



# Crok'eco

## Projet de Programmation 2

BATATAY Mallory  
KEGLO Partice

Informatique  
Faculté des Sciences  
Université de Montpellier

2024 - 2025



### Résumé

Crok'eco est une application collaborative permettant à chacun de s'informer sur l'impact écologique d'un plat.

## Remerciements

Nous tenons à exprimer notre sincère gratitude envers toutes les personnes qui ont contribué à la réussite de ce projet. Leur soutien et leur encadrement ont été d'une importance capitale tout au long de la création de cette application.

Nous souhaitons exprimer nos sincères remerciements à M. Eric Bourreau pour nous avoir permis de travailler sur ce sujet. Sa confiance et son soutien nous ont permis de le réaliser avec succès. Nous lui sommes reconnaissants pour cette opportunité qui nous a permis d'acquérir une expérience en génie logiciel.

Nous souhaitons tout particulièrement remercier Ewen PHILIPOT, pour nous avoir aidés à initier le projet malgré qu'il soit parti au cours de cette année scolaire. Ses connaissances approfondies nous ont permis d'avoir une base solide pour l'application.

Je tiens également à exprimer ma gratitude envers Mme Elisabeth BAERT, responsable de la licence 3 Informatique, pour nous avoir permis de réaliser ces projets durant le second semestre.

Enfin, j'aimerais remercier chaleureusement toutes les personnes qui m'ont aidé dans la création de l'application ainsi que dans la rédaction de ce rapport et qui ont relu ce dernier. Leur soutien et leurs précieux commentaires ont grandement amélioré la qualité de ce travail.

## Table des matières

1	Introduction .....	4
2	Spécification du sujet .....	5
2.1	Enjeux climatiques .....	5
2.2	Approche du sujet .....	5
3	Gestion du projet .....	7
4	Choix technique .....	8
5	Architectures .....	9
5.1	Architectures de l'application .....	9
5.2	Modele Dynamique .....	10
6	Application .....	11
6.1	Base de données .....	11
6.1.1	Recherche .....	11
6.1.2	Modélisation .....	11
6.1.3	Implémentation .....	12
6.2	Application utilisateur .....	13
6.2.1	Analyse d'un menu .....	13
6.2.2	Historique .....	17
6.2.3	Recherche .....	17
6.2.4	Ajout de plat et vote .....	18
6.2.5	Design .....	18
6.3	Fonctionnalités non implémenté .....	18
6.4	Statistiques .....	19
7	Conclusion .....	20
8	Bibliographie .....	21
	Bibliographie .....	21

# 1 Introduction

Nous réalisons ce projet dans le cadre de notre troisième année en Licence Informatique. Le projet a débuté en décembre 2024 et nous a accompagnés tout au long de notre 2e semestre. Le sujet que nous avons choisi est celui de Monsieur Bourreau. Le but du projet est de créer une application permettant de noter l'impact écologique avec une couleur. Celle-ci peut être de couleur Verte, Orange ou Rouge, respectivement d'une empreinte carbone faible à élevée.

Ce projet a été réalisé avec KEGLO Patrice, BARATAY Mallory et PHILIPOT Ewen. PHILIPOT Ewen ayant arrêté la Licence avant les vacances de février, nous avons réalisé la majeure partie du projet à deux.

## 2 Spécification du sujet

### 2.1 Enjeux climatiques

Selon l'INSEE, en 2018, l'alimentation a représenté 22% de l'empreinte carbone de la France [1]. Que ce soit lié au transport de la nourriture, à l'élevage des animaux, à la quantité d'eau utilisée tout au long de la production, il y a beaucoup de facteurs polluants pour amener la nourriture dans nos assiettes. Une majorité de la population n'est pas informée sur ce sujet et ne pense pas à l'impact que sa nourriture a sur la planète.

Des solutions existent déjà pour réduire l'émission de gaz à effet de serre liée à notre alimentation. En effet, tous les aliments ne se valent pas en matière d'impact environnemental. Certains aliments tels que la viande et le poisson ont un impact écologique bien plus important que les fruits, légumes ou céréales. Le fait de consommer des produits de saison et locaux permet aussi de réduire l'émission due à notre alimentation.

