



PROJET DE FIN D'ANNEE

5ème Année en Ingénierie Informatique et Réseaux

Suivi alimentaire avec OCR et IA

Réalisé par :

Yasmine BENOMAR

Yassemine EL MAAZIZI

Tuteur (s) :

MR. EL BAKKALI MOHAMMED

Année universitaire : 2025/2026



Dédicaces :

Au terme de ce projet, c'est avec un immense sentiment de gratitude que nous tenons à exprimer notre profonde reconnaissance à tous ceux qui ont contribué à son aboutissement.

En premier lieu, nous adressons nos plus sincères remerciements à nos parents pour leur soutien indéfectible tout au long de notre parcours.

Votre foi en nous et vos encouragements constants ont été notre source de motivation et de persévérance.

Nous tenons également à témoigner notre reconnaissance à nos professeurs et encadrant, particulièrement à Monsieur **EL BAKKALI MOHAMMED**, pour leur expertise, leurs conseils avisés et leur patience. Vous avez su nous guider et nous transmettre votre passion pour le domaine, faisant de ce projet une expérience enrichissante et stimulante.

À tous ceux qui ont contribué, de près ou de loin, à cette aventure, nous dédions ce travail avec gratitude et humilité.

Que cette humble réalisation soit le témoignage de notre apprentissage et le prélude à de nouvelles découvertes.

Merci pour chaque encouragement, chaque conseil et chaque sourire qui ont rendu ce voyage inoubliable.



Remerciements :

<< Le vrai secret du bonheur réside dans la gratitude. >> - Marcel Proust

Tout d'abord, nous tiendrons à remercier le corps professoral qui nous a encadré et orienté vers la bonne direction. Un grand merci au département informatique. Nous tenons à exprimer notre sincère gratitude envers nos professeurs et notre encadrant Monsieur **EL BAKKALI MOHAMMED**, pour leur dévouement, leur soutien précieux qui ont été essentiels à la réalisation de ce projet.

Leurs conseils avisés, leur disponibilité et leur encouragement constant ont été une source d'inspiration et de motivation tout au long de notre parcours.

Nous adressons également nos remerciements aux membres du jury pour leur temps, leur expertise et leur investissement dans l'évaluation de notre travail. Leurs retours constructifs ont enrichi notre réflexion et ont contribué à notre développement académique. Enfin, nous exprimons notre reconnaissance envers toutes les personnes qui ont joué un rôle, grand ou petit, dans la réalisation de ce projet.

Votre soutien et vos encouragements ont été des éléments clés de notre réussite académique et nous vous en sommes profondément reconnaissants.



Table des matières:

Dédicaces :	2
Remerciements :	3
TABLE DE FIGURE :	6
Liste des tableaux	7
Introduction général :	8
Chapitre1: Présentation cadre de Projet	9
1. Introduction :	10
1. État de l'art :	10
2. Présentation du projet :	10
1. Problématique :	11
2. Motivation :	11
3. Solution :	11
3. Démarche et planification :	11
3.1 Méthodologies du travail :	11
3.2. Calendrier prévisionnelle :	12
3.3. Diagramme de Gantt :	13
4. Conclusion :	14
Chapitre 2: Analyse et Conception	15
1. Introduction :	16
2. Cahier de charge :	16
2.1. Etude de l'existant:	16
2.2. Description de l'existant :	16
Solutions suggérées :	16
2.3. Identification des besoins :	17
2.3. Identification des acteurs :	17
2.4. Besoins des acteurs :	17
3. Etude conceptuelle :	18
3.1. Diagramme de cas d'utilisation :	18
3.2. Diagramme de séquence:	19
➤ <i>Diagramme de séquence (recevoir des recommandations nutritionnelles)</i> :	21
3.3. Diagramme de classe:	21
4. Conclusion :	22
Chapitre 3 : Réalisation du Système	23



1. Introduction :	24
2. Environments de travail:	24
2.1. IntelliJ IDEA:	24
2.2. postgresql:	24
2.3. vs code:	24
3. Language utilisé :	25
3.1. Java:	25
3.2. python:	25
3.3. spring boot:	25
3.4. Angular:	25
3.5. Kafka:	26
4. Présentation des interfaces graphiques :	26
inscription :	26
Page de connexion.	27
Page de resultat calories IA .	27
Page scan du repas :	28
Page Historique Nutritionnel :	29
5. Conclusion :	31
Conclusion générale :	32

TABLE DE FIGURE :

Figure 1 :diagramme de cas d'utilisation.....	19
Figure 2 :diagramme de séquence (enregistrement de repas dans le journal)	20
Figure 3 :diagramme de séquence(consulter les calories détectées)	20
Figure 4 :diagramme de séquence (recevoir des recommandations nutritionnelles).....	21
Figure 5 :diagramme de classe	21
Figure 6 :logo intellij idea.....	24
Figure 7 :logo postgresql.....	24
Figure 8 :logo vs code.....	24
Figure 9 :logo java.....	25
Figure 10 :logo python	25
Figure 11 :logo spring boot.....	25
Figure 12 :logo angular.....	25
Figure 13 :logo kafka	26
Figure 14 :interface inscription.....	26
Figure 15 :interface connexion.....	27
Figure 16 :interface de resultat calories IA	27
Figure 17 :interfacce scan repas	28
Figure 18 :interface historique nutritionnel	29
Figure 19 :interface Gestion Aliments.....	30



Liste des tableaux

Tableau 1 : tableau des tâches.....	13
Tableau 2 : tableau besoin des acteurs :.....	18



Introduction général :

À l'ère du numérique, la santé et le bien-être occupent une place de plus en plus centrale dans les préoccupations individuelles et sociétales. Parmi les facteurs déterminants de la santé, l'alimentation joue un rôle essentiel dans la prévention de nombreuses maladies chroniques telles que l'obésité, le diabète et les troubles cardiovasculaires. Une alimentation équilibrée et contrôlée contribue significativement à l'amélioration de la qualité de vie et à la réduction des risques sanitaires à long terme.

Cependant, le suivi alimentaire quotidien demeure une tâche contraignante pour de nombreux individus. Les méthodes traditionnelles reposant sur la saisie manuelle des aliments consommés sont souvent chronophages, peu précises et sujettes à l'abandon. Malgré l'existence de nombreuses applications numériques dédiées au suivi nutritionnel, la majorité d'entre elles exigent encore une intervention manuelle importante de la part de l'utilisateur, ce qui limite leur efficacité et leur adoption à long terme.

Les avancées récentes dans les domaines de l'intelligence artificielle (IA) et de la reconnaissance optique de caractères (OCR) offrent de nouvelles opportunités pour automatiser et simplifier ce processus. Grâce à ces technologies, il devient possible d'analyser automatiquement des images de repas ou des tickets de caisse afin d'extraire des informations nutritionnelles pertinentes et de fournir des recommandations personnalisées.

