



Projet de Fin de Semestre

5ème année Génie informatique

Conception et développement d'une application mobile



« SuperEYE Search Engine (SSE) »

Application Mobile d'IA Agricole avec Moteur de Recherche Intelligent Intégré

Réalisé par :

-AHMAMO HAMZA
-AOURIR AISSAM
-ER-RETBY ABDELILAH
-HMIDDOUCH ABDESSADEK

Encadré par :

-Dr. BEKARRI AISSAM
-Dr. ATLAS ABDELGHAFOUR

Soutenu le 05/01/2026

Année universitaire 2025/2026

Remerciements :

À l'issue de ce projet de fin de semestre, nous souhaitons adresser nos sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin, à sa réussite. Leur soutien et leurs encouragements ont été essentiels tout au long de cette expérience enrichissante.

Nous exprimons tout particulièrement notre profonde gratitude à **M. Aissam Bekarri** et Dr. Atlas Abdelghafour , notre encadrant et professeur, pour sa disponibilité constante, ses conseils avisés et son accompagnement attentif tout au long du projet. Son expertise en intelligence artificielle, sa pédagogie exemplaire et ses orientations précieuses ont été une source d'inspiration permanente et nous ont permis de mener à bien le développement de l'application mobile SuperEYE dans des conditions optimales.

Ce travail constitue une étape importante dans notre parcours académique à l'ENSA Marrakech. Il nous a permis d'acquérir des compétences techniques solides en développement mobile, vision par ordinateur et recherche d'information, ainsi que des connaissances pratiques qui nous seront d'une grande utilité pour notre avenir professionnel.

Nous tenons enfin à remercier l'ensemble du corps enseignant du département Génie Informatique pour la qualité de la formation dispensée durant ces années d'études.

Résumé :

Ce projet a pour objectif la conception et le développement d'une application mobile intelligente dédiée au diagnostic biologique agricole, visant à moderniser et à faciliter l'identification rapide des espèces végétales et animales ainsi que la détection de leurs maladies ou anomalies.

À travers une plateforme mobile accessible, les utilisateurs (agriculteurs, éleveurs, citoyens) peuvent scanner une plante directement via la caméra de leur smartphone, obtenir un diagnostic instantané par intelligence artificielle, le diagnostic, les symptômes détectés et les recommandations de traitement. Le diagnostic est enrichi par un moteur de recherche intelligent hybride qui recherche automatiquement parmi une base de fiches techniques expertes et affiche les cas similaires les plus pertinents.

L'application propose également un espace administrateur sécurisé permettant la gestion des utilisateurs, la validation manuelle des scans sensibles ou rares, le suivi des statistiques d'utilisation (scans les plus fréquents, maladies détectées, tendances régionales) et la maintenance de la base de ressources documentaires.

Ce projet s'inscrit dans une logique d'innovation appliquée au secteur agricole marocain, en réponse aux besoins croissants de rapidité, de précision et d'accessibilité dans le diagnostic des pathologies des cultures et du bétail, contribuant ainsi à une agriculture plus durable, informée et résiliente.

Table de matières:

I-Introduction générale :	6
----------------------------------	----------

1-Contexte générale du projet.....	6
2-Motivation et objectifs:.....	6
II-Analyse et conception.....	8
1-Analyse des besoins.....	8
1-1)-Spécification des besoins fonctionnels :.....	8
1-2)-Spécification des besoins non fonctionnels :.....	11
2-Conception.....	13
1)-Diagramme de cad'utilisation :.....	13
2)-Diagramme de classe :	14
3)-Diagramme de séquence :.....	14
3-Planning du Projet.....	14
III- Outils et Technologies utilisés.....	15
IV-Réalisation.....	21
1-Architecture de projet:.....	21
1)- Service Gestion SuperEYE(Service Application Principal).....	21
2)- Service IA Detection.....	22
3)- Service Système de Recherche d'Informations (SRI).....	22
2-Présentation des interfaces:.....	23
1)-Côté chercheur.....	23
2)-Côté client.....	27
3)-Côté admin.....	27
3- Modules spécifiques de SUPEREYE	28
3-1) Module SRI (Système de Recherche d'Informations)	28
3-1-1) Thématique du Moteur de Recherche	28
3-1-2) Fonctionnalités principales	29
3-1-3) Processus d'indexation et de recherche	29
3-1-3-1) Indexation : automatique sans prétraitement linguistique avancé	30
3-1-3-2) Stockage	30
3-1-3-3) Recherche : modèle vectoriel avec similarité cosinus	31

3-1) Module IA de détection de maladies	30
3-1-1) Fonctionnement général	30
3-1-2) Détection de la maladie et type de plante	31
3-1-3) Interaction avec le SRI pour générer les réponses	31
3-1-4) Méthodologie et évaluation	33
3-2) Module IA de détection de maladies	35
V-Conclusion.....	37

Table des figures :

Figure 1:Diagramme de cas d'utilisation	11
.....	11
Figure 2:Diagramme de classe.....	12
Figure30 :Diagramme de séquence.....	13
Figure 3:Diagramme de Gantt	14
Figure 4:Architecture du projet.....	19
Figure 5:Architecture du backend	20
Figure 6:Architecture du frontend	21

I-Introduction générale :

1-Contexte générale du projet

Dans un monde où l'agriculture durable et la préservation de la biodiversité sont devenues des enjeux majeurs, la digitalisation des outils de diagnostic biologique représente une nécessité stratégique. Le secteur agricole, particulièrement au Maroc, fait face à des défis croissants liés à la détection rapide des maladies des cultures et du bétail, à la perte de rendement due à des diagnostics tardifs et à la dépendance aux experts spécialisés. Les technologies numériques, notamment l'intelligence artificielle et les systèmes de recherche d'information, offrent aujourd'hui la possibilité de démocratiser l'accès à des diagnostics précis et rapides directement sur le terrain.

Le projet intitulé **SuperEYE** s'inscrit pleinement dans cette dynamique. L'objectif principal est de développer une application mobile intelligente qui permet aux utilisateurs (agriculteurs, éleveurs et citoyens) d'identifier instantanément les espèces végétales ou animales, de détecter d'éventuelles maladies ou anomalies, et d'enrichir ce diagnostic grâce à un moteur de recherche intelligent intégré capable de retrouver et proposer des cas similaires issus de ressources expertes.

La solution propose une plateforme mobile simple et accessible, offrant aux utilisateurs la possibilité de scanner une plante ou un animal, d'obtenir un diagnostic fiable, de visualiser un modèle 3D interactif de l'espèce, de générer un rapport PDF détaillé et personnalisé, et de bénéficier d'informations complémentaires issues d'une base de fiches techniques spécialisées. Un espace administrateur dédié permet quant à lui une gestion efficace des utilisateurs, des contenus et des statistiques d'utilisation.

Ce projet vise ainsi à proposer une solution intuitive, complète et innovante, répondant aux besoins actuels du secteur agricole en connectant efficacement les observations terrain à des connaissances expertes, dans un environnement sécurisé, éducatif et contributif à une agriculture plus résiliente.

2-Motivation et objectifs:

La motivation principale derrière le développement de l'application mobile **SuperEYE** réside dans notre volonté de contribuer concrètement à la modernisation et à la résilience du secteur agricole marocain, un pilier fondamental de l'économie nationale.

En tant qu'étudiants en Génie Informatique à l'ENSA Marrakech, nous avons été particulièrement sensibilisés aux défis auxquels font face les agriculteurs et éleveurs au quotidien : diagnostics tardifs des maladies des cultures ou du bétail, pertes de rendement importantes, accès limité aux experts agronomes ou vétérinaires, et manque d'outils accessibles pour une prise de décision rapide sur le terrain. Ces problématiques entraînent non seulement des conséquences économiques directes pour les exploitants, mais aussi un impact négatif sur la sécurité alimentaire et la préservation de la biodiversité.

Ce projet nous a offert l'opportunité unique d'appliquer nos compétences en intelligence artificielle et en systèmes de recherche d'information pour concevoir une solution mobile pratique et innovante. En intégrant un **service de détection IA** capable d'analyser les images capturées sur le terrain, un **service SRI** qui enrichit automatiquement le diagnostic avec des fiches techniques expertes, et un **service de gestion central (SuperEYE)** orchestrant l'ensemble des interactions utilisateurs, nous avons cherché à créer un outil qui démocratise l'accès à des connaissances spécialisées, même dans les zones rurales les plus isolées.

Au-delà de l'aspect technique, notre motivation est profondément ancrée dans une dimension sociétale et environnementale : favoriser une agriculture plus durable en permettant une intervention précoce face aux pathologies, réduire l'usage excessif de produits phytosanitaires grâce à des diagnostics précis, et encourager la participation citoyenne à la surveillance biologique. Nous croyons fermement que des outils numériques comme SuperEYE peuvent jouer un rôle clé dans la transition vers une agriculture intelligente et résiliente, tout en valorisant le savoir-faire local et les ressources documentaires marocaines (ONSSA, FAO, etc.).

Enfin, ce projet représente pour nous une expérience enrichissante sur le plan personnel et professionnel : il nous a permis de collaborer en équipe, de maîtriser des concepts avancés d'IA et de recherche d'information, et de concevoir une application complète du début à la fin, avec une réelle valeur d'usage pour la société.

II-Analyse et conception

1-Analyse des besoins

1-1)-Spécification des besoins fonctionnels :

Pour les utilisateurs (clients) :

1. Création de compte et connexion :

- Le client peut créer un compte personnel en fournissant les informations nécessaires (nom, prénom, email, mot de passe...).
 - Le client peut se connecter à son compte via son email et mot de passe.
 - En cas d'identifiants incorrects, un message d'erreur est affiché.
 - Une fois connecté, le client accède à une interface utilisateur personnalisée.
2. Consultation des scans et posts des autres utilisateurs :
- Sur la page d'accueil, tous les scans et posts partagés par les autres utilisateurs sont affichés avec leurs informations principales (image scannée, type de plante, maladie détectée, réactions, etc.).
 - Le client peut cliquer sur un scan ou post pour consulter ses détails complets :
 - Image scannée
 - Diagnostic (type de plante, maladies associées)
 - Réponse enrichie (informations extraites des ressources expertes)
 - Réactions et commentaires des autres utilisateurs
3. Partage de scans et réactions :
- Le client peut décider de partager un scan qu'il a effectué, en le rendant visible pour les autres utilisateurs.
 - Le client peut ajouter des réactions (like, dislike, commentaires) sur les scans ou posts partagés par les autres.
 - Les scans partagés contribuent à enrichir la communauté, en permettant des discussions et des échanges sur les diagnostics.
4. Scan d'une plante :
- Le client peut scanner une plante via la caméra de son smartphone.
 - Le scan est transmis automatiquement au backend pour traitement.
 - Le backend utilise un service IA composé de deux modèles :
 - Le premier pour détecter le type de plante.
 - Le second pour identifier les maladies associées.
 - Les résultats (type de plante et maladies) sont ensuite transmis à un service de recherche qui interroge une base de données de fichiers PDF structurés contenant des informations sur les plantes et leurs maladies.
 - Les informations pertinentes extraites des ressources expertes sont envoyées à un LLM pour générer une réponse complète et contextualisée, affichée au client.
5. Accès au service de recherche :
- Le client peut accéder à un espace de recherche dédié pour effectuer des recherches manuelles sur des plantes ou des maladies spécifiques.
 - Le client peut uploader des fichiers ou documents (PDF, images, etc.) pour enrichir la base de données, rendant l'application décentralisée et communautaire (les contributions des utilisateurs développent et améliorent la plateforme).
6. Génération de rapports et historique :
- Une fois le scan traité, un rapport PDF personnalisé est généré et téléchargeable, contenant l'image scannée, le diagnostic, les maladies détectées et les informations extraites des ressources expertes.
 - Le client peut accéder à une section "Mon historique" pour consulter tous ses scans précédents, avec leurs détails (image, diagnostic, réponse enrichie).
7. Gestion du profil utilisateur :
- Le client peut accéder à la section "Mon profil".
 - Il peut y :
 - Consulter ses informations personnelles (nom, email, téléphone...).
 - Modifier son mot de passe.
 - Mettre à jour ses coordonnées.