Entre 2009 et 2019, la quantité de viande consommée a baissé de 5 % en France métropolitaine selon l'INSEE [2]. On constate donc une volonté en France de consommer plus écologiquement, et la demande concernant des applications permettant de se renseigner devient de plus en plus grande. On note également une augmentation de projets allant dans ce sens, que ce soit avec la nourriture mais également d'autres produits du quotidien comme les cosmétiques et les textiles, qui sont des domaines eux aussi très polluants.

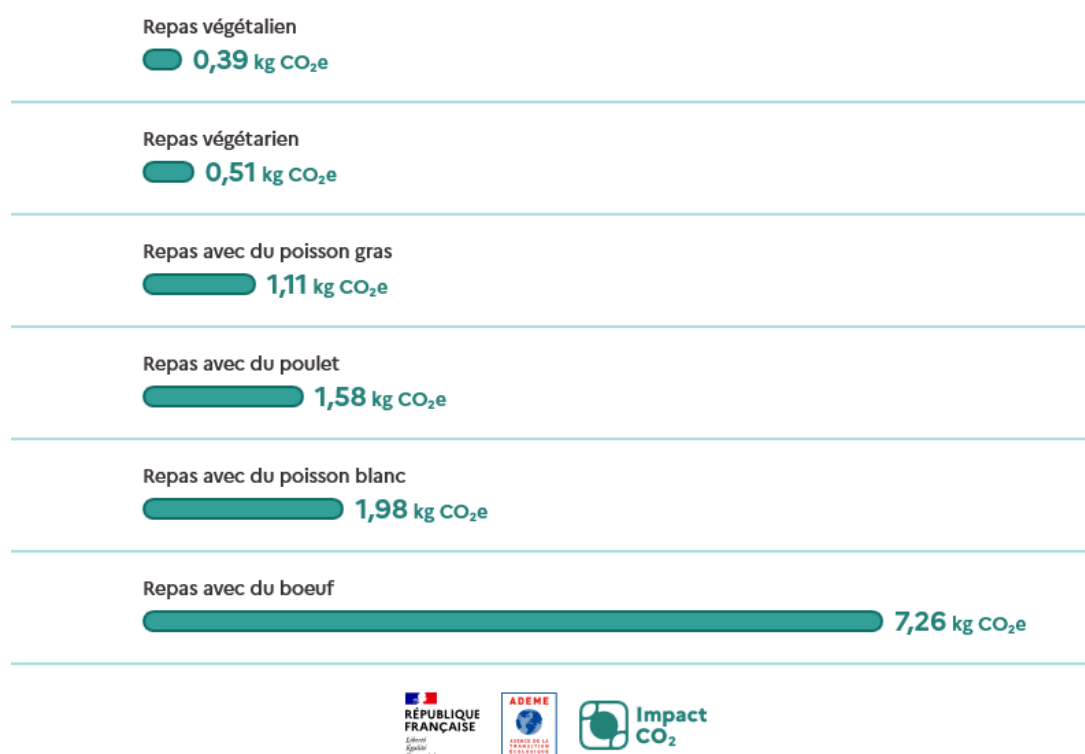


Fig. 1. – Emission de kg de CO<sub>2</sub> en fonction du type de repas

Source: ADEME [3]

### 2.2 Approche du sujet

Afin d'aider la population à faire des gestes écologiques, il est important de lui donner les outils permettant d'atteindre ses objectifs. La composition d'un plat étant la première cause de l'impact écologique du plat, il serait intéressant de pouvoir être informé de l'empreinte écologique du plat que l'on souhaite manger. Afin de pallier ce problème, nous avons imaginé plusieurs solutions :

Afin de proposer une solution aux étudiants, dans un premier temps, il serait possible de s'accorder avec le Crous ou les gérants des restaurants de la Faculté des Sciences pour faire afficher sur les téléviseurs l'impact écologique de chaque plat. L'affichage pourrait se faire à l'aide d'une pastille de couleur correspondant au degré de pollution.

Sinon, nous pourrions afficher le résultat à l'aide d'un QR Code. Le QR Code redirigerait vers une page qui détaillerait l'empreinte carbone de chaque plat.