C'est dans ce contexte que s'inscrit ce projet, qui vise à concevoir et modéliser une application web intelligente de suivi alimentaire basée sur l'OCR et l'IA. Cette application a pour objectif d'automatiser la collecte et l'analyse des données nutritionnelles tout en offrant une expérience utilisateur fluide, intuitive et moderne. Elle repose sur une architecture microservices utilisant Spring Boot pour le backend et Angular pour le frontend, garantissant modularité, évolutivité et maintenabilité.

Ce rapport présente l'ensemble des étapes suivies pour la réalisation de ce projet, depuis l'étude du contexte et l'analyse des besoins jusqu'à la conception et la réalisation du système.

Chapitre 1 : Ce chapitre présente le contexte général du projet et l'état de l'art des solutions existantes en suivi alimentaire. Il expose la problématique, les objectifs visés et la solution proposée. Il décrit également la méthodologie adoptée et la planification du projet.

Chapitre 2 : Ce chapitre est consacré à l'analyse des besoins fonctionnels et non fonctionnels de l'application. Il présente le cahier des charges ainsi que la modélisation du système à travers des diagrammes UML. Ces éléments permettent de définir l'architecture globale de la solution.

Chapitre 3 : Ce chapitre décrit la mise en œuvre technique de l'application et les technologies utilisées. Il présente les environnements de développement, les langages et frameworks adoptés. Il illustre également les principales interfaces graphiques de l'application.



Chapitre1: Présentation cadre de Projet

1. Introduction :

L'alimentation constitue l'un des déterminants majeurs de la santé humaine. Les études scientifiques ont établi le lien entre habitudes alimentaires et pathologies chroniques.

Parallèlement, l'intelligence artificielle connaît une transformation radicale avec des applications concrètes dans la santé et la nutrition.

Ce chapitre pose les fondations de notre projet en explorant l'état actuel des technologies de suivi alimentaire, leurs limites, et les opportunités d'amélioration.

Nous présenterons la problématique, nos motivations, la solution proposée, et notre méthodologie de travail.

1. État de l'art :

Le marché actuel des applications de suivi alimentaire est dominé par plusieurs acteurs majeurs qui proposent des approches variées pour répondre aux besoins des utilisateurs soucieux de leur nutrition.

MyFitnessPal est l'une des applications les plus populaires avec plus de 200 millions d'utilisateurs dans le monde.

Elle propose une vaste base de données de plus de 14 millions d'aliments et permet aux utilisateurs d'enregistrer leurs repas quotidiens. Cependant, la saisie reste essentiellement manuelle : l'utilisateur doit rechercher chaque aliment, ajuster les portions et valider les entrées. Bien que l'application intègre un lecteur de codes-barres pour faciliter l'enregistrement des produits emballés, cette fonctionnalité ne couvre pas les repas faits maison ou les plats de restaurant.

Yuka adopte une approche différente en se concentrant sur l'analyse de la qualité nutritionnelle des produits alimentaires via le scan de codes-barres.

L'application attribue une note de 0 à 100 basée sur la qualité nutritionnelle, la présence d'additifs et le caractère biologique du produit. Toutefois, Yuka ne propose pas de suivi alimentaire quotidien complet et se limite aux produits industriels disposant d'un code-barres.

Foodvisor représente une avancée significative en intégrant la reconnaissance d'images basée sur l'intelligence artificielle. Les utilisateurs peuvent photographier leurs repas et l'application identifie automatiquement les aliments présents dans l'assiette pour estimer les valeurs nutritionnelles. Cette approche innovante réduit considérablement le temps de saisie.

Néanmoins, la précision de la reconnaissance reste perfectible, notamment pour les plats complexes ou les recettes régionales, et l'application propose des fonctionnalités avancées uniquement dans sa version premium.

2. Présentation du projet :

Notre projet vise à développer une application web innovante de suivi alimentaire qui répond aux limitations identifiées dans les solutions existantes.

L'ambition principale est de créer une expérience utilisateur fluide et intuitive où la technologie se fait transparente, permettant à chacun de surveiller son alimentation sans effort conscient ni contrainte temporelle.

L'application s'articule autour de trois piliers technologiques majeurs : **la reconnaissance optique de caractères (OCR)** pour extraire automatiquement les informations depuis les tickets de caisse ou les menus photographiés, la reconnaissance d'images **basée sur l'IA** pour identifier les aliments présents dans les photos de repas, et un système de recommandations intelligent qui apprend des habitudes de l'utilisateur pour proposer des conseils nutritionnels personnalisés et évolutifs. L'architecture technique repose sur une approche microservices

développée avec Spring Boot, garantissant une séparation claire des responsabilités, une scalabilité horizontale aisée et une maintenance facilitée.

Cette modularité permet également une évolution progressive du système, avec la possibilité d'ajouter de nouveaux services ou de remplacer des composants existants sans impact majeur sur l'ensemble de l'application.

1. Problématique :

la problématique centrale que notre projet cherche à résoudre peut être formulée ainsi : Comment automatiser et simplifier radicalement le processus de suivi alimentaire tout en maintenant une précision nutritionnelle élevée et en offrant des recommandations personnalisées pertinentes ?

2. Motivation :

Le projet est motivé par la volonté de :

- Proposer une solution spécialisée dans le domaine de la nutrition et du suivi alimentaire, en intégrant des technologies avancées comme l'OCR et l'IA pour automatiser l'analyse des repas.
- Offrir une interface simple, fluide et fonctionnelle pour les utilisateurs finaux, avec un frontend responsive en Angular et un design intuitif adapté à tous les niveaux d'expertise .
- ·Introduire une touche d'intelligence artificielle pour l'optimisation des calculs nutritionnels, l'estimation des calories et des nutriments, ainsi que des recommandations personnalisées basées sur l'historique des utilisateurs .
- Exploiter les outils modernes (Spring Boot pour une architecture microservices scalable, Angular pour le frontend dynamique, OCR via Tesseract ou similaires, et modèles IA comme TensorFlow) dans un contexte concret de santé digitale.

3. Solution :

Après analyse des besoins, la réalisation d'une application web s'impose comme la solution la plus adaptée.

Celle-ci facilitera la gestion des aliments pour les utilisateurs et l'analyse nutritionnelle des repas, de manière simple et efficace.

De plus, elle permettra une meilleure organisation, réduira les erreurs de saisie manuelle et améliorera l'accès aux informations nutritionnelles, le tout via une interface conviviale et moderne. L'intégration d'un algorithme d'OCR et d'intelligence artificielle contribuera à automatiser l'extraction et l'optimisation des données nutritionnelles selon les habitudes des utilisateurs.

3.Démarche et planification :

3.1 Méthodologies du travail :

Analyse des besoins et spécifications : Nous effectuerons une recherche approfondie sur les besoins du cahier des charges pour faciliter la manière de travail, en identifiant les exigences fonctionnelles comme le suivi des aliments, l'analyse nutritionnelle via OCR et IA, et les contraintes techniques telles que l'intégration de PostgreSQL comme base de données.