Pour les chercheurs (utilisateurs normaux avec droits étendus) :

1. Création de compte et connexion :
 - Le chercheur peut créer un compte en fournissant ses informations (nom, prénom, email, mot de passe, affiliation si applicable).
 - Il peut se connecter à son espace via email et mot de passe.
 - Une fois connecté, il accède à un espace dédié avec fonctionnalités avancées pour la recherche.
2. Upload de fichiers et documents :
 - Le chercheur peut uploader des fichiers (PDF, documents, images) pour enrichir la base de données des ressources expertes.
 - Les uploads sont soumis à validation (automatique ou par admin) avant intégration, contribuant à développer l'application de manière décentralisée.
3. Demandes et recherches dans le service de recherche :
 - Le chercheur peut effectuer des demandes manuelles dans le service de recherche, en interrogeant la base de données sur des plantes spécifiques, des maladies ou des thèmes associés.
 - Le service fonctionne comme un chatbot interactif, permettant des requêtes conversationnelles (ex. : "Quelles sont les maladies courantes des tomates au Maroc ?") pour obtenir des réponses enrichies des ressources expertes.
 - Les résultats incluent des extraits pertinents des fichiers PDF, avec possibilités de raffinement des recherches (filtres par région, gravité, etc.).
4. Accès à l'historique et aux contributions communautaires :
 - Le chercheur peut consulter son historique de recherches et uploads.
 - Il peut interagir avec les scans et posts des autres utilisateurs (réactions, commentaires), et exporter des données pour des analyses externes.
5. Gestion du profil chercheur :
 - Le chercheur peut accéder à son profil pour consulter ou modifier ses informations.
 - Il peut suivre les contributions validées et leur impact sur la communauté (ex. : nombre de fois où un upload a été utilisé dans des diagnostics).

Pour l'administrateur :

1. Création de compte et connexion :
 - L'administrateur peut se connecter à son espace dédié via des identifiants sécurisés.
 - Une fois connecté, il accède à un tableau de bord de gestion.
2. Gestion des scans et posts :
 - L'administrateur peut consulter tous les scans et posts partagés par les utilisateurs.
 - Pour chaque scan ou post, il peut :
 - Voir les détails (image, diagnostic, réactions).
 - Valider ou rejeter les contributions sensibles (ex. : uploads de fichiers).
 - Supprimer du contenu inapproprié ou non pertinent.
3. Gestion des utilisateurs :
 - L'administrateur peut consulter la liste des utilisateurs inscrits.
 - Il peut :
 - Modifier les informations d'un utilisateur.
 - Désactiver ou supprimer un compte en cas de besoin.
4. Gestion des ressources expertes :
 - L'administrateur peut accéder à la base de données des fichiers PDF structurés.
 - Il peut :
 - Ajouter de nouvelles fiches techniques sur les plantes et maladies.
 - Valider les uploads des utilisateurs (y compris chercheurs) pour enrichir la base.
 - Supprimer ou modifier des ressources obsolètes.
5. Statistiques et suivi :
 - L'administrateur dispose d'un tableau de bord statistique avec :

- Nombre total de scans effectués.
- Nombre de posts et réactions.
- Maladies les plus détectées et tendances régionales.
- Nombre d'uploads et enrichissements communautaires.

1-1)-Spécification des besoins non fonctionnels :

- Sécurité :
 - Le système doit assurer une authentification sécurisée des utilisateurs à l'aide de JWT (JSON Web Tokens) pour la gestion des sessions et des accès.
 - Les mots de passe des utilisateurs seront stockés de manière sécurisée en base de données à l'aide d'un algorithme de hachage (par exemple : bcrypt).
- Compatibilité :
 - L'interface sera conçue de manière à s'adapter aux différentes tailles d'écran sans perte d'ergonomie (responsive design).
 - Le système sera compatible avec les appareils mobiles modernes (iOS, Android).
- Utilisabilité :
 - L'interface utilisateur sera conçue de manière intuitive et ergonomique, avec une navigation claire entre les différentes fonctionnalités.
 - L'expérience utilisateur sera optimisée pour minimiser le nombre de clics nécessaires et faciliter la prise en main, même pour un utilisateur non technique.

2-Conception

2-1)-Diagramme de cas d'utilisation :

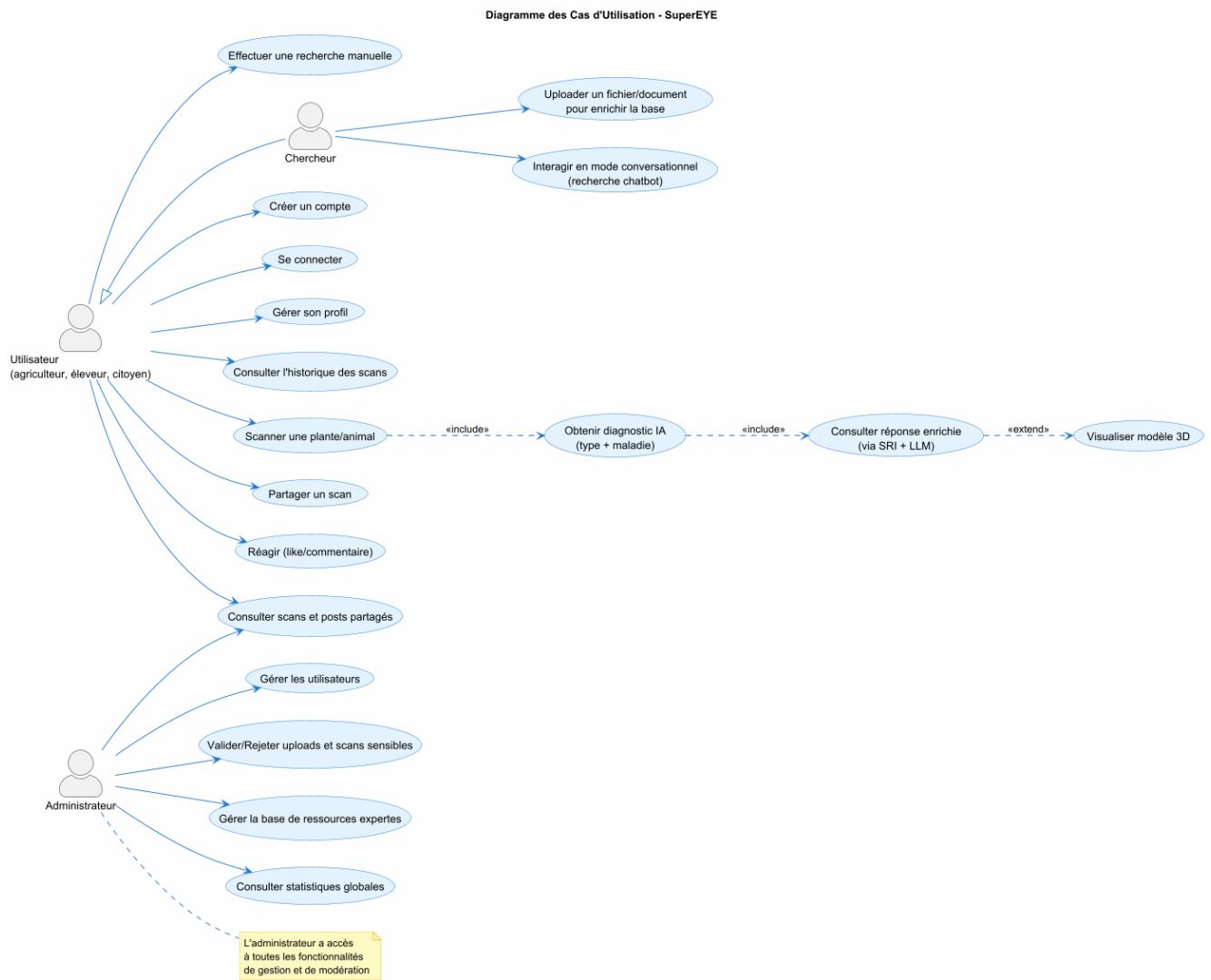


Figure 1:Diagramme de cas d'utilisation

2-2)-Diagramme de classe :

Diagramme de Classes - Modèle de Données SuperEYE (Backend Spring Boot - JPA/Hibernate)

