation sur GitHub

Nous avons utilisé GitHub pour la gestion du code source et la collaboration sur le projet. Voici les éléments organisés sur notre dépôt GitHub :

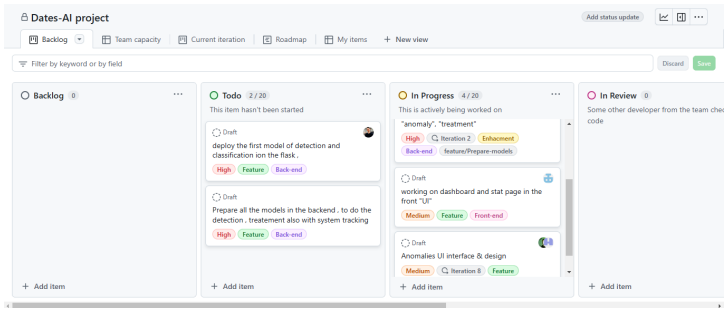
— **Dépôt principal** : Le code source principal du projet est hébergé dans un dépôt GitHub

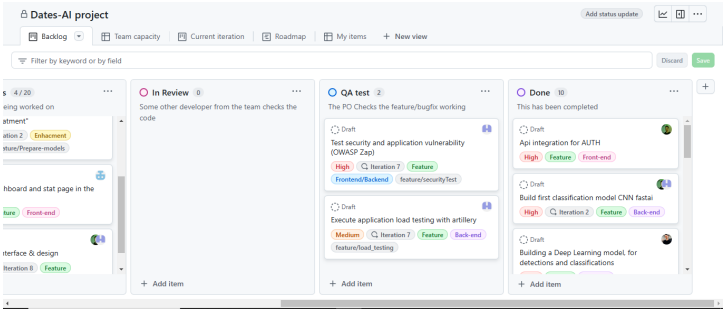


— **Branches de développement** : Le flux de travail suit une structure de branches avec des branches principales comme main pour la version stable, dev pour les fonctionnalités en développement, et des branches spécifiques pour chaque fonctionnalité (par exemple, feature/anomaly-detection)

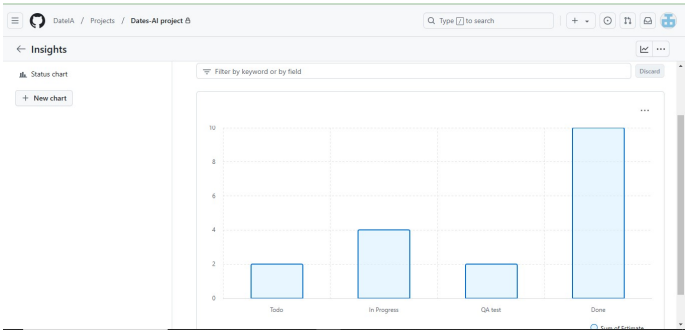


— **Issues et Discussion** : Nous avons utilisé les "Issues" de GitHub pour suivre les tâches à réaliser, les problèmes à résoudre et les fonctionnalités à implémenter. Les discussions sur chaque issue permettent de suivre les progrès et de collaborer efficacement sur les détails techniques.





— **Suivi de l'Avancement : Status Chart** : Pour suivre l'avancement du projet, nous avons utilisé des status charts (tableaux de statut) qui fournissent une vue d'ensemble claire du progrès des différentes tâches dans le backlog. Ces graphiques permettent de suivre l'état d'avancement des tâches sur le planification du sprint.



Conclusion

Tout au long de ce chapitre, nous avons présenté l'environnement matériel et logiciel utilisé, ainsi que les choix technologiques effectués, ainsi que l'organisation et la planification sur GitHub. Dans le chapitre suivant, nous détaillerons la réalisation de notre travail et les résultats obtenus.

Chapitre 4

Développement du Sprint 1

Introduction

Dans ce premier sprint, notre objectif était de poser les bases de notre application, en intégrant les principales fonctionnalités nécessaires à la création de comptes utilisateurs et à la gestion de leurs sessions.

4.1 Planification du Sprint

La planification a consisté à sélectionner les user stories issues de notre backlog, jugées essentielles pour la fondation de notre plateforme. Ces tâches ont été estimées et assignées aux développeurs en fonction de leurs expertises et de leurs disponibilités.

4.2 Backlog du Sprint 1

Une réunion est organisée au début de chaque sprint afin de planifier les tâches sélectionnées depuis le backlog du produit et d'établir leurs priorités, en vue de concevoir un livrable à la fin du sprint.

— Date de début du sprint : 9 février 2024

— Date de fin du sprint : 8 mars 2024

— Temps estimé en heures : 16 heures

— Échelle de mesure : une journée ouvrable équivaut à 8h de travail

— Objectif du sprint : Authentifier les administrateurs, ajouter et gérer des chambres, et ajouter une tâche.