Cette solution avait déjà été envisagée l'année dernière, mais sans succès, car, comme la première, elle présente un problème. En effet, elle nécessite un accord avec les gérants des restaurants qui sont assez réticents à voir l'impact écologique de leurs plats révélé. **De plus, même s'il avait été possible d'avoir l'accord pour les restaurants de la Faculté des Sciences, ces solutions demanderaient énormément d'efforts pour être déployées à l'échelle de la France, car elles nécessiteraient l'accord de chaque restaurant.**

Nous avons donc choisi de faire une application collaborative qui calcule l'impact écologique de tous les plats en scannant leurs noms sur un menu. Le but de cette dernière méthode est de permettre à tout le monde de participer à améliorer l'écologie en rendant l'application collaborative. Cette méthode peut fonctionner dans n'importe quel restaurant traditionnel et peut être étendue au self et cantine scolaire.

### 3 Gestion du projet

Le projet s'est déroulé en 4 grandes étapes, comme le décrit la Fig. 2 ci dessous, qui sont, la mise en place de la base de données, le développement des différentes fonctionnalités de l'application, la création du serveur distant ainsi qu'une base de données permettant de stocker et d'effectuer des mises à jour de données et enfin l'Implémentation complète et finition du design de l'application ainsi que les tests de fonctionnalités.

Globalement, le temps estimé pour chaque étapes du projet a été respecté selon nos prévisions, à part pour la base données que nous avons dû retravailler au cours du projet à cause du manque de pertinence des données, ce qui a nécessité une réadaptation du code de différentes parties de l'application.

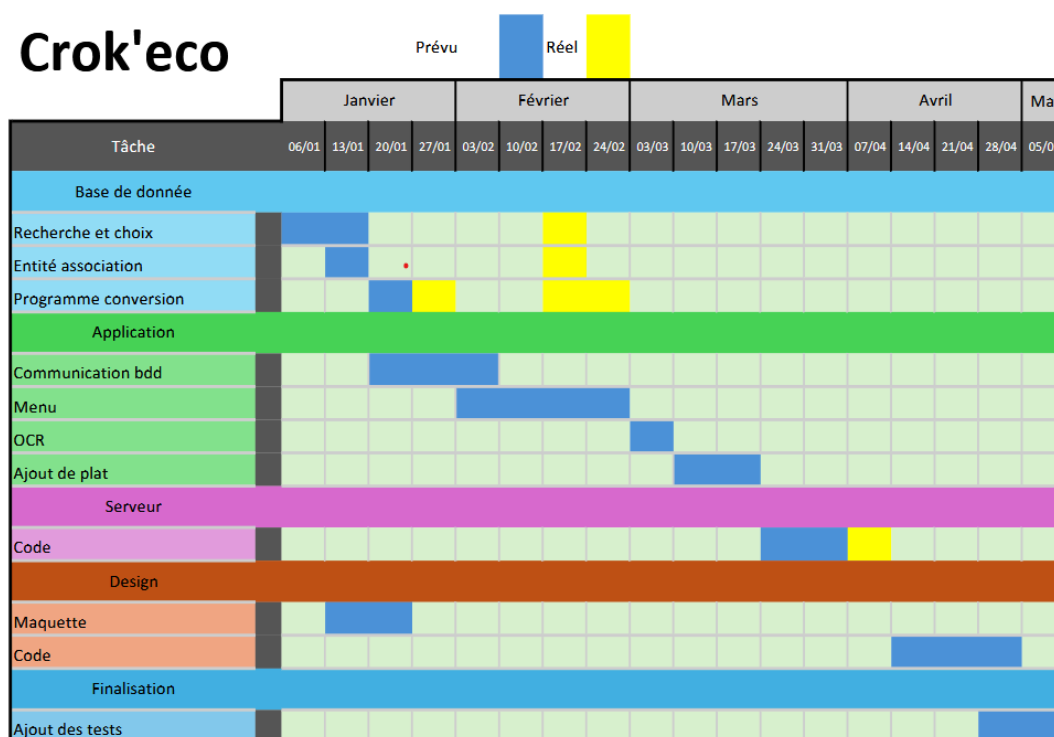


Fig. 2. – Diagramme de Gantt

## 4 Choix technique

Le développement de l'application a été réalisé avec Expo, un framework open-source basé sur React Native et TypeScript, un sur-ensemble typé de JavaScript, afin de bénéficier d'un typage statique, d'une meilleure lisibilité du code et d'une maintenance facilitée.