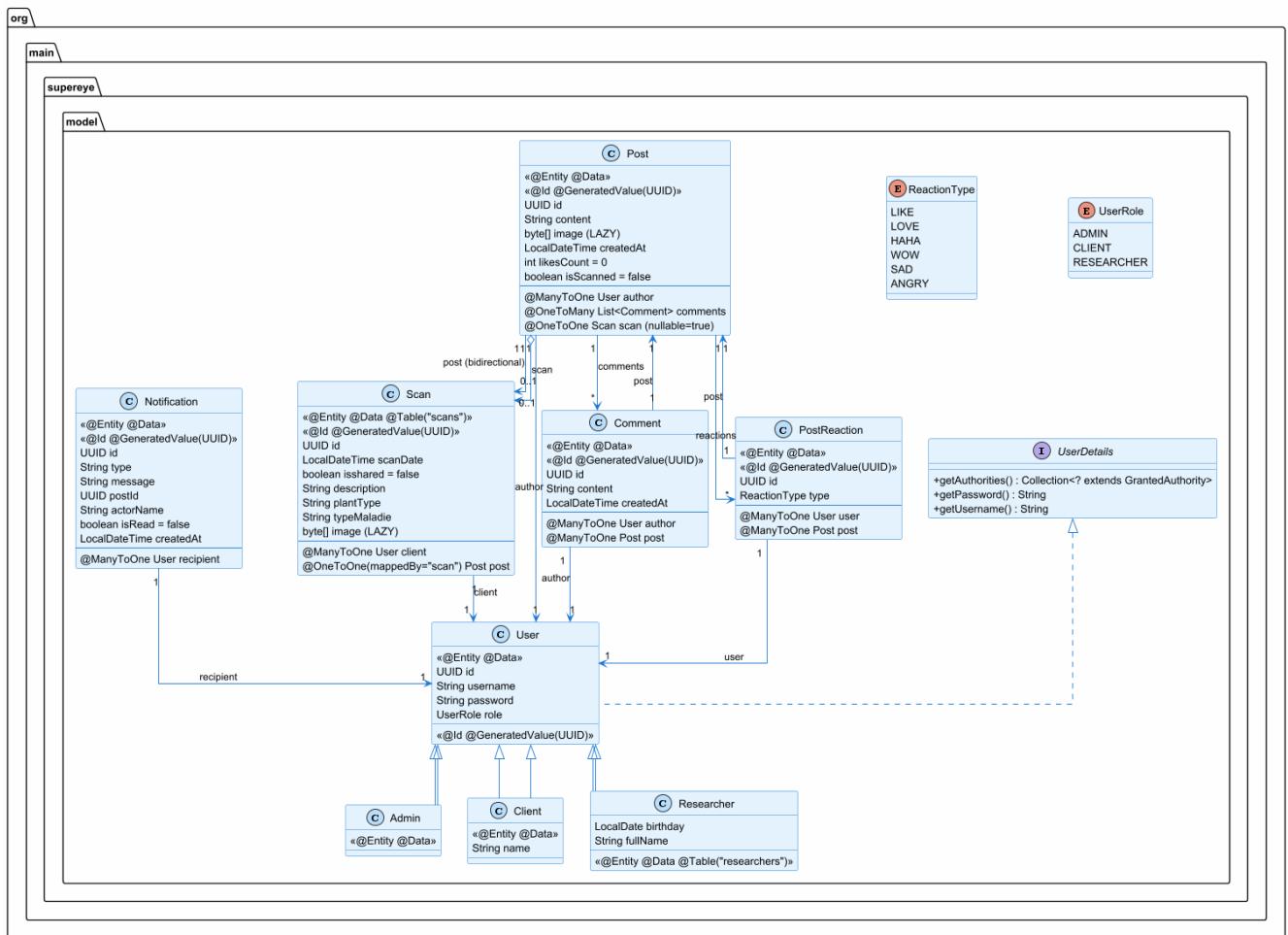


Figure 2:Diagramme de classe

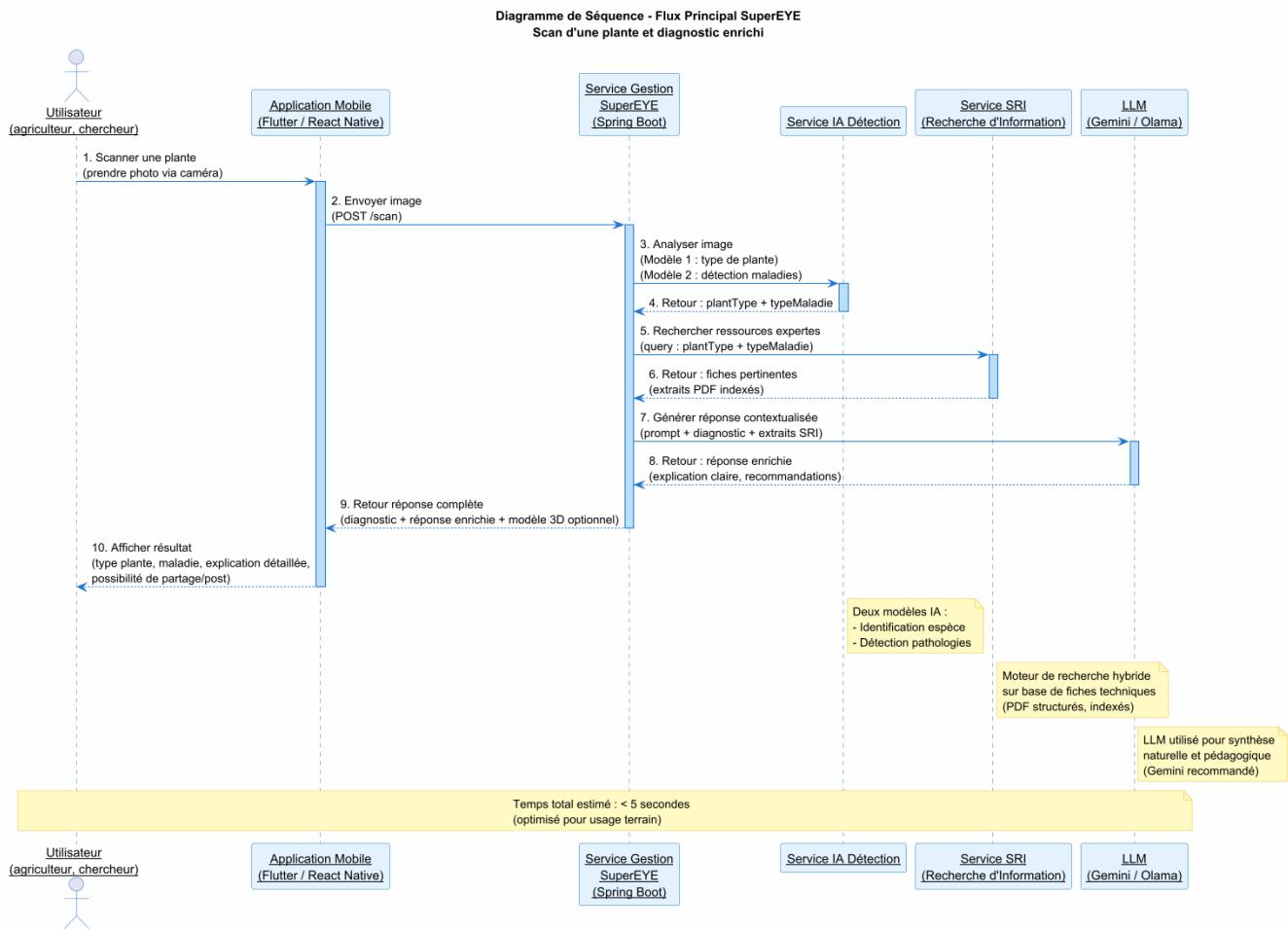


Figure30 :Diagramme de séquence

3-Planning du Projet

Diagramme de GANTT :

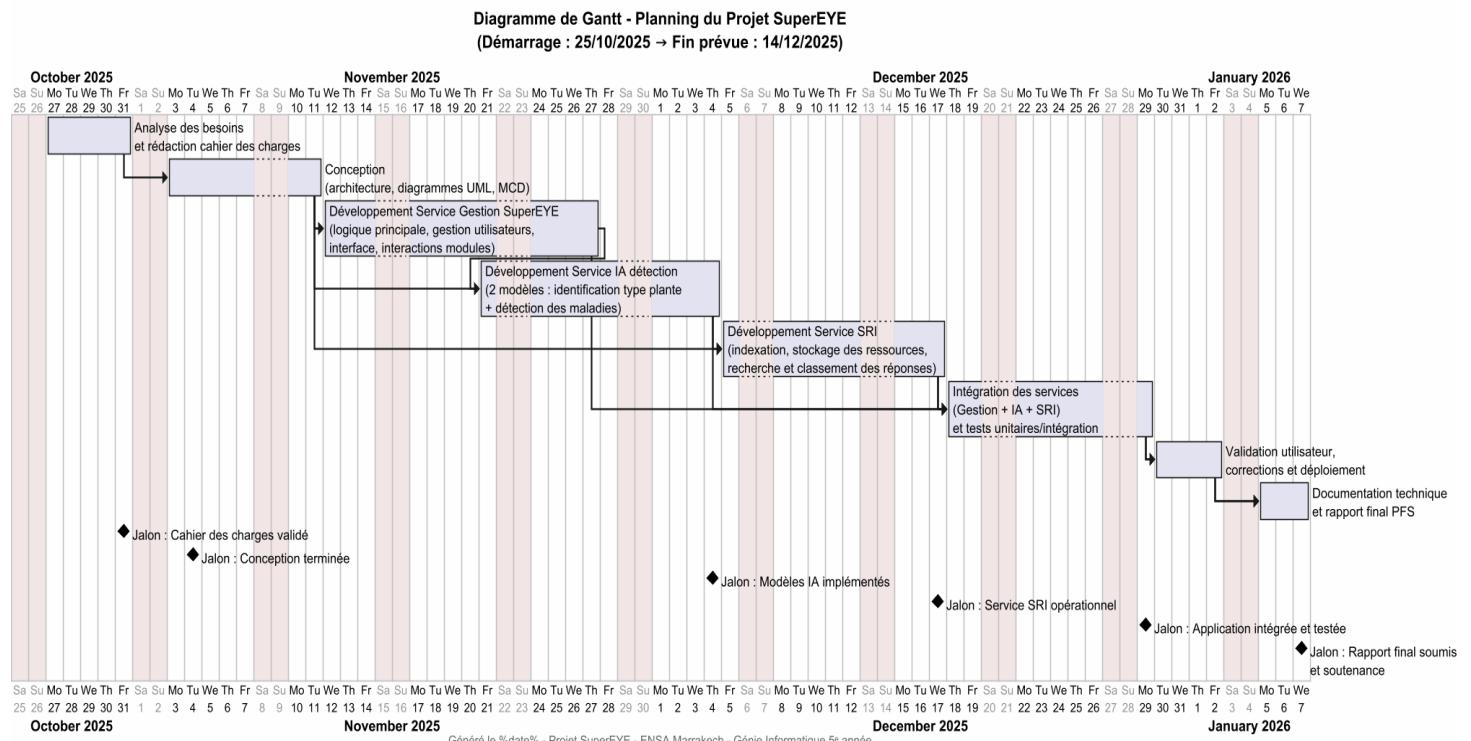


Figure 3:Diagramme de Gantt

III- Outils et Technologies utilisés

Spring Boot:



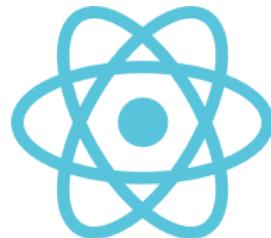
Backend principal, gérant la logique métier, les API REST sécurisées (JWT) et l'orchestration des services.

Flutter:



Framework multiplateforme pour le développement de l'application mobile native (interface client : scan, communauté, historique).

React.js:



Bibliothèque JavaScript utilisée pour **développer des interfaces utilisateurs interactives et réactives**.

Angular.js:



Framework frontend puissant basé sur TypeScript, utilisé pour développer l'interface web d'administration. Il permet la création d'une application monopage (SPA) dynamique, réactive et riche en fonctionnalités pour la gestion des utilisateurs, la modération des contenus et les statistiques.

Docker:



Conteneurisation de l'application pour un déploiement reproduit, isolé et portable.

POSTGRESQL:



Base de données relationnelle pour la persistance des données

Elasticsearch :



Moteur de recherche et d'indexation rapide et scalable, utilisé pour la recherche textuelle et vectorielle dans de grandes bases de documents.

Kibana :



Interface de visualisation interactive des données, utilisée pour explorer et analyser les résultats stockés dans Elasticsearch.

API REST:



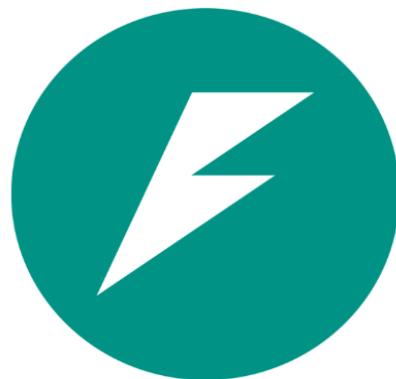
Toutes les communications entre les frontends (Flutter et Angular) et le backend se font via des API RESTful sécurisées (HTTPS + JWT), garantissant un échange de données fiable, standardisé et protégé.

JWT (JSON Web Token)



Intégré pour gérer l'authentification et l'autorisation, avec manipulation des tokens JWT (JSON Web Tokens) pour une sécurité stateless et robuste.

FASTAPI:



Framework Python utilisé pour **développer des API rapides, sécurisées et performantes**.

Flask :



Framework Python léger utilisé pour **développer rapidement des applications web et APIs simples**.

Pytorch:



Framework Python léger utilisé pour **développer rapidement des applications web et APIs simples**.

IntelliJ IDEA :



IntelliJ IDEA est un environnement de développement intégré (IDE) conçu par JetBrains, utilisé dans ce projet pour le développement de la partie backend basée sur Spring Boot.

VS CODE :



Visual Studio Code est un éditeur de code source développé par **Microsoft**, utilisé dans ce projet pour le développement de la partie **frontend** de l'application (React, Angular et Flutter).

Postman:



Outil utilisé pour **tester, déboguer et documenter les API** de manière rapide et efficace.

IV-Réalisation

1-Architecture de projet:

L'application SuperEYE repose sur une architecture modulaire et orientée services, conçue pour garantir une séparation claire des responsabilités, une maintenabilité optimale et une évolutivité future. Le système est structuré autour de trois services principaux qui collaborent pour offrir une expérience fluide et intelligente à l'utilisateur.