Éléments du backlog et Tâches

ID	Fonctionnalité	ID	User Story	Priorité	Complexité
1	Gestion utilisateur	1	En tant qu'utilisateur, je souhaite m'enregistrer et me connecter de manière sécurisée afin d'accéder aux fonctionnalités de l'application.	Moyenne	8
		2	En tant qu'administrateur, je souhaite m'enregistrer et me connecter de manière sécurisée.	Faible	3
2	Identification d'anomalie	3	En tant qu'utilisateur, je souhaite télécharger une image d'une feuille de palmier pour détecter d'éventuelles maladies	Moyenne	5
5	Tableau de bord	9	En tant qu'administrateur, je souhaite ajouter une anomalie.	Moyenne	5
		10	En tant qu'administrateur, je souhaite gérer les utilisateurs.	Faible	3

4.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation pour notre premier sprint est présenté dans la figure **??**.

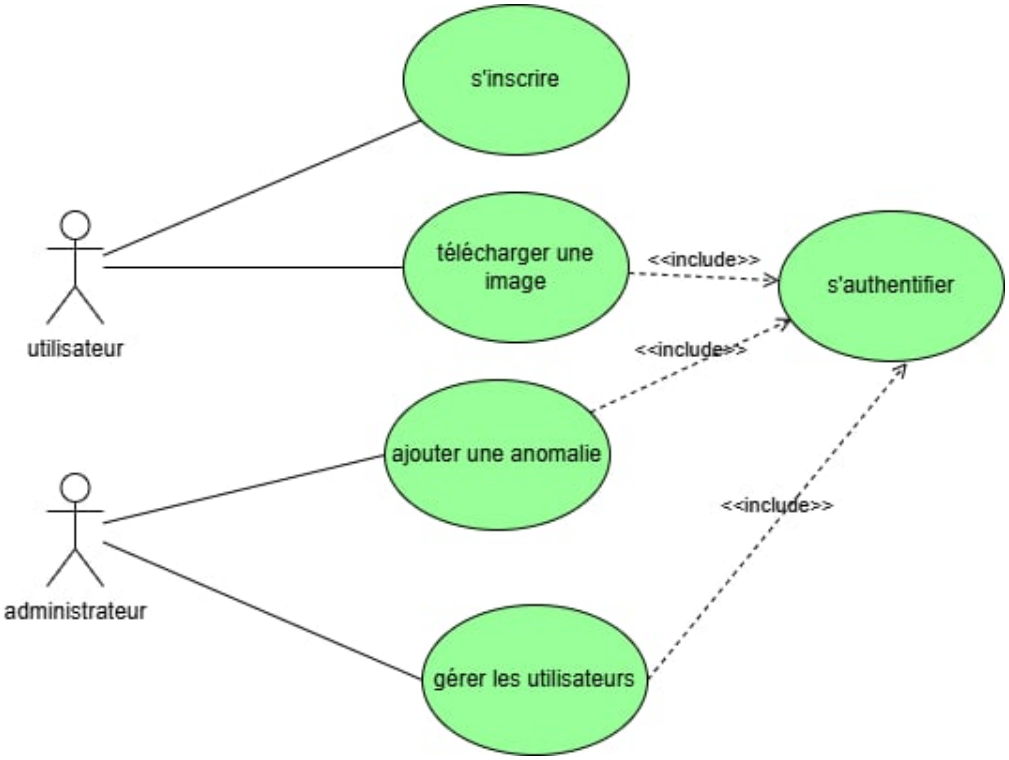


Figure 4.1 : Diagramme de cas d'utilisation du sprint 1

4.4 Les interfaces de l'application

Dans cette section, nous présentons les interfaces réelles de l'application. Chaque capture d'écran illustre une fonctionnalité clé du système et montre comment l'utilisateur interagit avec l'application.

4.4.1 L'interface « d'inscription »

Dans cette section, nous présentons les interfaces réelles de l'application. Chaque capture d'écran illustre une fonctionnalité clé du système et montre comment l'utilisateur interagit avec l'application.

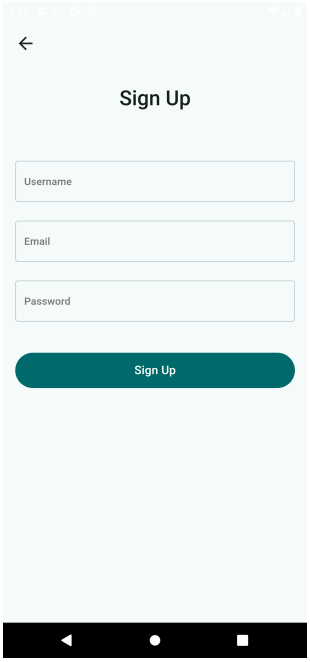


Figure 4.2 : Sign-up

Cas de succès :

- Utilisateur inscrit.