Le développement avec Expo nous permet d'avoir une application multi-plateformes (Android et iOS) et même en version web.

### Outils et bibliothèques principaux :

- Pandas(Python) pour la gestion de données en CSV
- Text-recognition de react-native-ml-kit , pour la reconnaissance de text sur une image
- Crypto-js pour encrypter les messages et assurer une sécurité dans le transfert de données entre notre serveur et l'application
- Netinfo de react-native-community pour mettre en place des mise à jour automatique de données entre l'application de notre serveur, en permettant de connaître l'état de l'utilisateur (Connecté ou non Connecté)
- fastify pour mettre en place une api pour notre backend.
- Yup qui permet de vérifier les formulaires
- Jest pour tester les différentes fonctionnalités de notre application
- ainsi que d'autres librairies d'expo et de react-native.

L'application a été testé à la fois sur simulateurs Android via Android Studio et sur appareils physiques pour s'assurer d'une bonne compatibilité et d'une expérience utilisateur fluide.

Nous avons utilisé GitHub comme plateforme de gestion de version tout au long du projet afin de collaborer efficacement, et d'avoir un historique clair des modifications. Les branches ont été utilisées pour séparer le développement des différentes fonctionnalités, ce qui a facilité l'intégration progressive des différentes parties de l'application dans la branche principale (Branche MAIN).

Les données que nous utilisons pour notre base de données proviennent toutes du programme de collecte de données AGRIBALYSE 2.0.