Conception et modélisation de l'application : Nous définirons les différentes fonctionnalités ainsi que les interactions de notre application web afin d'assurer sa conception et son fonctionnement optimaux, incluant la modélisation des microservices (Authentication, Discovery, Gateway, GestionAliments, HistoriqueNutritionnel, IACalories, OCRIngestion)



et les diagrammes UML pour le frontend Angular.

Développement de l'application : Nous utiliserons des technologies modernes adaptées au développement web (Angular pour le frontend avec Tailwind CSS, Spring Boot pour l'architecture microservices backend, et PostgreSQL pour la gestion des données) pour coder les fonctionnalités essentielles telles que la gestion des aliments, l'historique nutritionnel, le calcul des calories via IA, l'ingestion de données OCR, ainsi que l'intégration d'une intelligence artificielle pour l'analyse automatique des nutriments et des recommandations personnalisées.

Tests et débogage : À la fin de chaque phase de codage, nous procédons à des tests approfondis pour identifier les éventuelles erreurs ou anomalies. Cette étape nous permet de garantir le bon fonctionnement de chaque partie de l'application web avant d'envisager sa mise en production. Les tests comprennent à la fois des tests unitaires et des tests d'intégration (Postman pour APIs).

Rédaction du rapport : Pendant la période de développement de l'application, nous commençons à effectuer en parallèle la rédaction du rapport, en décrivant les différentes étapes effectuées durant ce projet, les choix technologiques ainsi que les problèmes rencontrés et leur solution.

Préparation de la présentation PowerPoint : Après que l'application et le rapport aient été terminés, nous pouvons commencer à préparer une présentation PowerPoint. Cette présentation abordera les problèmes rencontrés, les solutions apportées, ainsi que les fonctionnalités de l'application web, en mettant en avant les résultats obtenus.

3.2. Calendrier prévisionnelle :

➤ Les tâches du projet :

L'accomplissement de ce projet s'est effectué à travers la réalisation d'une série de différentes étapes :

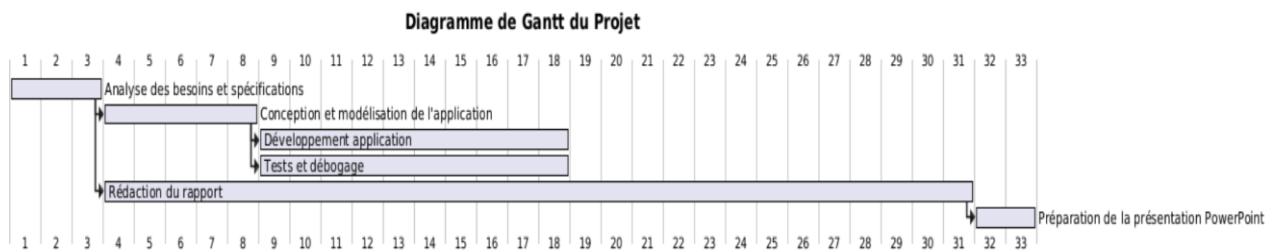
- ✓ Analyse des besoins et spécifications : 3 jours
- ✓ Conception et modélisation de l'application : 5 jours
- ✓ Développement de l'application : 10 jours.
- ✓ Tests et débogage : 10 jours.
- ✓ Rédaction du rapport : 28 jours.
- ✓ Préparation de la présentation PowerPoint : 2 jours.

- Le tableau ci-dessous détaille les tâches effectuées dans ce projet :

Tâches		Durée(jrs)	Antécédant
1	Analyse des besoins et spécifications	3	--
2	Conception et modélisation de l'application	5	1
3	Développement application	10	2
4	Tests et débogage	10	2
5	Rédaction du rapport	28	1
6	Préparation de la présentation PowerPoint	2	5

Tableau 1 : tableau des taches

3.3. Diagramme de Gantt :



Ce diagramme de Gantt représente la planification d'un projet de développement d'application sur une durée totale d'environ 58 jours. Il illustre six tâches principales avec leurs durées et dépendances :

Le projet commence par :

l'**analyse des besoins** (3 jours), suivie de la **conception** (5 jours). Ensuite, deux tâches se déroulent en parallèle :

- Le **développement** et les **tests** (10 jours chacune).
- La **rédaction du rapport** est la tâche la plus longue (28 jours) et démarre dès la fin de l'analyse.
- La **préparation de la présentation PowerPoint** (2 jours) clôture le projet après la fin du rapport.

Les dépendances entre tâches sont clairement visualisées, montrant comment certaines activités doivent attendre la fin d'autres avant de commencer, permettant ainsi une gestion efficace du temps et des ressources du projet.



4. Conclusion :

Comme on le voit dans ce chapitre, nous avons eu un aperçu global de notre sujet ainsi que de la méthodologie à suivre pour réaliser ce projet. Cette étape est donc primordiale pour acquérir le savoir nécessaire afin de définir les contours de sa réalisation. Le chapitre suivant sera donc consacré à spécifier les scénarios possibles d'utilisation à partir desquels nous extrairons les besoins.



Chapitre 2: Analyse et Conception



1. Introduction :

Dans le cycle de développement d'une application, l'analyse et la spécification des besoins constituent une étape fondamentale. Elles permettent d'identifier les acteurs du système et de définir leurs interactions afin d'assurer un résultat conforme aux attentes du client. Dans ce chapitre, nous commencerons par spécifier les besoins de l'application, avant de passer à leur analyse. Ensuite, nous procéderons à la modélisation et à la conception du système à travers des diagrammes UML.

2. Cahier de charge :

2.1. Etude de l'existant:

Des applications telles que MyFitnessPal ou Yazio permettent le suivi des apports caloriques et nutritionnels, mais elles sont souvent trop généralistes, dépendantes de bases de données propriétaires, et ne proposent pas de fonctionnalités adaptées à l'analyse automatique des repas via OCR et IA pour des contextes personnalisés ou locaux.

Aucune application locale simple ne propose à la fois la gestion d'une base nutritionnelle privée, l'historique des repas avec extraction automatique des nutriments via OCR, et la suggestion intelligente de recommandations nutritionnelles en fonction des habitudes et données historiques des utilisateurs.

2.2. Description de l'existant :

Actuellement, le suivi alimentaire se fait majoritairement de manière manuelle via des journaux papier ou des applications basiques de tracking calorique. Cela entraîne plusieurs inconvénients, tels que :

- ✓ Difficulté à estimer précisément les calories et nutriments sans outils automatisés, menant à des erreurs fréquentes.
- ✓ Problème de saisie manuelle fastidieuse et chronophage pour chaque repas ou aliment consommé.
- ✓ Absence de centralisation des données nutritionnelles historiques, rendant difficile l'analyse à long terme.
- ✓ Risque d'oublis ou d'inexactitudes dans les apports journaliers pour les utilisateurs.
- ✓ Charge de travail accrue pour les nutritionnistes ou utilisateurs cherchant des recommandations personnalisées sans IA intégrée.

Solutions suggérées :

Permettre aux utilisateurs de scanner des repas via OCR pour analyser automatiquement les nutriments et calories.