Cas d'échec :

- L'email est déjà présent dans la base de données.
- Informations non complètes.
- Email non validé.
- Un message d'erreur est affiché.

4.4.2 L'interface « connexion »

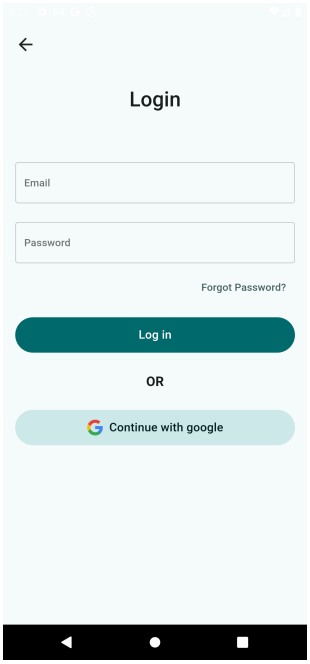


Figure 4.3 : Login

Cas de succès :

- Utilisateur connecté.
- Accès au système.

Cas d'échec :

- Email ou mot de passe non saisis.
- Mot de passe incorrect.
- Email non trouvé dans la base de données.
- Un message d'erreur est affiché.

4.4.3 Linterface « d'importation de photo »

Figure 4.4 : Camera

Cas de succès :

- Utilisateur connecté.
- Accès à la galerie.
- Prendre une photo avec l'appareil photo.

Cas d'échec :

- Données non renseignées.
- Un message d'erreur est affiché.

4.5 Tests effectués

Dans cette section, nous détaillons les différents tests réalisés pour garantir le bon fonctionnement des fonctionnalités d'authentification des utilisateurs, telles que la

création d'un compte, la connexion et la déconnexion, ainsi que les tests de sécurité, les tests unitaires et les tests de performance.

4.5.1 Cas d'utilisation liés à l'authentification

Les cas d'utilisation identifiés dans la phase de conception incluent des scénarios autour de l'authentification des utilisateurs. Cela permet de garantir que seuls les utilisateurs authentifiés peuvent accéder à certaines fonctionnalités sensibles telles que la consultation des rapports, l'historique des analyses et la gestion des informations personnelles.

4.5.1.1 Création d'un compte utilisateur

Description : Un utilisateur peut créer un compte dans l'application en fournissant une adresse email et un mot de passe sécurisé.

Cas de test : Vérifier que le processus d'inscription fonctionne correctement, que l'utilisateur reçoit un email de confirmation, et que les données sont correctement enregistrées dans la base de données MongoDB.

4.5.1.2 Connexion utilisateur

Description : Un utilisateur peut se connecter à l'application avec son adresse email et son mot de passe. La connexion doit être sécurisée, et la session doit être maintenue pendant la durée de l'utilisation.

Cas de test : Vérifier que l'utilisateur peut se connecter avec un mot de passe valide et que l'authentification renvoie une session valide.

4.5.1.3 Déconnexion

Description : L'utilisateur peut se déconnecter de l'application en toute sécurité.

Cas de test : Vérifier que la déconnexion supprime la session active et redirige

l'utilisateur vers la page d'accueil ou de connexion.

4.5.2 Tests de Sécurité

Les tests de sécurité ont été réalisés pour assurer que les informations des utilisateurs sont correctement protégées et que les systèmes sont protégés contre les attaques courantes.

Test de sécurité pour la création d'un compte utilisateur : S'assurer que l'application valide correctement les mots de passe, applique les règles de sécurité (par exemple, complexité du mot de passe) et empêche les attaques par injection SQL ou XSS (Cross-Site Scripting).

Test de la connexion sécurisée : Vérifier que les mots de passe sont correctement hachés et que les connexions sont sécurisées via HTTPS pour éviter les interceptions de données sensibles.

4.5.3 Tests unitaires

Les tests unitaires jouent un rôle crucial dans la validation du bon fonctionnement des composants individuels de l'application, en particulier les logiques métier relatives aux fonctionnalités de login et signup.

4.5.3.1 Tests du Contrôleur de Login et Signup

Les tests du contrôleur de login et signup visent à assurer que les entrées de l'utilisateur sont correctement traitées et que les réponses renvoyées par le serveur respectent les règles métier définies.

Test de la création d'un utilisateur (Signup) :

Description : Vérification que le contrôleur gère correctement la création d'un nouvel utilisateur avec des informations valides.

Résultat attendu : Le contrôleur doit appeler le service associé, effectuer les

validations nécessaires (par exemple, le format de l'email), et renvoyer un code de réponse 201 avec les informations de l'utilisateur créé.

Solution proposée : Vérification des erreurs possibles comme un email déjà existant ou un mot de passe trop faible.