## 5 Architectures

### 5.1 Architectures de l'application

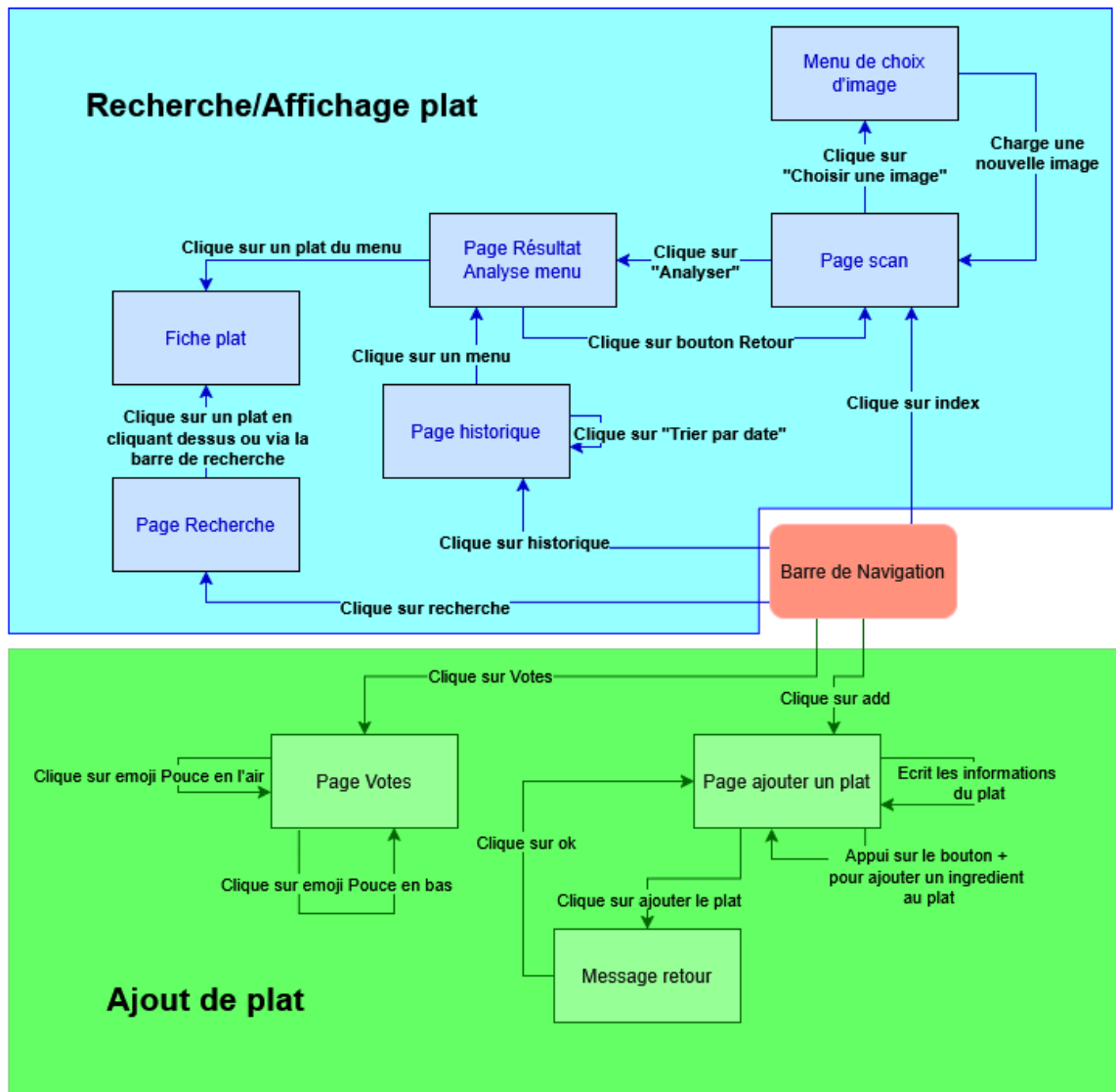


Fig. 3. – Architecture de l'application

La Fig. 3 illustre l'architecture de notre application du point de vue de l'utilisateur. L'application s'organise autour de la barre de navigation qui permet d'atteindre les 5 pages principales (Scan, Historique, Recherche, Votes, add). Ces 5 pages peuvent être redivisées en 2 catégories. Les pages permettant de rechercher et d'afficher la composition d'un plat et la partie permettant d'ajouter un plat à la base de données.

Dans la partie de recherche et d'affichage de plat on retrouve dans un premier temps la page scan(index). Cette page est la première qui s'ouvre lorsque l'on ouvre l'application. Elle a pour but de lancer l'analyse d'une image. Elle se compose de deux boutons. Le premier bouton, « Choisir une image », ouvre la médiathèque du téléphone pour que l'utilisateur sélectionne une image. Quand une image est choisie un menu s'ouvre pour proposer à l'utilisateur de modifier l'image en la rognant par exemple. Ensuite, l'image sélectionnée apparaît sur la page scan. L'utilisateur peut ensuite appuyer sur le bouton analyser.

## 5.2 Modele Dynamique

Lorsque l'utilisateur se sert de l'application, celle-ci communique avec le serveur afin de traiter les requêtes émises. Le diagramme de séquence (Fig. 4) ci-dessous illustre les échanges entre l'application et le serveur en fonction des actions réalisées par l'utilisateur.