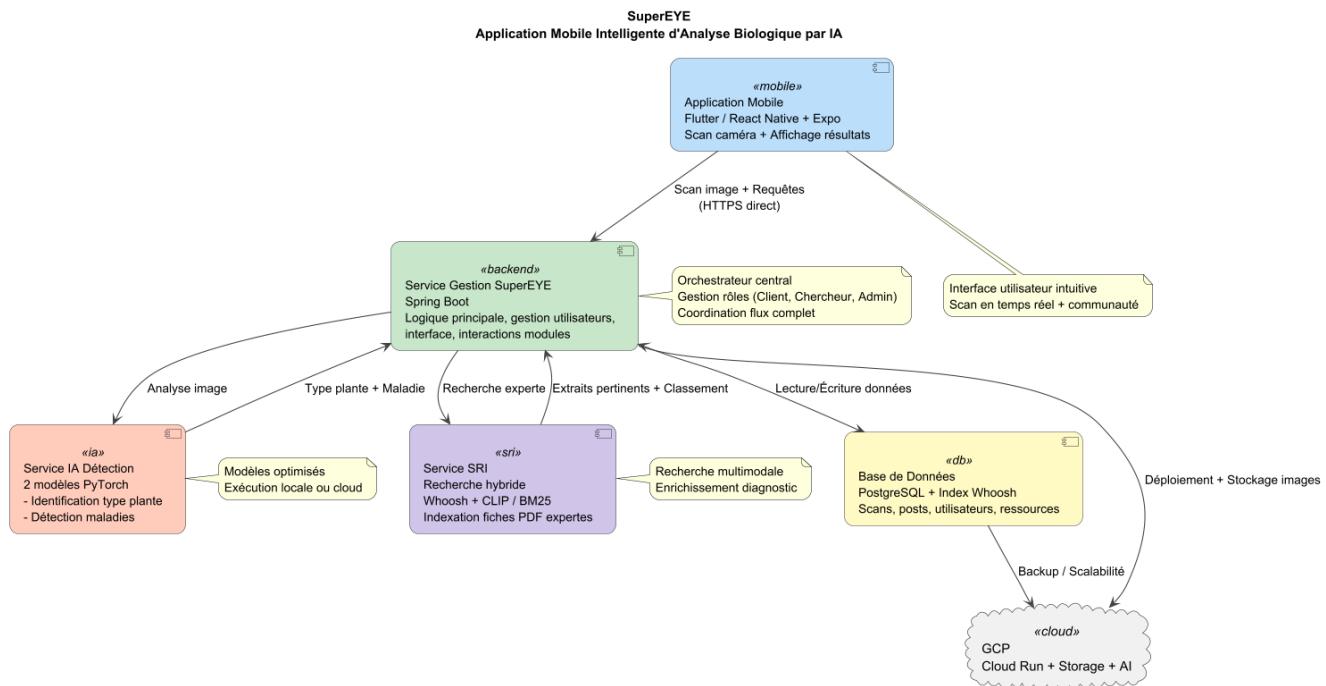


Figure 4: Architecture du projet

1-1)- Service Gestion SuperEYE (Service Application Principal)

Le **Service Gestion SuperEYE** constitue le cœur central et l'orchestre principal de l'application. Contrairement à une architecture strictement séparée, ce service intègre à la fois le backend et les interfaces frontend (mobile et web).

Le **backend** (développé avec **Spring Boot**) gère la logique métier principale, l'authentification et la gestion des utilisateurs (rôles : Client, Chercheur, Administrateur), la persistance des données (scans, posts, commentaires, réactions, notifications), ainsi que la coordination des appels vers les autres services spécialisés (Service IA Détection et Service SRI). Il expose des **API RESTful sécurisées** (HTTPS + JWT) qui servent de point d'entrée unique pour tous les frontends.

Les **frontends** sont les suivants :

- **Interface mobile (utilisateurs clients)** : développée avec **Flutter** pour une expérience native fluide sur Android et iOS.
- **Interface web administrateur** : développée avec **Angular** pour un panel d'administration riche, réactif et complet.

Les deux frontends (Flutter et Angular) communiquent **exclusivement** avec le backend Spring Boot via des **API REST sécurisées** (HTTPS + JWT pour l'authentification). Toutes les requêtes (scan d'image, partage de post, consultation d'historique, recherche manuelle, gestion utilisateurs, validation de contenus, etc.) transitent par ce point d'entrée unique.

Rôles principaux du Service Gestion SuperEYE :

- Orchestration complète du flux utilisateur : réception des données des frontends, délégation au Service IA pour l'analyse, appel au Service SRI pour l'enrichissement contextuel, assemblage de la réponse finale, et renvoi aux frontends.
- Gestion centralisée de la sécurité, des sessions et des autorisations.

- Stockage et récupération des données via la base PostgreSQL (scans, posts, utilisateurs, ressources uploadées).
- Interface unique d'exposition pour les frontends : toutes les interactions passent par des endpoints REST bien définis (ex. : /api/scan, /api/search, /api/posts, /api/admin/users).
- Point de contrôle pour les fonctionnalités communautaires (partage, réactions, commentaires) et administratives (validation de contenus, statistiques).

Grâce à cette centralisation, le Service Gestion SuperEYE garantit une cohérence globale, une maintenabilité simplifiée et une évolutivité maîtrisée, tout en offrant une expérience utilisateur fluide et sécurisée sur mobile (Flutter) et web (Angular).

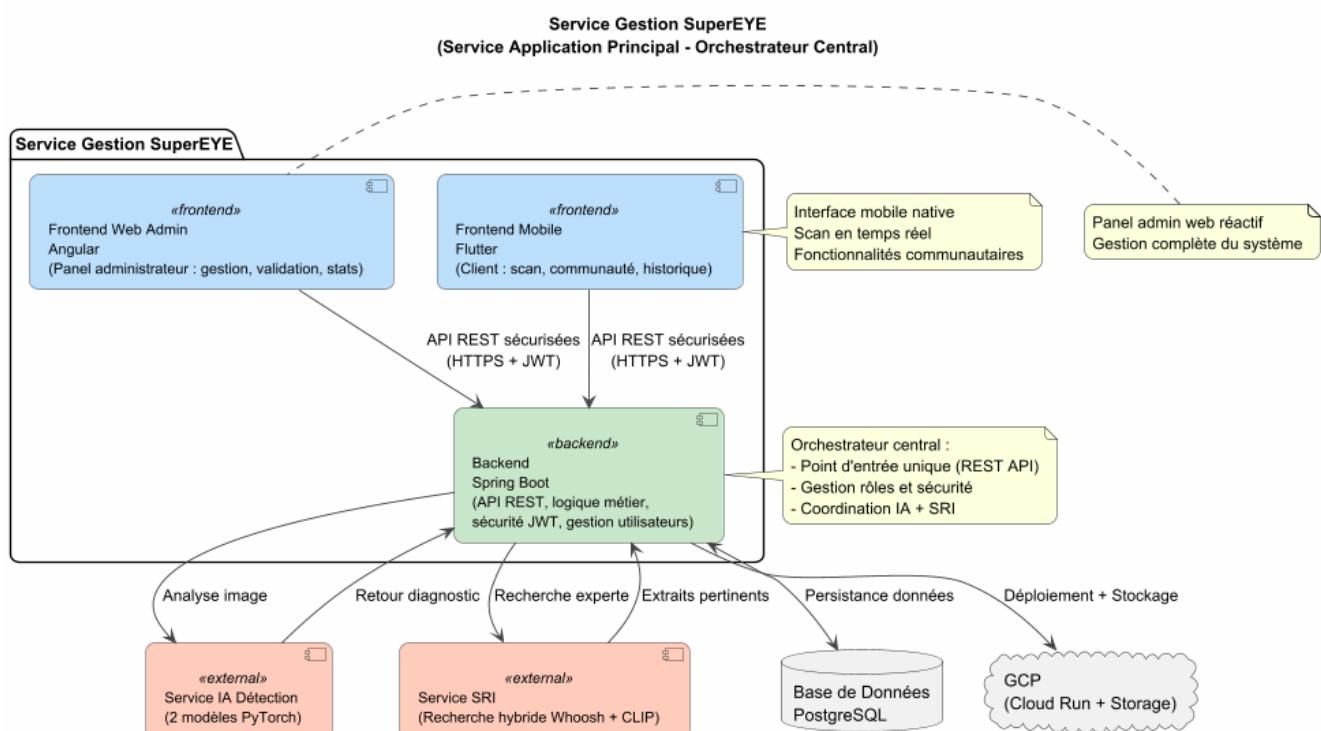


Figure 5: Architecture du backend

1-2)- Service IA Detection

Le Service IA Détection est le module spécialisé responsable de l'analyse intelligente des images capturées par l'utilisateur. Il constitue le premier maillon essentiel du processus de diagnostic automatique.

Ce service est indépendant du Service Gestion SuperEYE et est invoqué uniquement lorsque une image nécessite une analyse. Il est implémenté à l'aide de **PyTorch** et repose sur **deux modèles d'intelligence artificielle distincts et complémentaires** :

- **Modèle 1** : Identification du type de plante (ou animal). Ce modèle classe l'espèce observée parmi une base d'entités biologiques connues.
- **Modèle 2** : Détection des maladies ou anomalies. Une fois le type identifié, ce modèle analyse les symptômes visibles (taches, déformations, discolorations, etc.) pour déterminer la ou les pathologies probables.

Rôles principaux du Service IA Détection :

- Réception de l'image brute provenant du Service Gestion SuperEYE (via appel interne ou API dédiée).

- Prétraitement de l'image (redimensionnement, normalisation, augmentation si nécessaire) pour optimiser la précision.
- Exécution séquentielle ou parallèle des deux modèles afin d'obtenir un diagnostic complet (type d'espèce + maladie(s) détectée(s) avec niveaux de confiance).
- Transmission des résultats structurés (plantType, diseaseType, confidence scores) au Service Gestion SuperEYE, qui les relaie ensuite au Service SRI pour enrichissement contextuel.
- Optimisation pour une exécution rapide (inférieure à 3 secondes en moyenne), même sur des images prises en conditions réelles (terrain, éclairage variable).

Ce service est conçu pour être évolutif : de nouveaux modèles peuvent être ajoutés ou remplacés sans impact sur les autres composants. Il représente le socle technique du caractère « intelligent » de SuperEYE, transformant une simple photo en un diagnostic biologique fiable et exploitable.

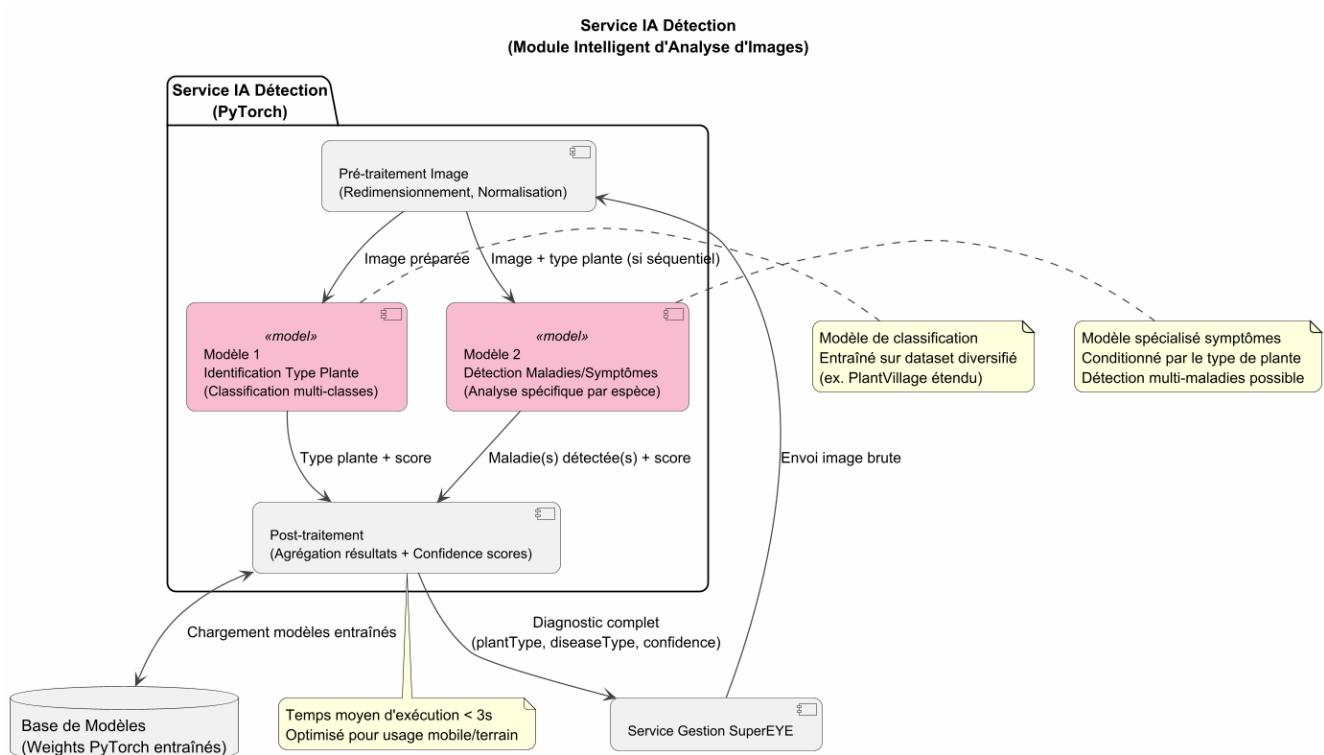


Figure 6: Architecture du frontend

1-3)- Service Système de Recherche d'Informations (SRI)

Le **Service SRI** est le module dédié à la recherche intelligente et à l'enrichissement contextuel des diagnostics. Il agit comme un moteur de recherche spécialisé qui transforme les résultats bruts du Service IA Détection en informations expertes, fiables et pédagogiques.