Test de la connexion d'un utilisateur (Login) :

Description : Vérification que le contrôleur effectue une validation correcte des identifiants de l'utilisateur (email et mot de passe).

Résultat attendu : En cas de succès, le contrôleur doit renvoyer un code 200 et un token d'authentification. En cas d'échec, un message d'erreur approprié doit être retourné (par exemple, "Identifiants incorrects").

Solution proposée : Validation des erreurs comme un mot de passe incorrect ou un utilisateur inexistant.

4.5.3.2 Tests du Service de Login et Signup

Les services de login et signup sont responsables de la logique métier de gestion des utilisateurs. Ces tests assurent que les règles et les comportements définis pour l'authentification et l'enregistrement d'utilisateurs sont respectés.

Test de l'inscription d'un nouvel utilisateur :

Description : Le service vérifie que les informations d'inscription (email, mot de passe, etc.) sont valides, que l'utilisateur n'existe pas déjà, et procède à la création de l'utilisateur.

Résultat attendu : Le service doit renvoyer l'utilisateur nouvellement créé ou un message d'erreur si l'email est déjà pris ou si les validations échouent.

Solution proposée : S'assurer que le mot de passe respecte les règles de complexité et que l'email n'est pas déjà utilisé.

Test de l'authentification d'un utilisateur existant :

Description : Le service authentifie un utilisateur en vérifiant ses identifiants et génère un token JWT.

Résultat attendu : Si l'authentification est réussie, le service doit renvoyer un token valide. Si l'utilisateur n'est pas trouvé ou si le mot de passe est incorrect, une erreur appropriée doit être levée.

Solution proposée : Vérification de la gestion des erreurs, comme l'échec de la connexion ou des données invalides.

4.5.4 Tests de Performance

Des tests de performance ont été réalisés pour garantir que l'application peut supporter une charge d'utilisateurs croissante sans impact majeur sur ses performances. Ces tests ont été effectués à l'aide de JMeter et Swagger, qui permettent de simuler des charges de travail sur l'application et de mesurer la vitesse de réponse du serveur.

Tests de Performance avec JMeter : Simulation de multiples connexions simultanées pour évaluer la réactivité du serveur et le traitement des demandes d'authentification.

Tests de Performance avec Swagger : Utilisation de Swagger pour tester les API d'authentification et vérifier les délais de réponse en fonction de la charge.

4.5.5 Tests Front-End (Flutter)

Les tests front-end ont été réalisés sur les écrans et services de login et signup développés avec Flutter. Ces tests ont été divisés en deux catégories principales : les tests unitaires, pour vérifier le bon fonctionnement des composants individuels, et les tests d'intégration, pour évaluer le comportement global de l'application lorsqu'elle interagit avec les différents services et écrans.

4.5.5.1 Tests Unitaires du Login et Signup

Test de l'écran de login :

Description : Vérification que l'écran de login prend correctement les informations d'entrée de l'utilisateur (email et mot de passe), affiche les messages d'erreur appropriés et déclenche les actions de connexion.

Résultat attendu : L'écran doit afficher les erreurs liées à un email invalide ou un mot de passe incorrect, et appeler le service de login avec les bonnes données.

Test du service de login :

Description : Vérification que le service de login effectue correctement l'authentification en appelant l'API de manière appropriée et renvoie un token ou un message d'erreur en fonction du résultat.

Test de l'écran de signup :

Description : Vérification que l'écran de signup prend correctement les informations nécessaires pour l'inscription (email, mot de passe, etc.), valide les données et appelle le service de création de compte.

4.5.5.2 Tests d'Intégration du Login et Signup

Les tests d'intégration ont été réalisés pour vérifier que les écrans de login et signup interagissent correctement avec leurs services respectifs, en simulant le parcours complet de l'utilisateur, depuis l'entrée des données jusqu'à la réception des réponses appropriées.

Chapitre 5

Backlog du Sprint 2

5.1 Introduction

Le Sprint 2 est consacré à l'intégration de la fonctionnalité d'analyse d'images au sein de notre application. L'objectif principal est de mettre en place un modèle de machine learning (ML) capable de traiter des images scannées, de détecter des anomalies et de fournir des recommandations sur les traitements appropriés. Nous allons utiliser FastAI, un framework performant pour le ML, afin de concevoir et déployer ces modèles.

5.2 Planification du Sprint

La planification de ce sprint a consisté à définir les user stories essentielles liées à l'apprentissage automatique et à la sélection des algorithmes les mieux adaptés à notre projet. Nous avons estimé et priorisé les tâches en fonction des besoins en ML, du temps nécessaire pour entraîner les modèles et des ressources à disposition.

5.3 Backlog du Sprint 2

Une réunion est organisée au début de chaque sprint pour planifier les tâches à partir du backlog du produit, en définissant les priorités pour concevoir un livrable à la fin du sprint.