- Offrir un espace personnel pour gérer la base d'aliments et l'historique nutritionnel.
- Ajouter un historique des repas pour améliorer le suivi alimentaire.
- Introduire un algorithme simple d'IA pour suggérer des recommandations nutritionnelles selon les habitudes de l'utilisateur.

2.3. Identification des besoins :

➤ Besoins fonctionnels :

Les besoins fonctionnels définissent les actions spécifiques que le système doit accomplir pour répondre aux attentes des utilisateurs. L'application web de suivi alimentaire avec OCR et IA doit répondre aux besoins suivants :

Création de compte utilisateur : chaque utilisateur peut créer un compte en saisissant ses informations personnelles.

- Authentification des utilisateurs : chaque utilisateur se connecte avec son adresse e-mail et son mot de passe, via le microservice Authentication.
- Gestion des aliments (utilisateurs) : les utilisateurs peuvent consulter la liste des aliments disponibles, ajouter, modifier ou supprimer un aliment avec ses détails nutritionnels (calories, nutriments).
- Analyse des repas via OCR (utilisateurs) : les utilisateurs peuvent uploader une photo de repas ou de ticket de caisse, et le système extrait automatiquement les informations via le microservice OCRIgestion.
- Calcul et analyse nutritionnelle (utilisateurs) : l'application utilise l'IA pour estimer calories et nutriments d'un repas, via le microservice IACalories.
- Suivi de l'historique nutritionnel (utilisateurs) : les utilisateurs peuvent consulter leurs repas passés et analyses nutritionnelles, via le microservice HistoriqueNutritionnel.
- Recommandations personnalisées : un algorithme IA basé sur l'historique et les habitudes suggère automatiquement des ajustements nutritionnels ou des aliments alternatifs.

➤ Besoins non fonctionnels :

- Portabilité : l'application est accessible sur tous les navigateurs web modernes et dispositifs en tant qu'application web responsive.
- Facilité d'utilisation : l'interface est intuitive, fluide et ergonomique, adaptée aux profils non techniques, avec un design mobile-first pour une navigation simple et rapide.
- Fiabilité : l'application fonctionne de manière stable, sans interruption ni perte de données, grâce à une architecture microservices robuste et une base de données PostgreSQL pour une persistance fiable.

2.3. Identification des acteurs :

En génie logiciel et plus particulièrement en UML, un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé : Les acteurs présents dans notre application sont :

➤ Utilisateur

2.4. Besoins des acteurs :

- Le tableau suivant (tableau 2) présente les acteurs intervenant dans notre application ainsi que leurs besoins :

Acteur	Besoin
Utilisateur	<ul style="list-style-type: none"> - Créer un compte / Se connecter - Rechercher et consulter la liste des aliments disponibles dans la base nutritionnelle - Sélectionner un aliment et consulter ses détails nutritionnels (calories, nutriments) - Ajouter ou modifier un aliment en remplissant un formulaire avec nom, calories par portion et nutriments - Uploader une photo de repas ou ticket de caisse pour analyse via OCR - Bénéficier de suggestions nutritionnelles optimales via IA, selon l'historique des repas et les besoins de l'utilisateur - Consulter son historique nutritionnel : repas à venir et passés (journal quotidien)

Tableau 2 : tableau besoin des acteurs :

3. Etude conceptuelle :

3.1. Diagramme de cas d'utilisation :

➤ Définition :

Le diagramme de cas d'utilisation est un type de diagramme UML comportemental et est fréquemment utilisé pour analyser divers systèmes. Ils vous permettent de visualiser les différents types de rôles dans un système et la façon dont ces rôles interagissent avec le système.

➤ Acteurs :

Un acteur, au sens UML, représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. Il est représenté par un bonhomme en fil de fer. On représente généralement à gauche l'acteur principal, et à droite les acteurs secondaires. Il est à noter qu'un utilisateur peut amener à jouer plusieurs rôles vis-à-vis du système et à ce titre être modélisé par plusieurs acteurs. Pour notre application il y a qu'un acteur principale

➤ Diagramme de cas d'utilisation :

Suite à l'élaboration du cahier des charges, nous avons défini les différents cas d'utilisation représentant les interactions possibles entre les utilisateurs (acteurs) et notre application web. Ces cas d'utilisation permettent de visualiser les rôles et les fonctionnalités principales offertes par le système.

➤ Pour l'utilisateur :

L'utilisateur peut effectuer les actions suivantes dans l'application :

- Accepter ou refuser une analyse de repas proposée par l'IA.
- Ajouter des aliments à la base nutritionnelle (nom, calories, nutriments).
- Afficher la liste des aliments disponibles.
- Consulter les détails d'un aliment (calories par portion, nutriments, historique d'utilisation).

- Marquer un repas comme "complet" ou planifier un ajustement nutritionnel en cas de déséquilibre détecté.
- Consulter l'historique des repas passés.

- La figure 6 ci-dessous montre l'interaction de tous les acteurs avec l'application :

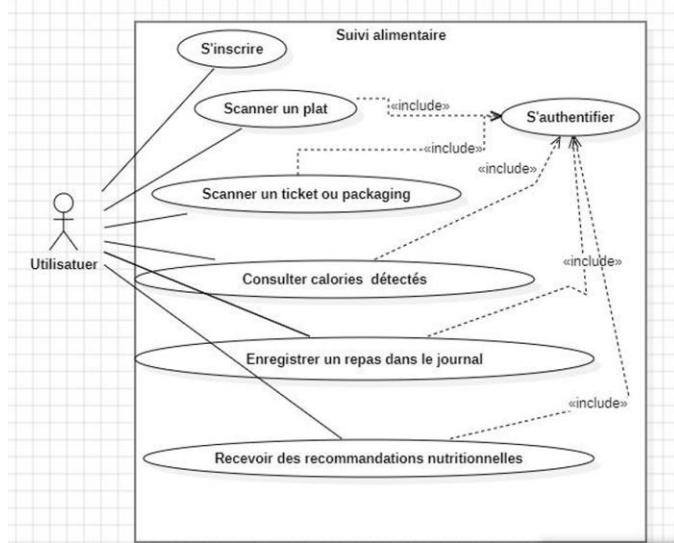


Figure 1 :diagramme de cas d'utilisation

Le diagramme des cas d'utilisation présenté ci-dessus illustre les interactions principales entre l'utilisateur et le système de suivi alimentaire avec OCR et IA. L'acteur principal est l'Utilisateur, représenté par une figure humaine, qui interagit avec plusieurs cas d'utilisation ovales connectés par des lignes pointillées ou solides indiquant des inclusions ou des extensions. Les cas d'utilisation incluent "S'inscrire" pour la création de compte, "S'authentifier" pour la connexion sécurisée, "Scanner un plat" pour l'analyse d'images de repas via OCR, "Scanner un ticket ou packaging" pour l'extraction d'informations nutritionnelles à partir de tickets de caisse ou emballages, "Consulter calories détectées" pour visualiser les estimations caloriques, "Enregistrer un repas dans le journal" pour stocker les données des repas dans un historique, et "Recevoir des recommandations nutritionnelles" pour obtenir des suggestions personnalisées basées sur l'IA.