Ce service est indépendant et est appelé uniquement après obtention du diagnostic IA (type de plante + maladie détectée). Il est implémenté à l'aide de **Whoosh** pour l'indexation textuelle et **CLIP** (ou un modèle similaire) pour la recherche hybride texte-image, avec une approche BM25 pour le classement.

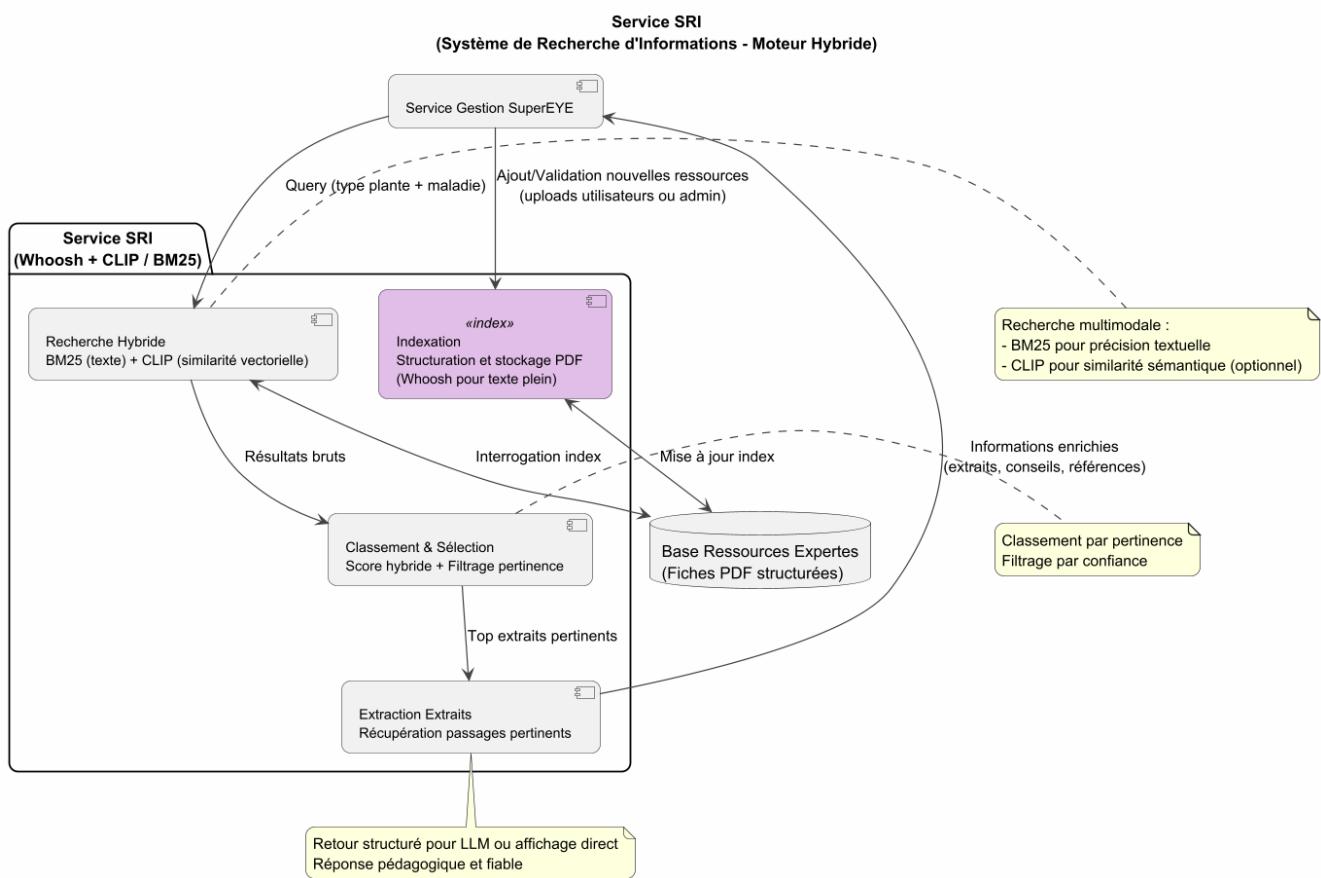
Rôles principaux du Service SRI :

- Indexation structurée et semi-automatique des ressources expertes (fiches techniques au format PDF sur les plantes, maladies, traitements et recommandations agronomiques).
- Recherche multimodale basée sur les résultats du Service IA : interrogation de la base documentaire à partir du type de plante et de la maladie identifiée.

- Classement et sélection des extraits les plus pertinents (textes, descriptions, conseils de traitement) grâce à un score hybride (BM25 pour le texte + similarité vectorielle CLIP si nécessaire).
 - Transmission des informations enrichies au Service Gestion SuperEYE, qui les utilise pour composer la réponse finale affichée à l'utilisateur.
 - Support des recherches manuelles effectuées directement par les utilisateurs (clients ou chercheurs) via l'interface mobile ou web.

Le Service SRI garantit que le diagnostic ne reste pas limité à une simple étiquette (ex. : « tomate – mildiou »), mais soit complété par des explications détaillées, des recommandations de traitement et des références issues de sources expertes validées (ONSSA, FAO, publications agronomiques locales, etc.).

Grâce à son architecture modulaire, ce service peut évoluer facilement : ajout de nouvelles ressources, amélioration du classement, ou extension à d'autres types de recherches (par région, saison, etc.), sans impact sur les autres composants.



2-Présentation des interfaces:

2-1)- Côté chercheur

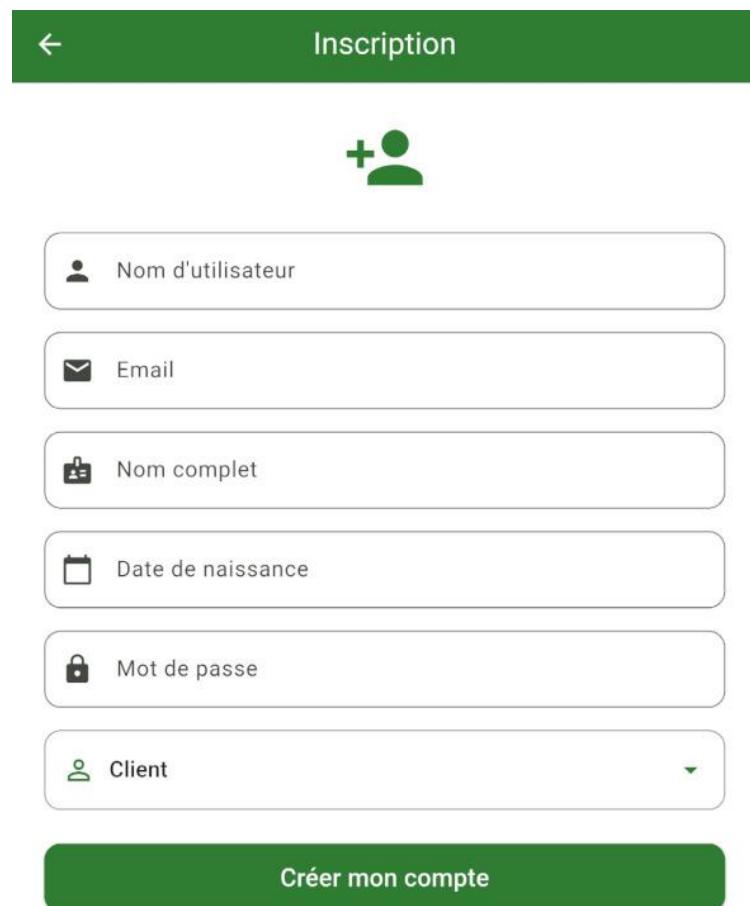
Pour accélérer sa recherche, le chercheur peut télécharger ses documents et rechercher les sujets qui l'intéressent.

The screenshot shows a dark-themed dashboard titled "Research LAB". At the top, it displays "Total Documents: 20" and "Total Chunks: 282". Below this, a section titled "Documents" lists ten flower-related PDF files, each accompanied by a green progress bar indicating the number of chunks: Azalea.pdf (16 Chunks), Begonia.pdf (15 Chunks), Aster.pdf (13 Chunks), Daffodil.pdf (16 Chunks), Chrysanthemum.pdf (13 Chunks), Iris.pdf (15 Chunks), Gladiolus.pdf (13 Chunks), Marigold.pdf (13 Chunks), Carnation.pdf (11 Chunks), and Geranium.pdf (12 Chunks). A "Pansy.pdf" entry is also partially visible.

Voici l'interface de recherche que le chercheur peut utiliser pour effectuer ses recherches.

The screenshot shows the "Research LAB" search interface. At the top, there's a search bar with the placeholder text: "Describe the key characteristics of Rosa (rose plants), including their botanical features, growth habits, environmental requirements, and common uses. Please provide a detailed, structured, and concise way suitable for educational and AI-based plant classification purposes." Below the search bar, a large text area displays the search results for "Rosa (Rose Plants)". It includes sections for "Scientific Name", "Botanical Features", "Growth Habits", "Environmental Requirements", "Common Uses", and "Ornamental", "Commercial", "Ecological" applications. To the right of the main content, a sidebar titled "Sources" lists four PDF files with their names and similarity scores: Rose.pdf (0.91), Gladiolus.pdf (0.76), Chrysanthemum.pdf (0.75), and Carnation.pdf (0.75).

Le client peut s'inscrire ou se connecter grâce à cette interface :



The image shows a registration form titled "Inscription" (Registration) with a back arrow icon. It features a green header bar and a large green "Create account" button at the bottom. The form includes fields for user name, email, full name, date of birth, password, and role (Client). A green plus sign icon is positioned above the first input field.