— Date de début du sprint : 9 février 2024

— Date de fin du sprint : 8 mars 2024

— Temps estimé en heures : 40 heures

— Échelle de mesure : une journée ouvrable équivaut à 8h de travail

— Objectif du sprint : Sélectionner des algorithmes ML appropriés pour la détection d'anomalies dans les images et implémenter un modèle de machine learning avec FastAI.

Éléments du backlog et Tâches

ID	Fonctionnalité	ID	User Story	Priorité	Complexité
2	Analyse d'image avec ML	4	En tant qu'utilisateur, je souhaite que l'application analyse une image afin de détecter des anomalies spécifiques.	Forte	21
		5	En tant qu'utilisateur, je souhaite recevoir des recommandations de traitement adaptées après détection des anomalies dans l'image.	Forte	13

Tâches spécifiques

— Sélectionner un algorithme de machine learning adapté pour l'analyse d'images (réseaux neuronaux convolutifs, par exemple).

— Préparer un jeu de données d'images pour l'entraînement du modèle.

— Implémenter un modèle de détection d'anomalies à l'aide de FastAI.

— Effectuer un prétraitement des images (normalisation, augmentation de données).

— Entraîner le modèle avec des images d'exemples pour détecter les anomalies.

— Évaluer les performances du modèle (précision, rappel, F1-score).

— Implémenter une interface pour afficher les résultats de la détection et recommander des traitements.

5.4 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation pour ce sprint est modifié pour inclure les interactions avec les fonctionnalités ML.

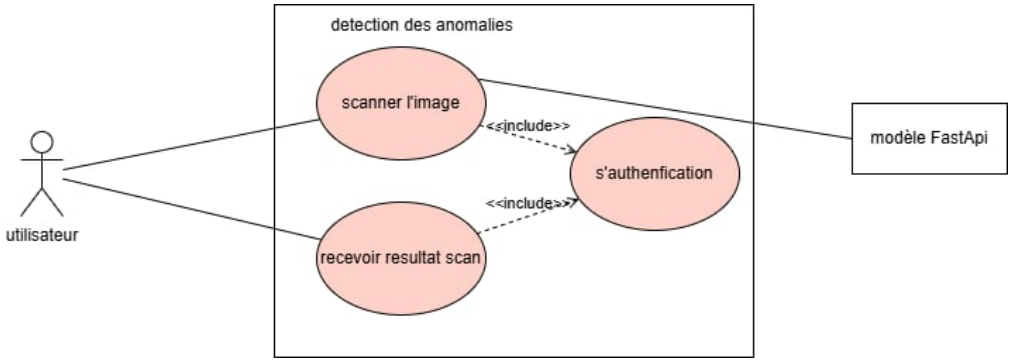


Figure 5.1 : Diagramme de cas d'utilisation du sprint 2

5.5 Les interfaces de l'application

Dans cette section, nous présentons les interfaces réelles de l'application. Chaque capture d'écran illustre une fonctionnalité clé du système et montre comment l'utilisateur interagit avec l'application.

5.5.1 Interface de résultats de détection

Après l'analyse de l'image, l'utilisateur pourra voir les anomalies détectées, accompagnées d'une recommandation de traitement. Ce processus est automatisé via le modèle de machine learning développé.

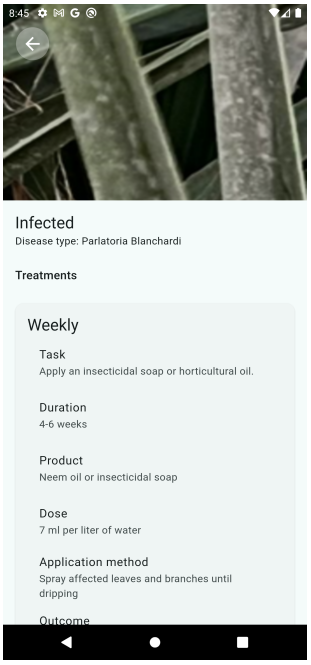


Figure 5.2 : scan results

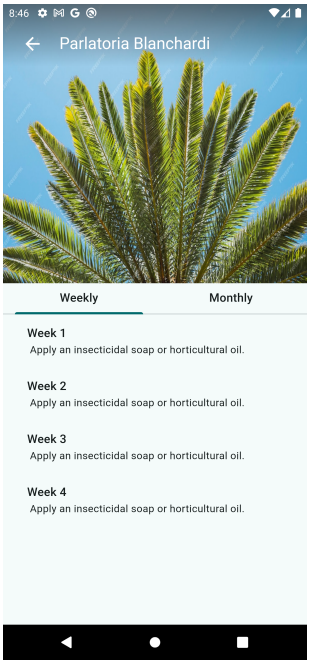


Figure 5.3 : scan treatment recommendation

5.6 Tests effectués

Des tests seront réalisés pour garantir que le modèle de ML fonctionne de manière optimale. Les tests incluent l'évaluation du modèle sur un ensemble d'images de test et la vérification de l'interface utilisateur pour la présentation des résultats.