3.2. Diagramme de séquence:

➤ Définition :

Un diagramme de séquence est un diagramme UML qui représente la séquence de messages entre les objets au cours d'une interaction. Il comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

➤ *Diagramme de séquence pour l'enregistrement du repas dans le journal:*

- Le diagramme ci-dessous montre l'enregistrement du repas dans le journal:

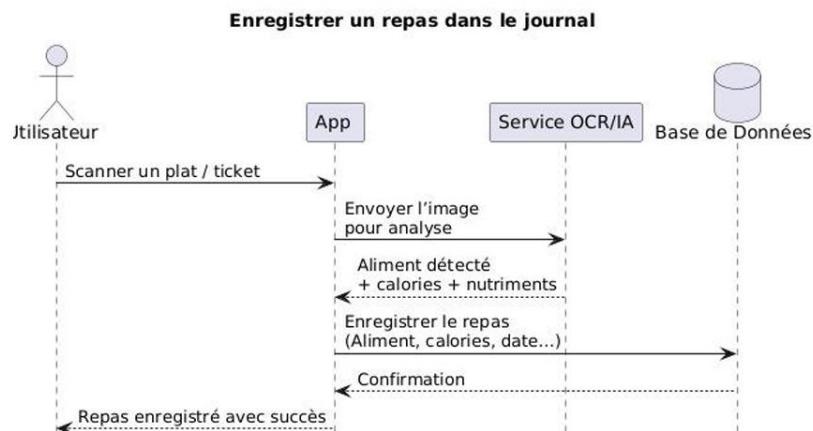


Figure 2 :diagramme de séquence (enregistrement de repas dans le journal)

Ce diagramme de séquence UML illustre le processus d'enregistrement d'un repas dans le journal alimentaire. L'utilisateur commence par scanner un plat ou un ticket, l'app envoie l'image au service OCR/IA qui détecte l'aliment, les nutriments et les calories. L'app enregistre ensuite les données (aliment, calories, date) dans la base de données, qui confirme l'opération, et l'app retourne un message de succès à l'utilisateur. Les flèches indiquent les flux d'inclusion et les interactions entre composants.

➤ **Diagramme de séquence (consulter les calories détectées):**

- Ce diagramme illustre consulter les calories détectées :

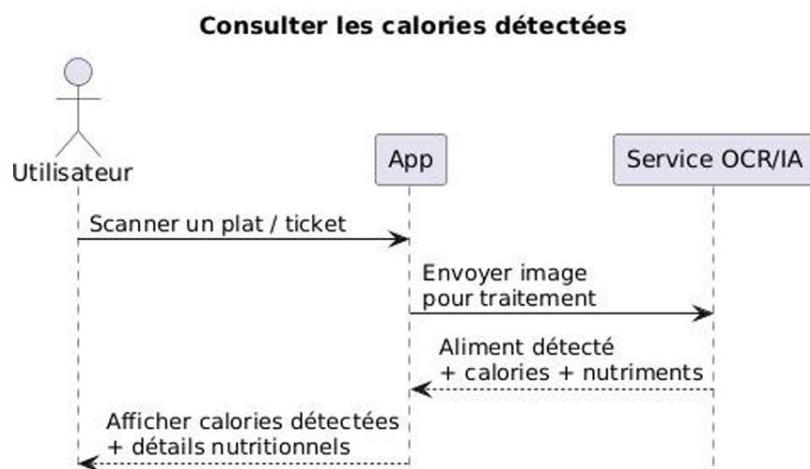


Figure 3 :diagramme de séquence(consulter les calories détectées)

➤ *Diagramme de séquence (recevoir des recommandations nutritionnelles):*

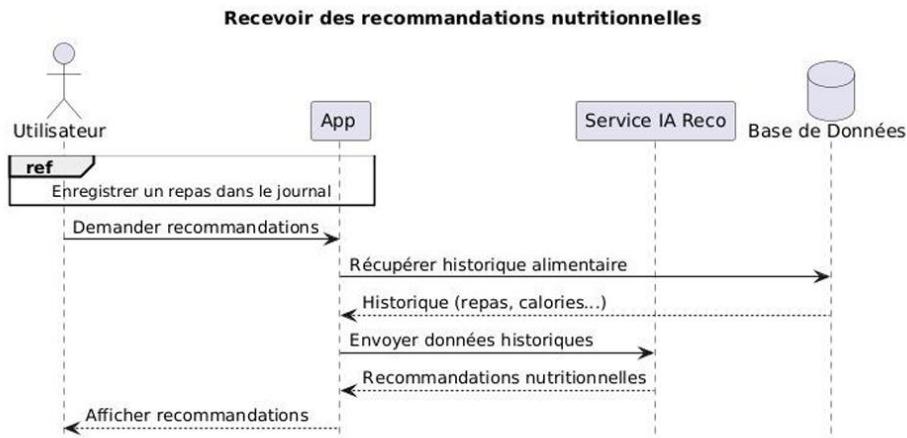


Figure 4 :diagramme de séquence (recevoir des recommandations nutritionnelles)

3.3. Diagramme de classe:

➤ Définition :

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

➤ Diagramme de classe :

l'illustration ci-après montre le diagramme de classe pour le projet :

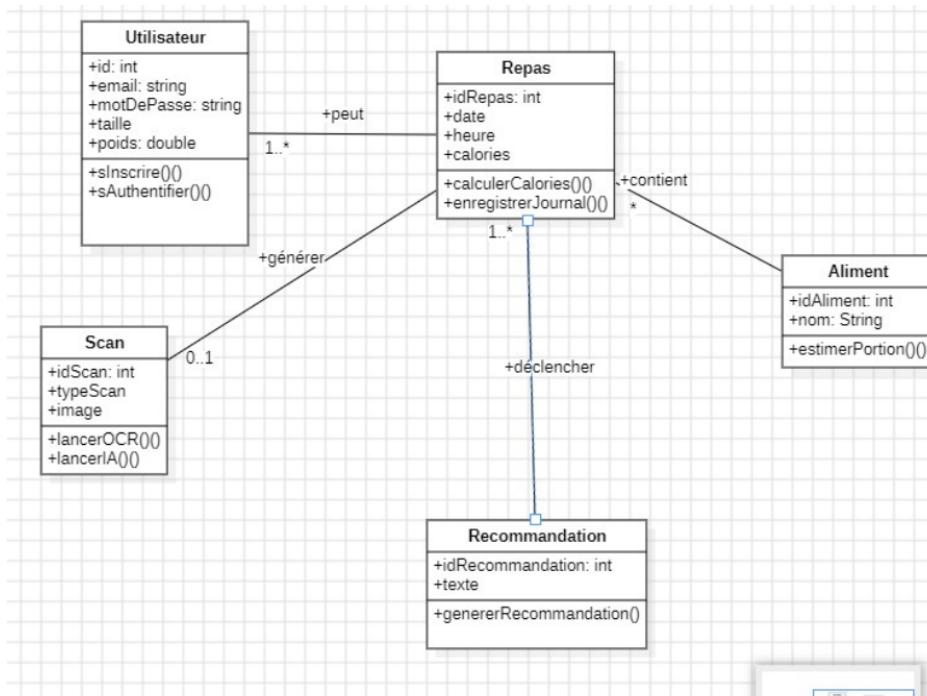


Figure 5 :diagramme de classe



4. Conclusion :

En guise de conclusion, ce chapitre a exploré le contexte global et le cadrage de notre application web de suivi alimentaire intelligent avec OCR et IA.