Inscription

+ 

 Nom d'utilisateur

 Email

 Nom complet

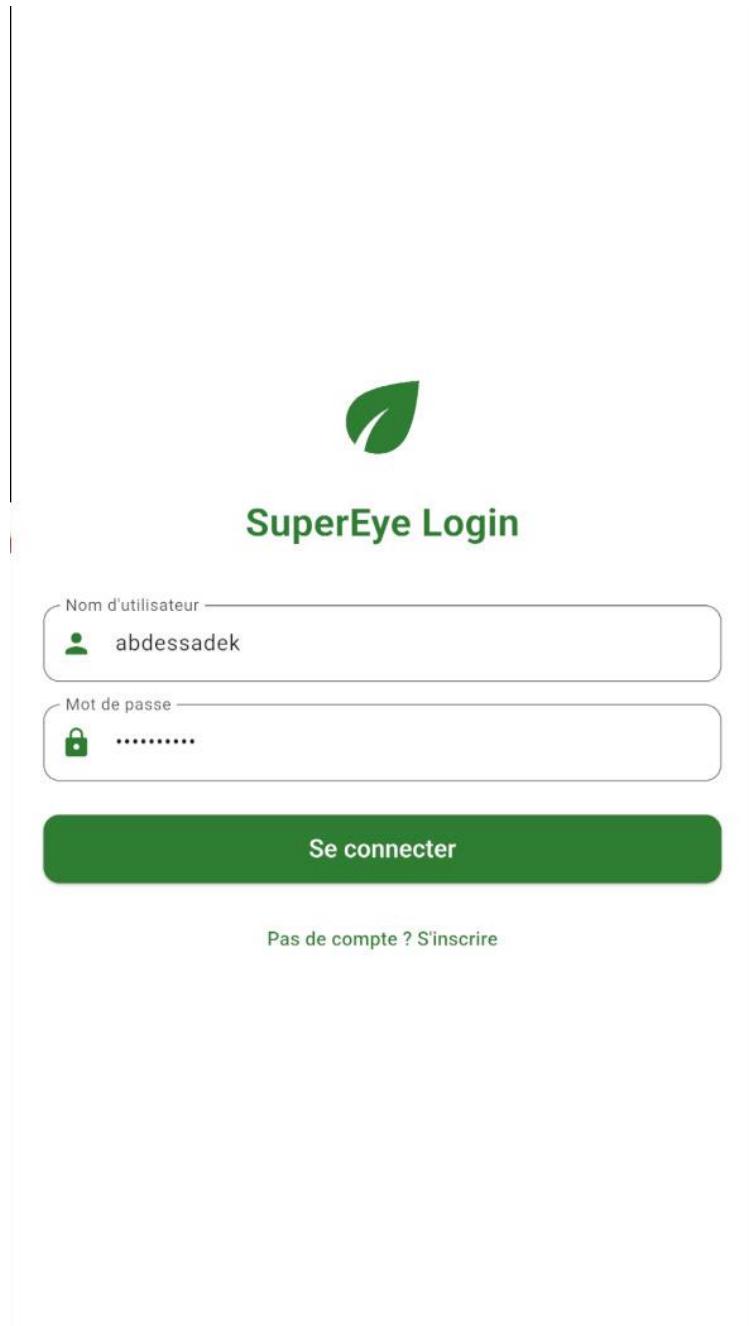
 Date de naissance

 Mot de passe

 Client 

Créer mon compte

le client peut se connecter et s'inscrire grâce à cette interface :



The image shows a mobile-style login screen titled "SuperEye Login". At the top center is a green leaf icon. Below it is the title "SuperEye Login" in a bold, dark green sans-serif font. The form consists of two input fields: "Nom d'utilisateur" (User name) containing "abdessadek" with a user icon, and "Mot de passe" (Password) containing a lock icon and several dots. A large green button labeled "Se connecter" (Connect) is centered below the inputs. At the bottom of the form is a link "Pas de compte ? S'inscrire" (No account? Sign up).

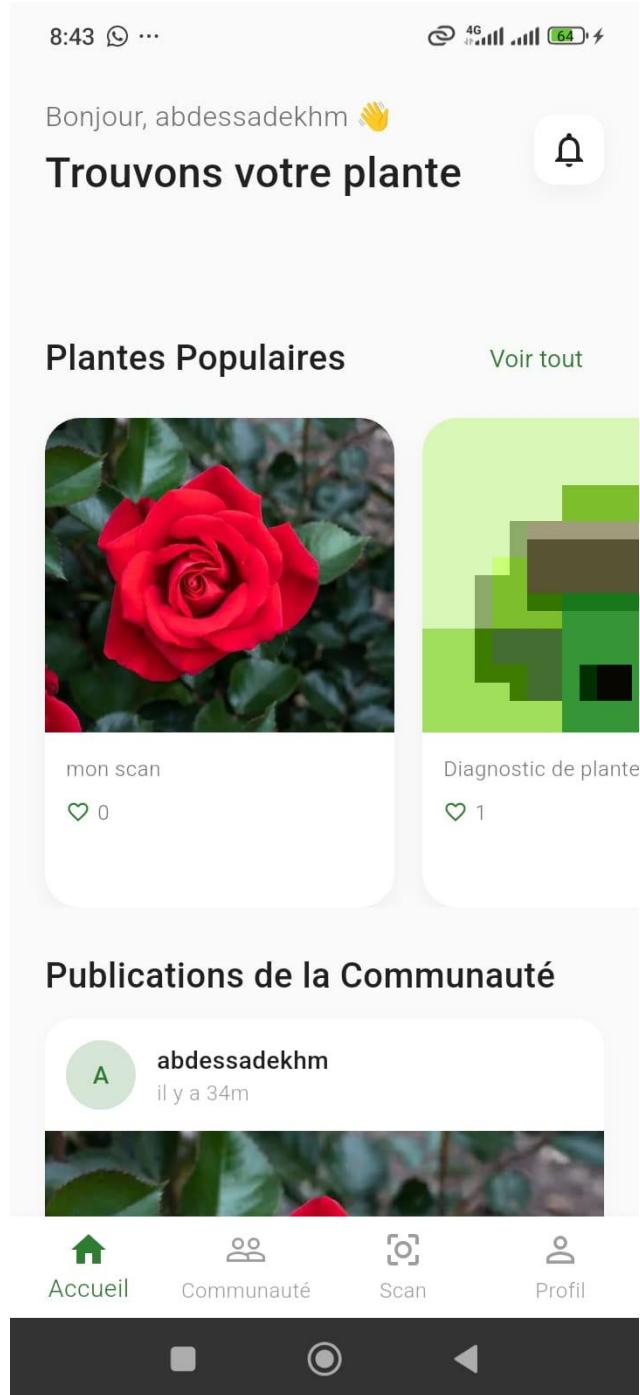
Nom d'utilisateur
abdessadek

Mot de passe
.....

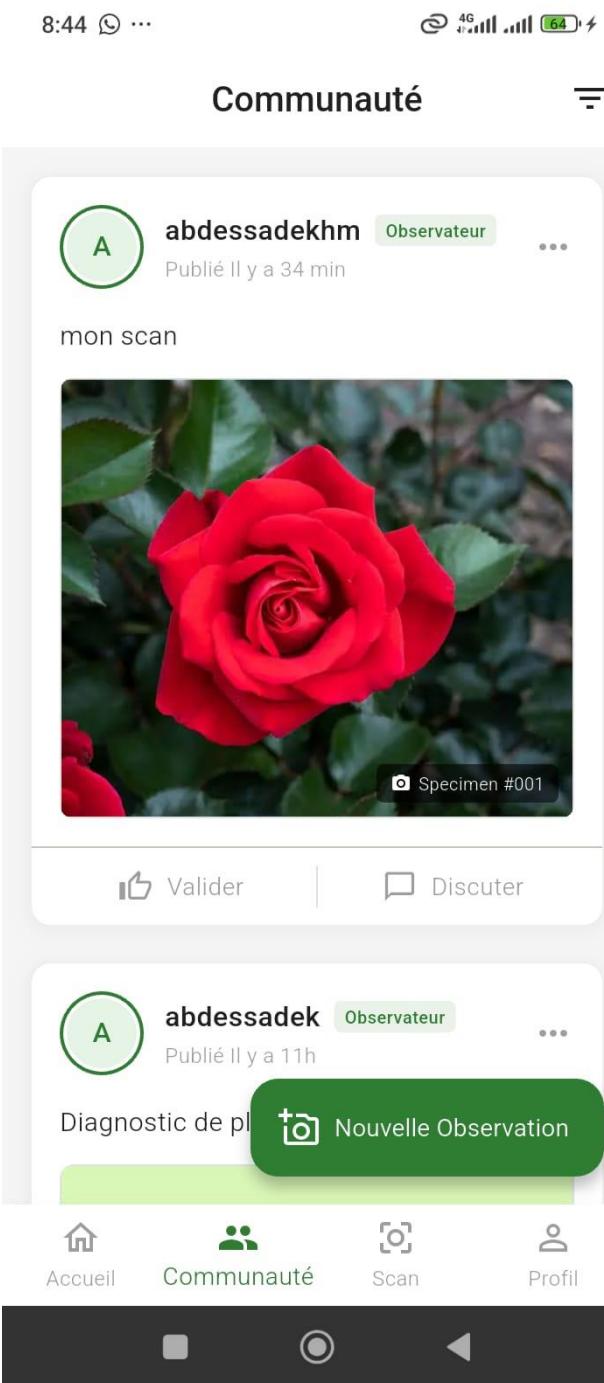
Se connecter

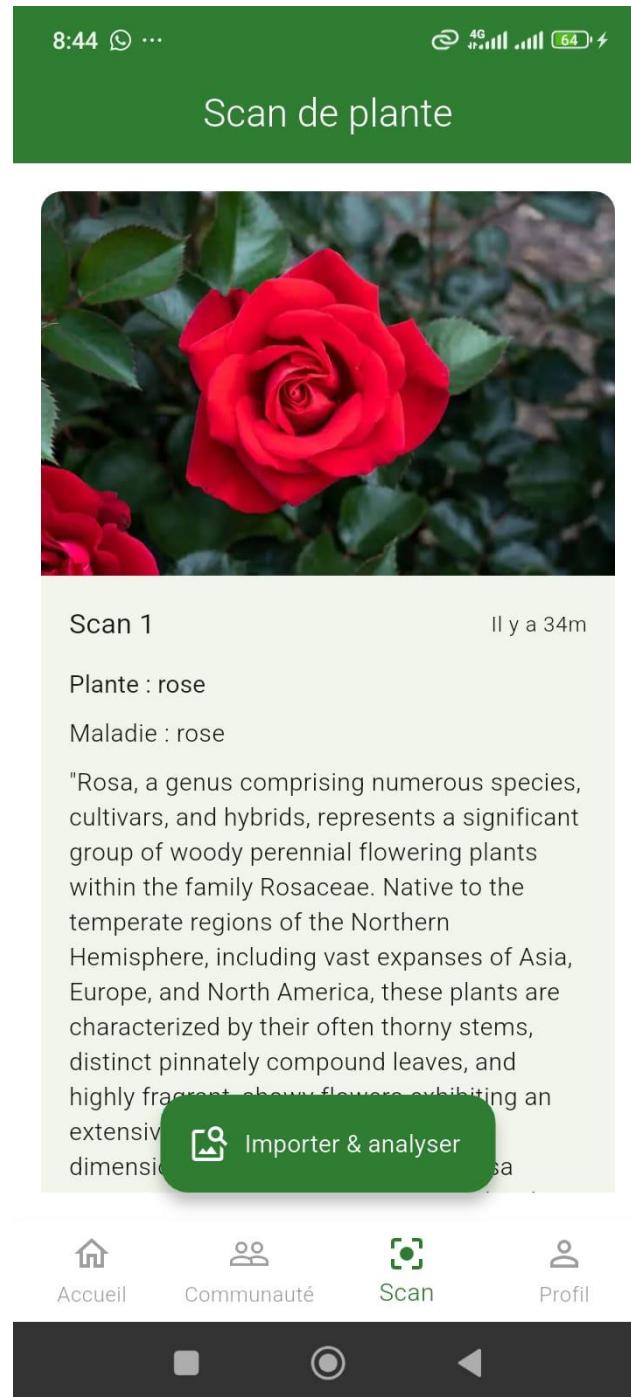
Pas de compte ? S'inscrire

L'interface d'accueil affiche les publications les plus populaires.