5.6.1 Tests de modèle ML

Des tests seront effectués pour valider les performances du modèle de machine learning, notamment en mesurant la précision et l'efficacité de la détection des anomalies. L'utilisation de FastAI facilite l'évaluation du modèle grâce à ses outils d'optimisation et de visualisation.

5.6.2 Tests d'interface utilisateur

L'interface utilisateur sera testée pour s'assurer que les images sont correctement importées et que les résultats du modèle sont présentés de manière claire et compréhensible. Des tests d'intégration seront effectués pour s'assurer que la communication entre le backend ML et le frontend fonctionne correctement.

5.6.3 Tests de performance ML

Des tests de performance seront réalisés pour évaluer la vitesse et l'efficacité du modèle de machine learning, en particulier pour garantir une exécution rapide lors de l'analyse des images en temps réel.

5.7 Conclusion

À la fin de ce sprint, l'objectif est de disposer d'un modèle de machine learning entraîné pour la détection d'anomalies dans les images, intégré dans l'application via

FastAI. Ce modèle devra être capable de fournir des recommandations de traitement adaptées à l'utilisateur, tout en étant optimisé pour des performances élevées.

Chapitre 6

Backlog du Sprint 3

6.1 Introduction

Le Sprint 3 se concentre sur la gestion du traitement des anomalies détectées. L'objectif est de permettre aux utilisateurs de lancer des traitements sur les anomalies détectées, de suivre l'avancement de ces traitements, et de recevoir des statistiques détaillées sur les anomalies détectées dans les images analysées. Ce sprint mettra en place des fonctionnalités permettant d'améliorer l'interaction de l'utilisateur avec l'application en termes de gestion des traitements et de suivi.

6.2 Planification du Sprint

La planification de ce sprint a pour objectif de fournir une interface conviviale pour lancer et suivre les traitements des anomalies détectées, ainsi que d'intégrer un module statistique pour fournir des informations détaillées sur les anomalies. Nous avons estimé et priorisé les tâches en fonction de leur impact sur la satisfaction de l'utilisateur et de la faisabilité technique.

6.3 Backlog du Sprint 3

Une réunion est organisée au début de chaque sprint pour planifier les tâches à partir du backlog du produit, en définissant les priorités et les ressources nécessaires à la réalisation des objectifs.

— Date de début du sprint : 9 mars 2024

— Date de fin du sprint : 6 avril 2024

— Temps estimé en heures : 40 heures

— Échelle de mesure : une journée ouvrable équivaut à 8h de travail

— Objectif du sprint : Implémenter la gestion du traitement des anomalies, le suivi du traitement et la génération de statistiques sur les anomalies.

Éléments du backlog et Tâches

ID	Fonctionnalité	ID	User Story	Priorité	Complexité
3	Gestion du traitement	6	En tant qu'utilisateur, je souhaite lancer le traitement des anomalies détectées.	Forte	5
		7	En tant qu'utilisateur, je souhaite suivre l'avancement du traitement lancé.	Moyenne	13
4	Statistique	8	En tant qu'utilisateur, je souhaite recevoir des statistiques détaillées sur les anomalies détectées.	Moyenne	8

Tâches spécifiques

— Développer une fonctionnalité pour permettre à l'utilisateur de lancer le traitement des anomalies détectées dans les images.

— Concevoir un système de suivi du traitement des anomalies, permettant à l'utilisateur de voir l'avancement du traitement en temps réel.

— Implémenter un module de statistiques pour fournir des informations détaillées sur les anomalies détectées, y compris des graphiques et des statistiques sur le nombre d'anomalies, leur type, etc.

— Tester la fonctionnalité de lancement et de suivi des traitements pour garantir qu'elle fonctionne de manière fluide et réactive.

— Tester le module statistique pour s'assurer que les données collectées sont correctes et présentées de manière claire à travers des KPI (Key Performance Indicators).

6.4 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation pour ce sprint est adapté pour inclure les nouvelles interactions liées à la gestion des traitements et à la réception des statistiques sur les anomalies.