Cette phase constitue une étape essentielle permettant de structurer efficacement la démarche de développement et de poser les fondations conceptuelles du projet. Nous avons analysé l'état de l'art des solutions existantes, identifié leurs limitations, et présenté les technologies modernes qui rendent possible une approche radicalement innovante.



Chapitre 3 : Réalisation du Système

1. Introduction :

Après la phase de conception, l'implémentation entre en jeu, jouant un rôle très important pour le processus de développement. Le choix des outils de développement est déterminant, influençant à la fois le temps de programmation et la flexibilité du produit final. L'objectif de cette phase est de transformer le modèle conceptuel en composants logiciels constituant notre système. Ce chapitre traitera des outils soigneusement sélectionnés pour répondre aux besoins du client ainsi que des différentes fonctionnalités offertes aux utilisateurs.

2. Environments de travail:

2.1. IntelliJ IDEA:

IntelliJ IDEA est un environnement de développement intégré (IDE) créé par JetBrains, utilisé principalement pour développer des applications en Java, mais aussi en Kotlin, Spring, JavaScript, SQL, etc.

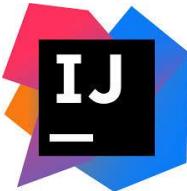


Figure 6 :logo intellij idea

2.2. postgresql:

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBD), open source, puissant et fiable.



Figure 7 :logo postgresql

2.3. vs code:

VS Code (Visual Studio Code) est un éditeur de code source gratuit développé par Microsoft. Il permet aux développeurs d'écrire, modifier, tester et déboguer du code dans de nombreux langages comme Python, JavaScript, Java, C++, HTML, CSS, etc.



Figure 8 :logo vs code

3. Language utilisé :

3.1. Java:

Java est un langage de programmation largement utilisé pour coder des applications web ,est un langage multiplateforme, orienté objet et centré sur le réseau, qui peut être utilisé comme une plateforme à part entière.

Il s'agit d'un langage de programmation rapide, sécurisé et fiable qui permet de tout coder, des applications mobiles aux logiciels d'entreprise en passant par les applications de big data et les technologies côté serveur.



Figure 9 :logo java

3.2. python:

Python est un langage de programmation de haut niveau, interprété et open source, connu pour sa syntaxe simple et lisible.



Figure 10 :logo python

3.3. spring boot:

Spring Boot :permet de développer rapidement des applications Java robustes et prêtes pour la production, en réduisant la configuration manuelle.



Figure 11 :logo spring boot

3.4. Angular:

Angular : est un framework JavaScript/TypeScript développé par Google pour créer des applications web dynamiques et interactives côté client (frontend).



Figure 12 :logo angular

3.5. Kafka:

Apache Kafka :est une plateforme de streaming distribuée qui permet de publier, stocker et traiter des flux de données en temps réel.



Figure 13 :logo kafka

4. Présentation des interfaces graphiques :

inscription :

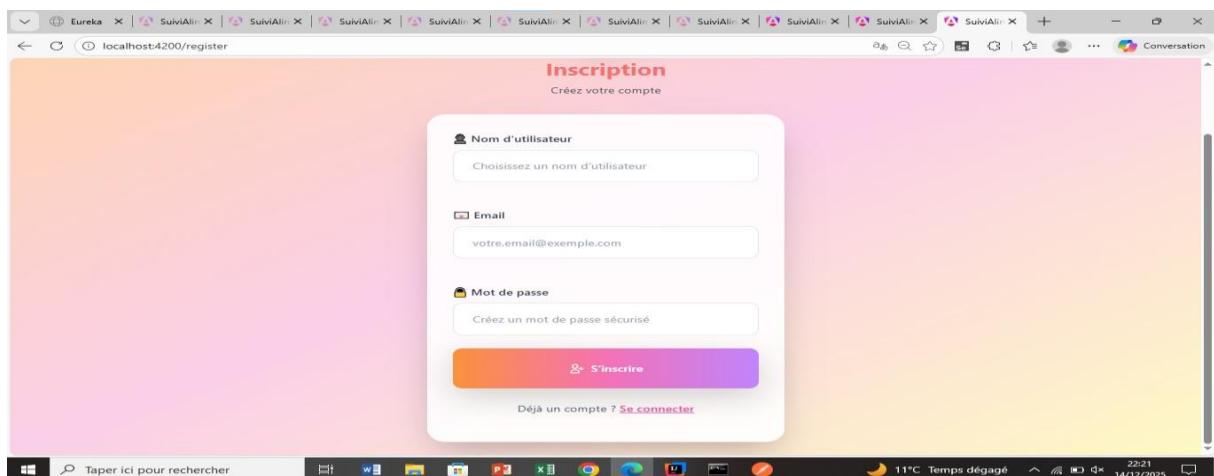


Figure 14 :interface inscription

Ce formulaire d'inscription présente une interface épurée et fonctionnelle, centrée sur la création de compte utilisateur. Composé de trois champs essentiels (nom d'utilisateur, email et mot de passe), il guide l'utilisateur avec des textes indicatifs clairs et propose un bouton d'action visible intitulé "S'inscrire". Un lien en bas permet de rediriger les utilisateurs déjà inscrits vers la connexion, bien qu'une coquille soit présente dans le texte ("Sa connecteur"). L'ensemble repose sur une mise en page aérée et intuitive, facilitant l'expérience d'inscription.

Page de connexion.