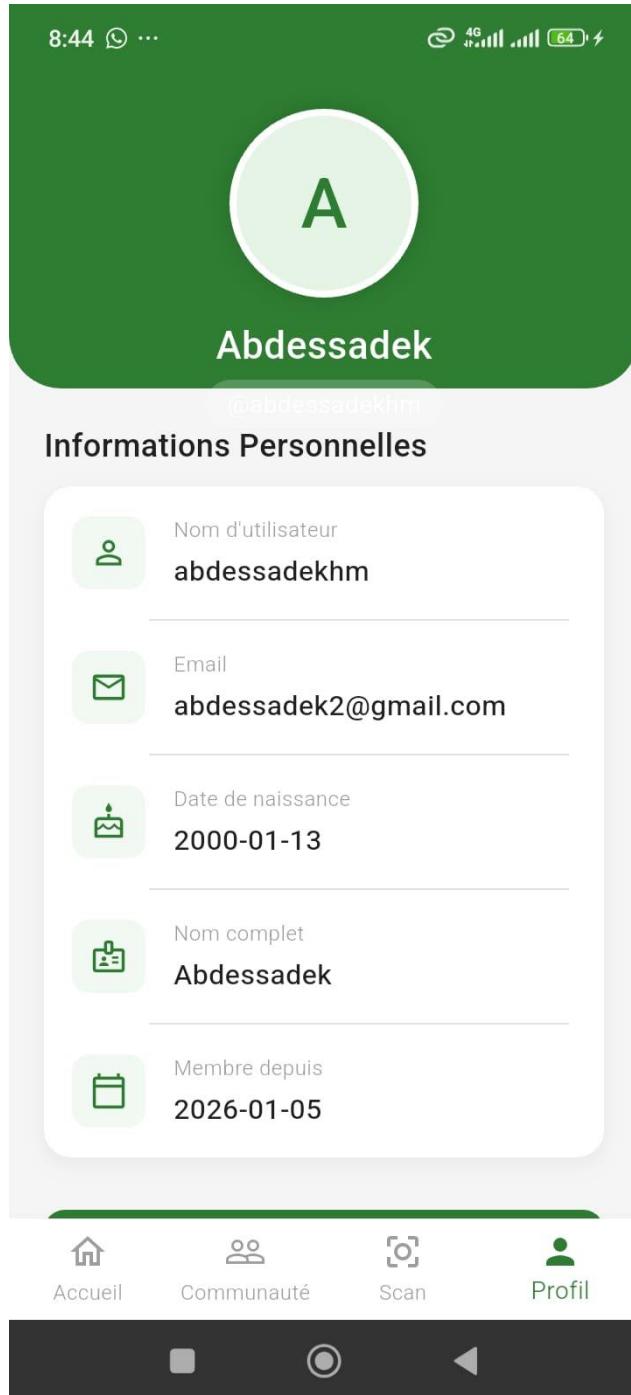


L'interface de communityl affiche les publications.





L'interface de profile.



3- Modules spécifiques de SUPEREYE :

3-1) Module SRI (Système de Recherche d'Informations)

3-1-1) Thématique du Moteur de Recherche

Le projet consiste à développer un moteur de recherche sémantique spécialisé dans le domaine des plantes et de leurs maladies associées, permettant de rechercher des informations à partir d'une base de documents volumineuse et de fournir une réponse naturelle et pertinente à l'utilisateur.

Contrairement aux moteurs de recherche traditionnels, qui se limitent à la correspondance exacte de mots-clés, ce système s'appuie sur des techniques avancées de traitement du langage naturel et de recherche d'information pour comprendre le sens réel des requêtes formulées par l'utilisateur, et ainsi générer des réponses précises et adaptées au besoin exprimé.

Le moteur est capable de retrouver non seulement les plantes et maladies pertinentes, mais également les documents PDF associés, offrant ainsi des informations complètes et détaillées. Cette approche permet de capturer les relations conceptuelles et implicites entre les différentes entités et les documents, garantissant une recherche plus précise, pertinente et enrichissante pour les utilisateurs.

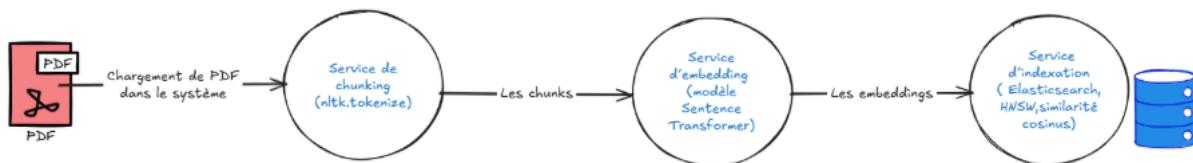
3-1-2) Fonctionnalités principales

Le système développé offre deux fonctionnalités principales :

A) Indexation :

Cette étape comprend plusieurs sous-processus :

- Chunking du texte à l'aide de NLTK,
- Génération d'embeddings via Sentence Transformer (all-mnlp-base-v2)
- Indexation des vecteurs dans Elasticsearch à l'aide de l'algorithme HNSW, permettant une recherche vectorielle rapide et efficace basée sur la proximité des documents

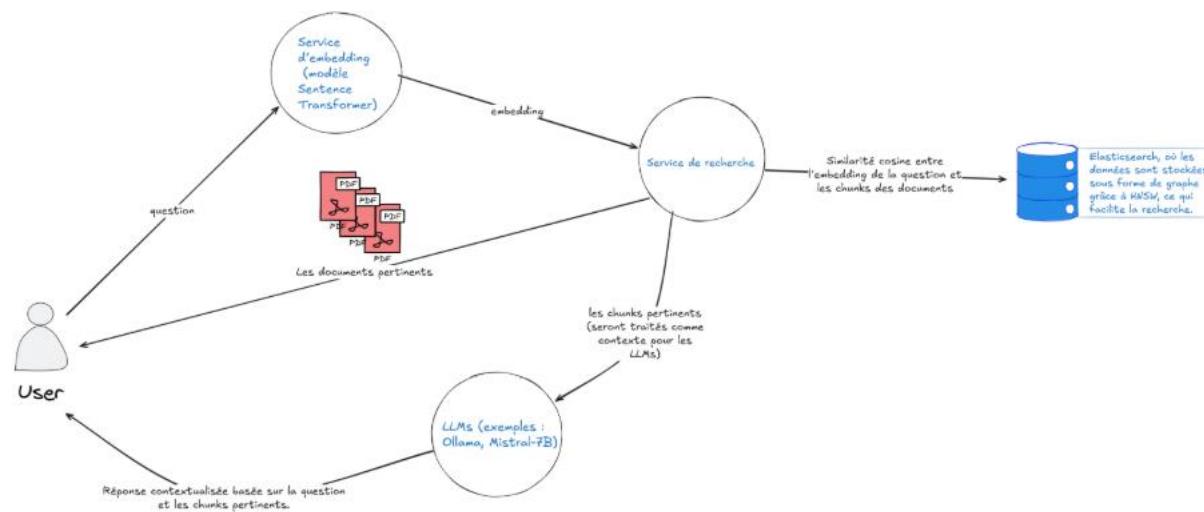


B) Recherche:

Cette fonctionnalité permet de **récupérer les chunks pertinents** d'un ensemble de documents en comparant les **embeddings de la requête utilisateur** avec les **embeddings des documents indexés**.

Les chunks les plus proches (les plus similaires) sont renvoyés au **modèle de langage (LLM)**, qui génère une réponse **contextualisée** à partir de :

- la requête de l'utilisateur,
- les chunks pertinents extraits.



3-1-3) Processus d'indexation et de recherche

3-1-3-1) Indexation : automatique sans prétraitement linguistique avancé

Dans ce système de recherche d'information, l'indexation est réalisée de manière automatique et repose principalement sur des représentations vectorielles sémantiques (embeddings) générées à partir du contenu textuel des documents. Contrairement aux approches classiques de recherche lexicale (basées sur la racinisation, la lemmatisation ou la suppression des stop-words), aucune étape de prétraitement linguistique avancé n'est appliquée. Le sens des textes est directement capturé par le modèle d'embeddings.

L'indexation concerne des documents PDF, qui sont découpés en segments textuels (chunks), transformés en vecteurs numériques, puis stockés dans un index Elasticsearch configuré pour la recherche vectorielle.

A)- Choix d'Indexation

- Type : Indexation automatique
- Moteur d'indexation : Elasticsearch, choisi pour sa robustesse, sa scalabilité et son support natif de la recherche vectorielle.
- Mesure de similarité : similarité cosinus (cosine similarity), adaptée à la comparaison de vecteurs normalisés issus de modèles de type Transformer.
- Algorithme de recherche approximative : HNSW (Hierarchical Navigable Small World), permettant une recherche rapide et efficace dans des espaces vectoriels de grande dimension. Les paramètres HNSW retenus sont :
- m=32:nombre maximal de connexions par nœud, assurant un bon compromis entre précision de la recherche et consommation mémoire.
- ef_construction = 100 : paramètre contrôlant la qualité du graphe lors de la phase de construction de l'index, améliorant la précision des résultats au prix d'un temps de construction légèrement plus élevé.

B)- Méthode d'Indexation

La méthode d'indexation repose sur un processus automatique en trois étapes principales : le découpage des documents, la génération des embeddings et l'indexation vectorielle dans Elasticsearch.

— Découpage des documents (Chunking) : Les documents PDF sont d'abord transformés en texte, puis découpés en segments sémantiquement cohérents (chunks). Cette étape est réalisée par la fonction `chunk_by_sentences_with_smart_tools()`, qui s'appuie sur les éléments suivants :

- `clean_text()` pour le nettoyage du texte;
- `sent_tokenize()` (NLTK) pour la segmentation en phrases;
- `find_sentence_continuation()` pour reconstituer les phrases coupées entre deux pages;
- `get_smart_text_start_improved()` et `get_smart_text_end_improved()` pour éviter les coupures non naturelles.

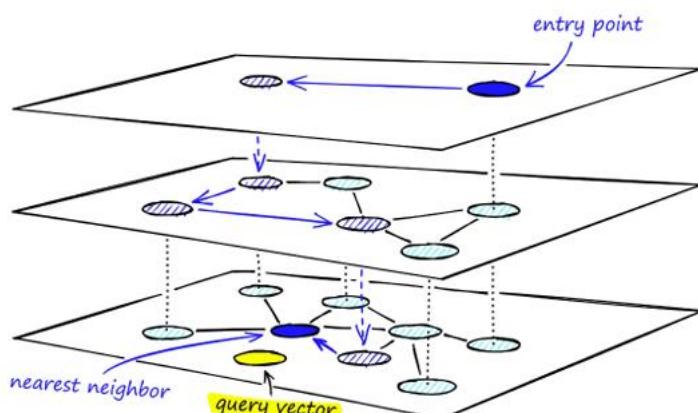
Ce découpage respecte le sens du texte, limite la fragmentation et assure une bonne continuité contextuelle entre les chunks.

— Génération des embeddings : Chaque chunk est ensuite converti en un vecteur numérique dense à l'aide de la fonction `embed()`. Cette fonction utilise le modèle Sentence Transformer (all-mpnet-base-v2), qui encode chaque segment textuel en un vecteur de dimension 768 capturant sa signification sémantique.

— Indexation dans Elasticsearch avec HNSW : Avant l'indexation, l'index Elastic search est créé via la fonction `create_index_if_not_exists()`. Les chunks vectorisés sont ensuite stockés à l'aide de la fonction `index_pdf()`, avec un mapping incluant un champ `dense_vector` configuré pour :

- la similarité cosinus;
- l'algorithme HNSW (`index_options : type = hnsw`).

L'algorithme HNSW (Hierarchical Navigable Small World) utilisé par Elasticsearch permet d'optimiser la recherche de similarité vectorielle en organisant les embeddings sous forme de graphes hiérarchiques. Comme illustré dans l'image ci-dessous, la recherche commence par un point d'entrée situé dans les couches supérieures, peu denses, afin de se rapprocher rapidement de la zone pertinente. Ensuite, la navigation descend progressivement vers des couches plus denses, où l'algorithme explore les voisins les plus proches du vecteur de requête. Cette approche hiérarchique réduit considérablement le nombre de comparaisons nécessaires, tout en maintenant une excellente précision, ce qui rend la recherche beaucoup plus rapide et efficace que les méthodes exhaustives.