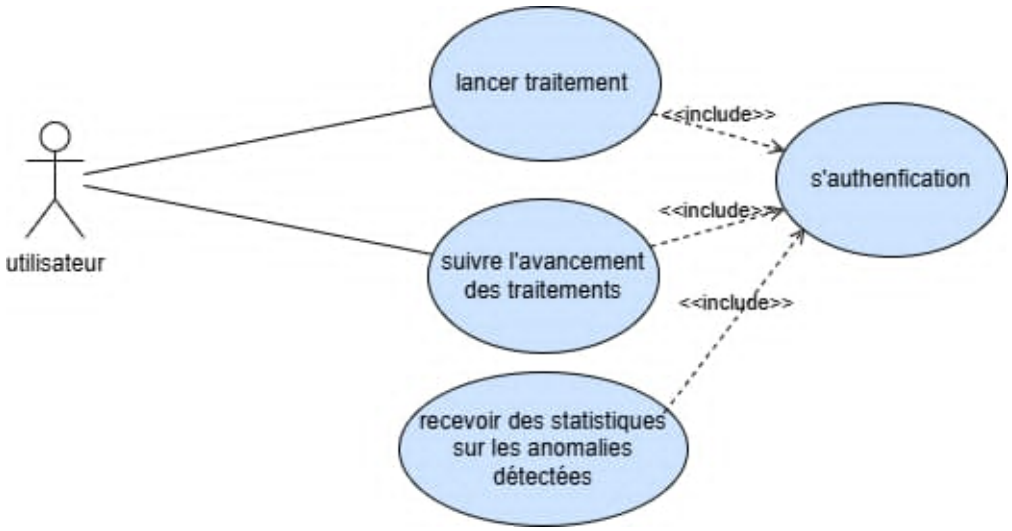


Figure 6.1 : Diagramme de cas d'utilisation du sprint 3

6.5 Les interfaces de l'application

Dans cette section, nous présentons les interfaces réelles de l'application. Chaque capture d'écran illustre une fonctionnalité clé du système et montre comment l'utilisateur interagit avec l'application.

6.5.1 Interface de statistiques

Une interface permettant à l'utilisateur de consulter des statistiques détaillées sur les anomalies détectées, telles que le nombre d'anomalies par type, la fréquence d'occurrence, et des graphiques représentatifs.

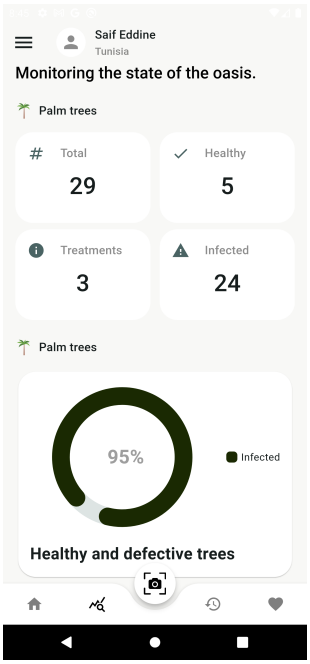


Figure 6.2 : Dashboard

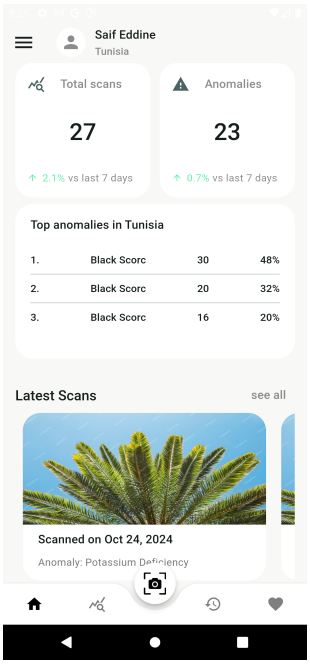


Figure 6.3 : home page

6.6 Tests effectués

Les tests seront réalisés pour s'assurer que les fonctionnalités de gestion des traitements, de suivi et de statistiques sont correctement implémentées et offrent une expérience utilisateur optimale.

6.6.1 Tests de la fonctionnalité de lancement du traitement

Des tests seront réalisés pour vérifier que le traitement peut être lancé correctement et que l'utilisateur reçoit des retours appropriés sur l'état du traitement.

6.6.2 Tests de suivi du traitement

Des tests de performance et de réactivité seront effectués pour garantir que l'avancement du traitement est suivi en temps réel, sans délais importants.

6.6.3 Tests du module de statistiques

Le module de statistiques sera testé pour s'assurer que les données collectées sur les anomalies sont correctes et que les graphiques sont bien générés et lisibles.

6.7 Conclusion

À la fin de ce sprint, l'objectif est de fournir une gestion complète des traitements d'anomalies détectées, de permettre le suivi en temps réel du traitement et de donner aux utilisateurs des statistiques détaillées sur les anomalies, afin de leur offrir un meilleur contrôle et une meilleure compréhension des résultats des analyses effectuées.

Conclusion

Le projet a permis de développer une application innovante visant à détecter, traiter et analyser des anomalies à partir d'images scannées. Au fil des sprints, plusieurs fonctionnalités clés ont été mises en œuvre, allant de l'intégration d'algorithmes de machine learning pour la détection d'anomalies à la gestion du traitement et des statistiques.

Au Sprint 1, nous avons posé les bases du système en implémentant l'authentification des utilisateurs et la gestion des données. Ce sprint a permis de s'assurer que l'application pouvait gérer efficacement les utilisateurs et leur offrir une interface fonctionnelle.