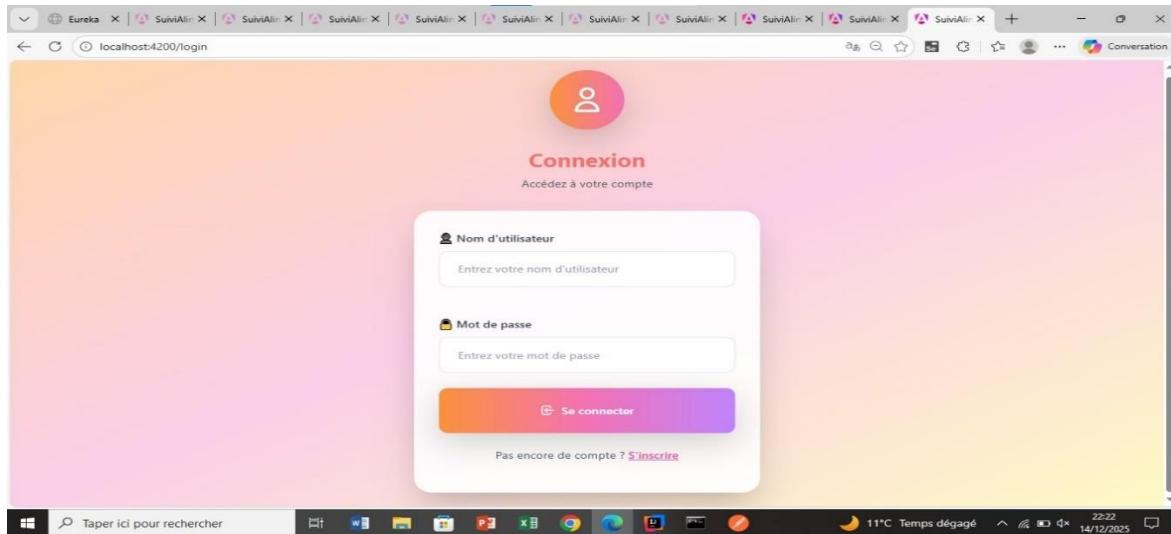


Figure 15 :interface connexion

Ce formulaire de connexion présente un bouton "Se connecter" bien mis en valeur et un lien vers l'inscription pour les nouveaux utilisateurs. La formulation claire et directe facilite l'authentification rapide et intuitive.

Page de résultat calories IA .

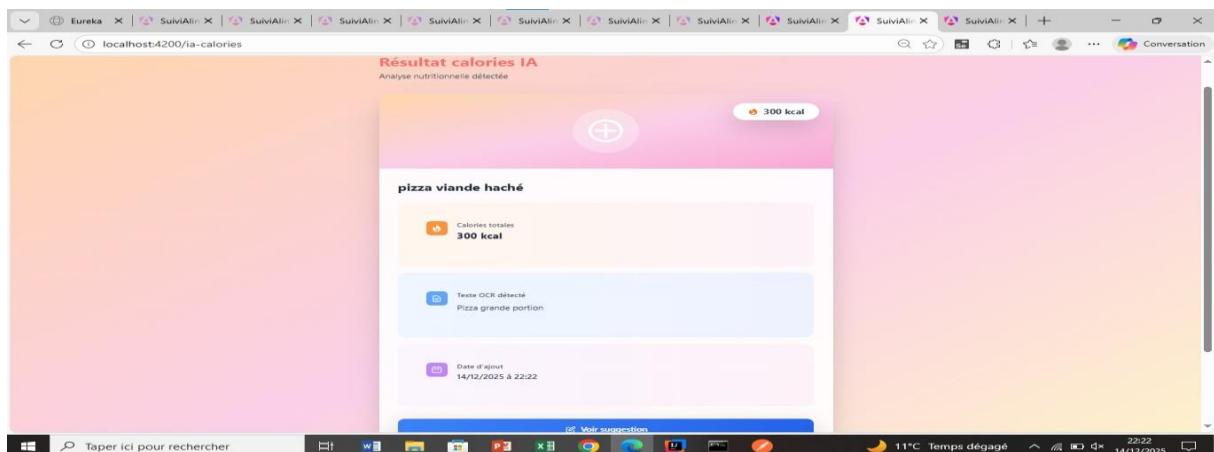


Figure 16 :interface de résultat calories IA

Cette interface de résultats d'analyse nutritionnelle par IA présente de manière claire et structurée les informations extraites d'un aliment, ici une "pizza viande haché". La valeur calorique principale (300 kcal) est mise en avant visuellement, tandis que les détails complémentaires (identification de la portion, date d'ajout)

Page scan du repas :

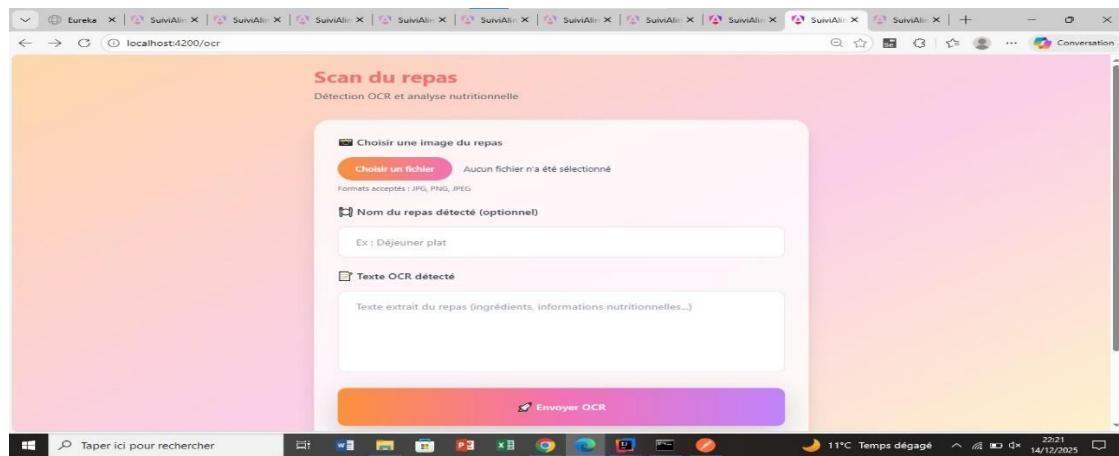


Figure 17 :interfacce scan repas

Cette interface présente une fonctionnalité de "Scan du repas" utilisant la détection OCR et l'analyse nutritionnelle. L'écran permet d'extraire automatiquement les informations nutritionnelles à partir d'une photo de repas ou d'étiquette alimentaire. Composants principaux :

1. Section d'upload d'image : o Titre "Choisir une image du repas" avec icône d'appareil photo o Bouton dégradé orange-rose "Choisir un fichier" o Message de statut : "Aucun fichier n'a été sélectionné" o Formats acceptés : JPG, PNG, JPEG
2. Champ "Nom du repas détecté (optionnel)" : o Zone de texte avec placeholder "Ex : Déjeuner plat" o Permet de nommer manuellement le repas scanné
3. Zone "Texte OCR détecté" : o Grande zone de texte en lecture seule o Affichera le texte extrait (ingrédients, informations nutritionnelles) o Placeholder explicatif en gris
4. Bouton d'action "Envoyer OCR" : o Grand bouton avec dégradé orange-rose-violet o Lance le traitement OCR de l'image

Page Historique Nutritionnel :