3-1-3-2) Stockage

Les segments de texte sont stockés dans Elasticsearch avec :

- un identifiant de document (`doc_id`) et de segment (`chunk_id`),
- les vecteurs dans un champ `dense_vector` utilisant la similarité cosinus,
- l'algorithme HNSW pour organiser les embeddings dans un graphe hiérarchique, permettant une recherche rapide et précise,
- la source du document pour retrouver l'origine des chunks.

Cette organisation assure un stockage structuré et performant pour la recherche vectorielle.

```

    return
body = {
  "mappings": {
    "properties": {
      "doc_id": {"type": "keyword"},
      "chunk_id": {"type": "integer"},
      "text": {"type": "text"},
      "embedding": {
        "type": "dense_vector",
        "dims": 768,
        "index": True,
        "similarity": "cosine",
        "index_options": {
          "type": "hnsw",
          "m": 32,
          "ef_construction": 100
        }
      },
      "source": {"type": "keyword"}
    }
  }
}

```

3-1-3-3) Recherche : modèle vectoriel avec similarité cosinus

Le modèle de recherche d'information adopté est basé sur la similarité vectorielle :

A) Modèle de RI

- Type : Recherche vectorielle sémantique
- Similarité : Cosinus entre vecteurs normalisés
- Encodage : Sentence Transformer (all-mpnet-base-v2)

Chaque requête et chaque segment de document (chunk) sont encodés en vecteurs d'embedding. La recherche se fait en comparant ces vecteurs dans l'espace sémantique. Voici le format de la requête utilisée :

```

body = {
  "knn": {
    "field": "embedding",
    "query_vector": q_vec,
    "k": k,
    "num_candidates": 100
  }
}

```

B) Flux de Recherche

Le processus de recherche dans le système sémantique se déroule selon les étapes suivantes :

1. Récupération de la requête : La question ou la description saisie par l'utilisateur est extraite afin d'être traitée par le système.
2. Encodage sémantique : La requête est convertie en un vecteur d'embedding à l'aide du modèle Sentence Transformer (all-mpnet-base-v2).

3. Calcul de similarité : Le vecteur de la requête est comparé aux embeddings des chunks issus des documents PDF. Dans le graphe HNSW :

- à chaque niveau, le vecteur de la requête est comparé aux 100 chunks les plus proches;
- les k chunks les plus pertinents sont retenus à chaque niveau;

— cette opération est répétée sur l'ensemble des niveaux du HNSW afin d'obtenir une sélection finale des k chunks les plus proches.

4. Retour des documents pertinents : Les k chunks les plus pertinents sont retournés. À partir de ces segments, les documents d'origine correspondants sont identifiés, permettant de présenter à l'utilisateur les documents les plus proches de sa requête

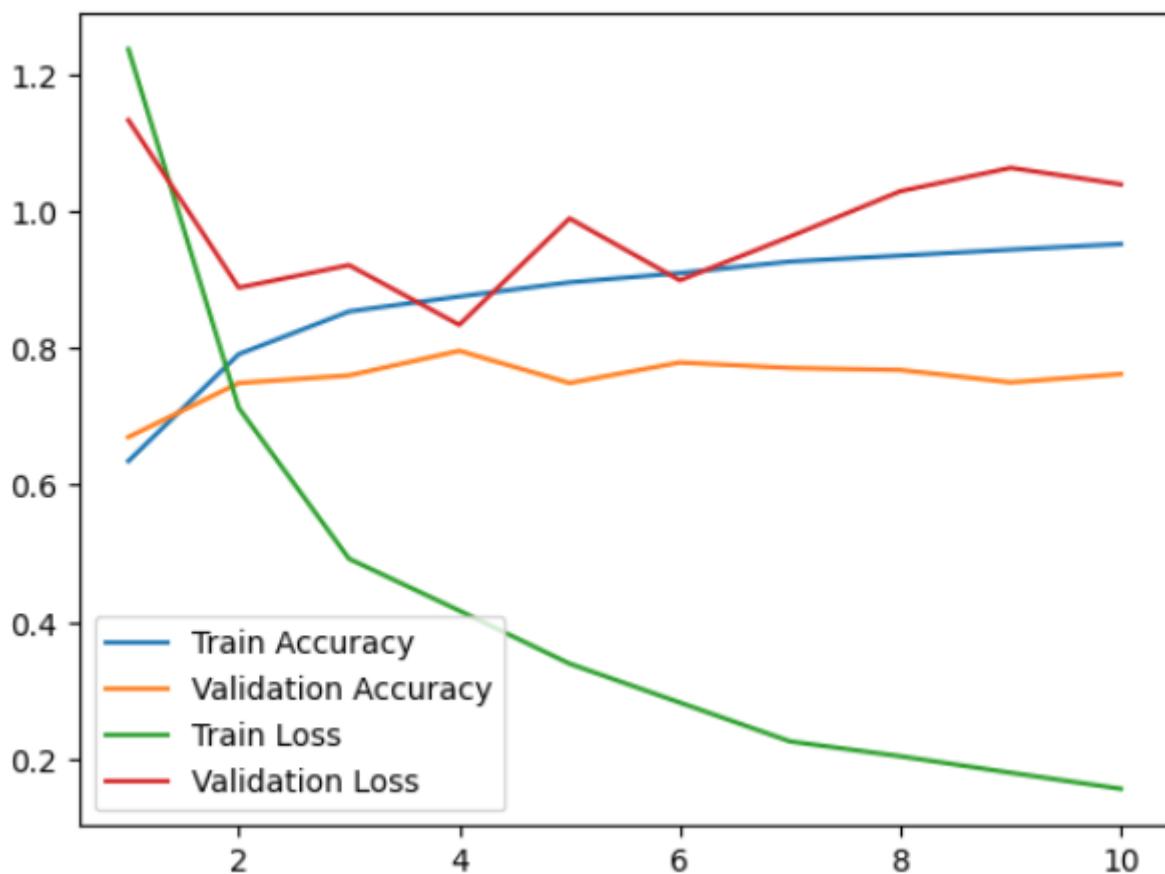
3-2) Modèle de prédictions

3-1-1) Prédiction de type d'image :

A) Modele Efficientnet V1:

Pour ce projet de classification de fleurs, nous avons opté pour l'architecture EfficientNet-v1, pré-entraînée sur la base de données ImageNet. Ce choix se justifie par l'efficacité exceptionnelle de ce modèle en termes de rapport précision/ressources. Contrairement aux architectures classiques qui augmentent arbitrairement la profondeur ou la largeur du réseau, EfficientNet utilise une méthode de mise à l'échelle composée (Compound Scaling).

Cette technique permet d'équilibrer uniformément la profondeur, la largeur et la résolution de l'image à l'aide d'un coefficient constant. En appliquant ce modèle à notre jeu de données de plus de 7 000 images réparties sur 20 classes, nous bénéficions d'une optimisation maximale : le modèle est suffisamment léger pour un entraînement rapide et une inférence fluide, tout en étant assez puissant pour capturer les détails complexes des textures et des couleurs des pétales. Le fine-tuning des paramètres d'ImageNet nous a permis d'adapter cette capacité d'abstraction aux spécificités botaniques de nos données, garantissant ainsi une convergence rapide et une haute précision.

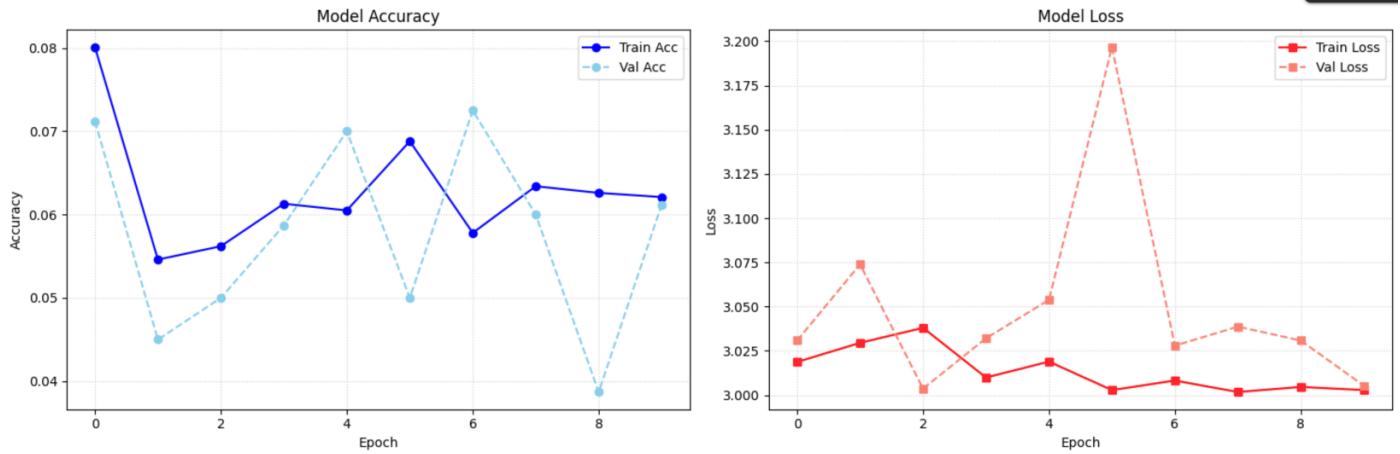


B) Modèle EfficientNet B2:

Pour notre modèle de classification, nous avons sélectionné l'architecture **EfficientNet-B2**, initialisée avec les poids de **ImageNet**. Ce choix représente un compromis optimal entre performance brute et efficacité computationnelle.

L'intérêt majeur d'EfficientNet-B2 réside dans sa méthode de **mise à l'échelle composée (Compound Scaling)**. Alors que la version B0 est le modèle de base, la version B2 augmente de manière équilibrée la profondeur des couches, la largeur du réseau et, surtout, la **Résolution d'entrée**. Avec notre jeu de données de plus de **7 000 images** et **20 classes** de fleurs, l'architecture B2 permet de capturer des détails morphologiques plus subtils (comme la forme des étamines ou la texture des sépales) que des modèles plus petits pourraient ignorer.

Le *fine-tuning* à partir des paramètres d'ImageNet nous permet de transférer des connaissances visuelles riches vers notre domaine spécifique. Grâce à sa structure optimisée, EfficientNet-B2 surpasse des modèles beaucoup plus lourds (comme ResNet-50) tout en restant plus rapide et moins gourmand en mémoire, garantissant ainsi une précision accrue pour nos 20 catégories botaniques sans sacrifier la fluidité de l'inférence.



V-Conclusion

En conclusion, le projet SuperEYE Search Engine (SSE) a permis de jeter les bases d'une solution technologique robuste alliant l'intelligence artificielle et la recherche d'information hybride au service de l'agriculture marocaine, transformant un simple diagnostic par image en un véritable outil d'aide à la décision. Pour l'avenir, cette plateforme est appelée à évoluer vers un écosystème métier complet en intégrant un marketplace de produits de traitement, un module de télé-expertise pour connecter agriculteurs et agronomes, ainsi qu'un système de cartographie en temps réel pour la surveillance épidémiologique régionale. Sur le plan technique, les perspectives d'ouverture sont tout aussi ambitieuses, incluant notamment l'intégration d'un chatbot intelligent (Rasa/LLM) pour une interaction naturelle, le déploiement de l'IA en mode hors-ligne (Edge AI) pour les zones rurales isolées, et l'usage de données IoT pour une détection prédictive des risques. Ainsi, en consolidant ces innovations fonctionnelles et technologiques, SuperEYE s'imposera comme un pilier de l'agriculture de précision, garantissant une meilleure productivité et une traçabilité renforcée pour les exploitations de demain.