Le Sprint 2 a concentré nos efforts sur l'intégration du scanner d'image et le développement de modèles de machine learning pour la détection des anomalies. Nous avons opté pour l'utilisation de FastAI afin de bénéficier d'algorithmes adaptés aux besoins de notre application, en tenant compte des ressources matérielles disponibles.

Le Sprint 3, quant à lui, a permis de finaliser les fonctionnalités liées au traitement des anomalies, avec l'ajout de la possibilité de suivre l'avancement des traitements lancés et de recevoir des statistiques détaillées sur les anomalies détectées. Ces fonctionnalités permettent non seulement d'améliorer l'expérience utilisateur, mais aussi d'apporter une valeur ajoutée en termes d'analyse et de recommandations basées sur les données collectées.

En conclusion, ce projet a permis d'atteindre les objectifs définis, tout en intégrant les bonnes pratiques du développement agile, grâce à des sprints bien structurés et une gestion efficace du backlog. Nous avons pu fournir une application fonctionnelle et évolutive, qui pourra être enrichie par de futures améliorations et optimisations. Le travail

effectué dans les sprints suivants permettra de renforcer l'applicabilité et la robustesse du système en vue de répondre aux besoins des utilisateurs finaux et de garantir une solution fiable et performante.

Nétéographie

- **FastAI Documentation** : La bibliothèque FastAI est un outil puissant et simple pour la création de modèles d'apprentissage automatique, notamment pour la vision par ordinateur. La documentation officielle présente en détail l'utilisation de techniques comme l'augmentation de données, les "DataBlocks" et les architectures préentraînées telles qu'EfficientNet.
  URL : https://docs.fast.ai/
- **EfficientNet** : EfficientNet est une architecture de réseau neuronal qui optimise l'utilisation des ressources tout en maximisant la précision des modèles. La première version d'EfficientNet a été publiée dans l'article "EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks" (2019).
  URL : https://arxiv.org/abs/1905.11946
- **Data Augmentation** : L'augmentation de données est une technique importante pour améliorer les performances des modèles, en particulier lorsque le nombre de données d'entraînement est limité. Pour un aperçu complet de l'augmentation de données en vision par ordinateur, consultez l'article "Understanding Data Augmentation for Deep Learning" (2020).
  URL : https://arxiv.org/abs/2005.11528
- **MixUp** : La régularisation MixUp consiste à créer de nouvelles images en mélangeant les pixels d'images existantes et leurs étiquettes. Le papier original de MixUp est "Mixup : Beyond Empirical Risk Minimization" (2018).
  URL : https://arxiv.org/abs/1710.09412
- **ResNet (Residual Networks)** : ResNet est une architecture de réseau qui introduit

des "skip connections" ou connexions résiduelles pour aider à l'apprentissage de modèles plus profonds sans perte de précision. L'article original de ResNet est intitulé "Deep Residual Learning for Image Recognition" (2015).
URL : https://arxiv.org/abs/1512.03385

- **ResNet-50** : C'est une version de ResNet avec 50 couches qui est couramment utilisée dans les tâches de classification d'images. La documentation et les implémentations sont largement disponibles sur des plateformes comme PyTorch et TensorFlow.
- **VGG-16** : VGG-16 est un modèle de réseau neuronal convolutif très populaire, connu pour sa simplicité et son efficacité. Le papier original "Very Deep Convolutional Networks for Large-Scale Image Recognition" (2014) présente cette architecture.
  URL : https://arxiv.org/abs/1409.1556
- **VGG-16 in PyTorch** : Vous pouvez consulter des ressources et des tutoriels pour l'implémentation de VGG-16 en PyTorch.
  URL : https://pytorch.org/vision/stable/models.html
- **Xception** : Xception est une architecture basée sur des convolutions séparables en profondeur et est une amélioration des architectures Inception. Le papier original est intitulé "Xception : Deep Learning with Depthwise Separable Convolutions" (2017).
  URL : https://arxiv.org/abs/1610.02357
- **Xception in TensorFlow/Keras** : Xception est implémenté dans TensorFlow et Keras, et des tutoriels détaillent son utilisation.
  URL : https://www.tensorflow.org/api$_d ocs/python/tf/keras/applications/Xception$
- Flutter UI to browse and order plants from a nursery
- YouTube tutorial on neural networks
- Download link for data
- ScienceDirect article
- ResearchGate article on date fruit sorting

- Xingseus app on Google Play
- PlantNet app on Google Play
- Agrio app on Google Play
- Oracle Cloud AI Foundations Learning Path
- Oracle Cloud AI Foundations Exam
- Agriculture Vegetables Fruits Time Series Prices on Kaggle
- Date Palm Data on Kaggle
- Date Fruit Image Data on Kaggle