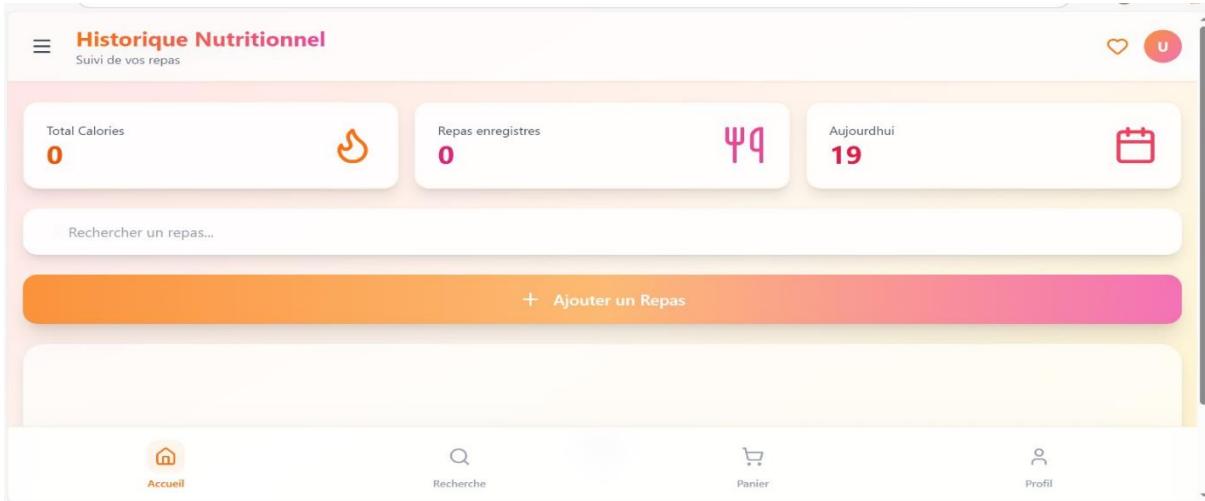


Figure 18 :interface historique nutritionnel

Cette interface présente un formulaire d'ajout de repas dans une application de suivi nutritionnel. L'écran "Historique Nutritionnel" avec le sous-titre "Suivi de vos repas" permet d'enregistrer un nouveau repas via plusieurs champs : Formulaire "Nouveau repas" :

- Type de repas : menu déroulant avec le placeholder "Sélectionnez un type" (petit-déjeuner, déjeuner, dîner, etc.)
- Date et heure : champ avec format "jj/mm/aaaa --:--" et icône de calendrier pour sélectionner la date
- Calories : champ numérique affichant "0" par défaut
- Nutriments : champ de texte avec exemple "Ex: Proteines: 20g, Glucides: 50g, Lipides: 15g" pour guider l'utilisateur

Gestion d'aliment :

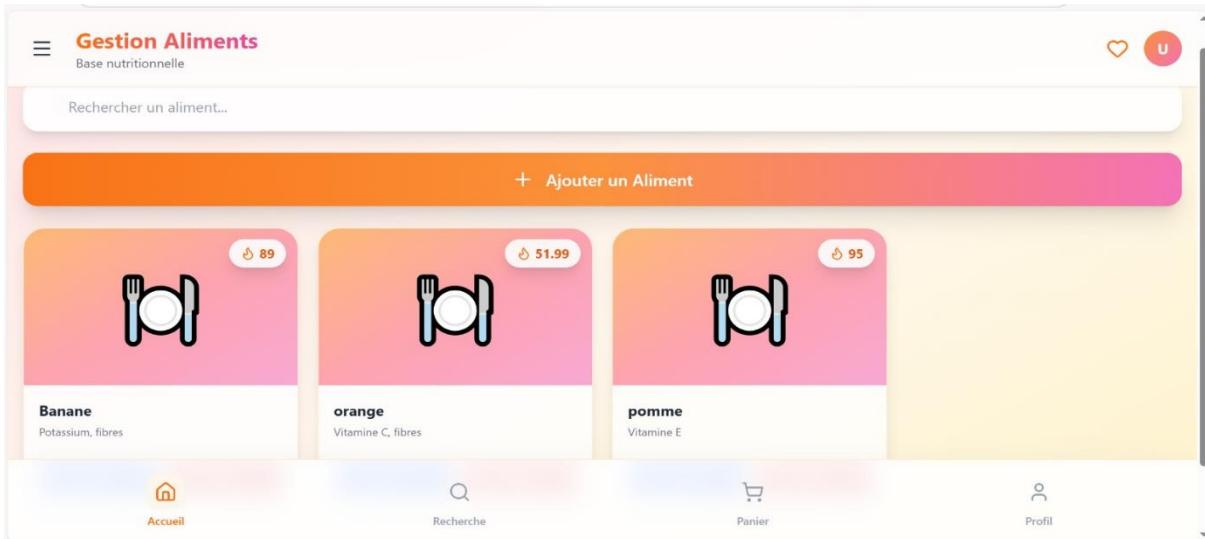


Figure 19 :interface Gestion Aliments

L'application permet de rechercher des aliments via une barre de recherche en haut, et d'en ajouter de nouveaux grâce au bouton proéminent "Ajouter un Aliment". Une barre de navigation en bas donne accès aux sections principales : Accueil, Recherche, Panier et Profil. L'interface privilégie une approche visuelle claire et intuitive pour faciliter le suivi nutritionnel.



5. Conclusion :

Ce chapitre a montré l'importance de l'environnement de travail, sans lequel il serait impossible de réaliser l'application. Nous avons également présenté quelques interfaces de l'application SuiviAlimentaire, en décrivant certaines fonctionnalités à travers des captures d'écran



Conclusion générale :

En guise de conclusion, ce projet de conception d'une application de suivi alimentaire assisté par OCR et IA nous a permis d'appliquer et de consolider nos connaissances en analyse des besoins, en conception logicielle orientée microservices et en modélisation UML.

L'objectif principal était de concevoir une solution innovante et automatisée permettant aux utilisateurs de suivre leur alimentation simplement en scannant un plat ou un ticket de caisse, sans saisie manuelle fastidieuse.

Nous avons défini une architecture modulaire basée sur des microservices Spring Boot, articulée autour de l'authentification sécurisée (JWT), du traitement d'image par OCR, de l'analyse nutritionnelle par IA, de la gestion du journal alimentaire et de la génération de recommandations personnalisées.

Les diagrammes de cas d'utilisation, de séquence et de classes réalisés illustrent le fonctionnement fluide et interactif entre l'utilisateur, l'application et les services backend.

Ce projet nous a fait parcourir les phases essentielles du cycle de développement logiciel : depuis l'analyse des besoins fonctionnels et non fonctionnels jusqu'à la conception détaillée des composants, en passant par la modélisation des interactions et la définition des entités métier.

Bien que l'application n'ait pas encore été implémentée et déployée, la conception proposée constitue une base solide et scalable, prête pour un développement futur.

Ce travail représente une expérience formatrice qui nous a permis d'approfondir nos compétences en conception système, de renforcer notre approche méthodologique et de mieux appréhender les défis techniques liés à l'intégration de l'IA et de l'OCR dans une application grand public.

Il ouvre également des perspectives d'évolution intéressantes, telles que l'amélioration des modèles d'IA, l'ajout de la reconnaissance visuelle directe, ou l'intégration avec des objets connectés et des bases de données nutritionnelles enrichies.