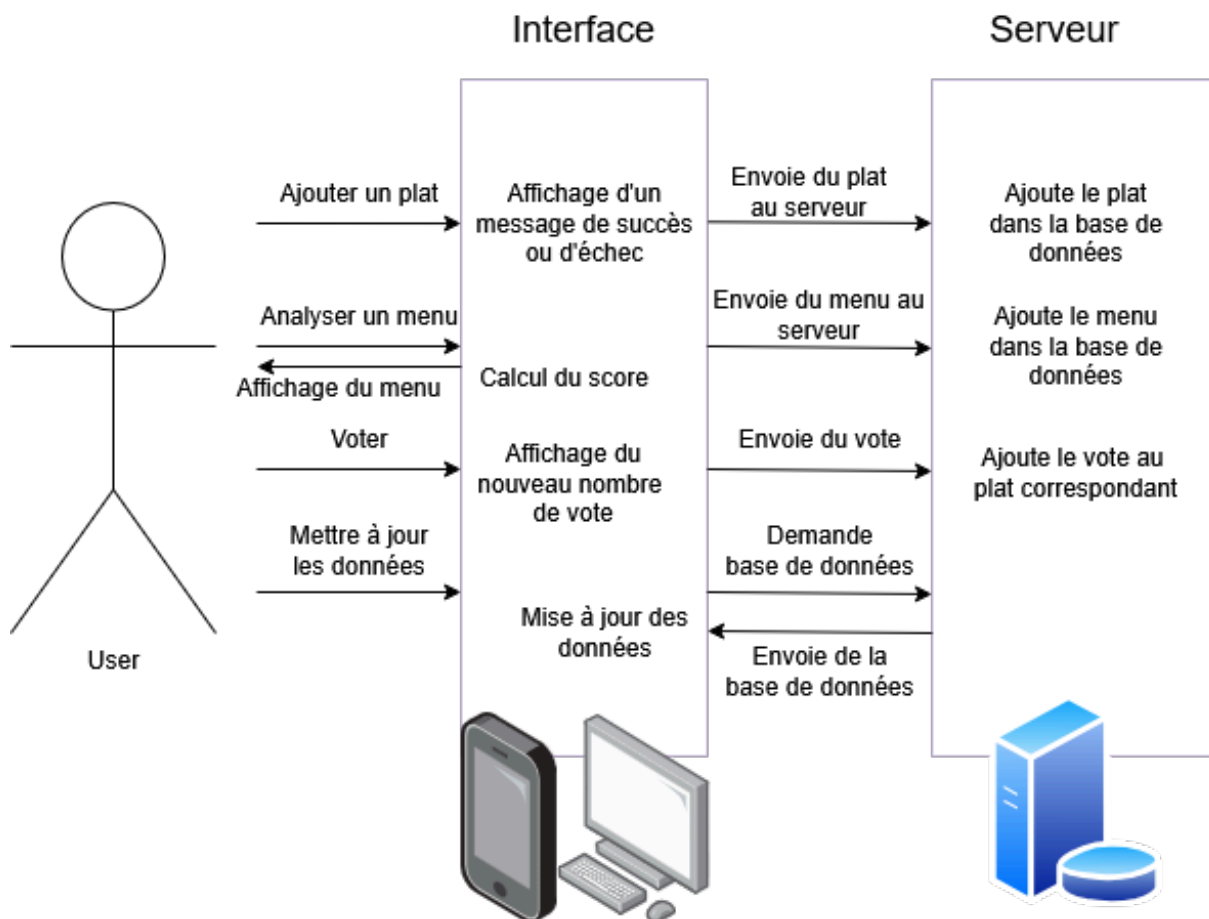


Fig. 4. – Diagramme de séquence

Les différentes requêtes présentes sur ce diagramme sont détaillées dans le Chapitre 6.2.

## 6 Application

### 6.1 Base de données

#### 6.1.1 Recherche

Afin de connaître l'impact écologique d'un plat nous avons choisi dans un premier temps de se servir de la base de données fournie par l'ADEME. La base de l'ADEME sur la consommation CO<sub>2</sub> est une immense base regroupant tout les types d'émission de gaz à effet de serre tel que celle du au textiles, à l'industrie ou autres, ainsi que toute les émissions liée à l'alimentation.



Fig. 5. – Logo AGRIBALYSE

En inspectant la base de données nous avons remarqué que toutes les informations liée à la nourriture provenait de deux bases de données qui sont AGRIBALYSE et AGRIBALYSE 2.0. AGRIBALYSE est un programme collectif et innovant qui met à disposition des données de référence sur les impacts environnementaux des produits agricoles et alimentaires à travers une base de données construite selon la méthodologie des Analyses du Cycle de Vie. Il est possible de se servir du site web d'AGRIBALYSE pour connaître l'impact environnemental d'un aliment ou bien de télécharger leur base de données.

Nous avons donc télécharger la base de données concernant dans un premier temps uniquement les plats ayant nécessité une transformation. Cette base de données était disponible au format CSV. Afin de traiter de rendre les données utilisable nous avons coder un programme python servant à initialiser une base de données en sql comportant tous les plats décrit dans le CSV.

Le CSV était construit de la manière suivante : par plat présent dans la base, il y avait une ligne pour chaque ingrédient. Cela signifie qu'on retrouve l'impact écologique d'un ingrédient pour que dans chaque plats comportant cette ingrédient, mais l'impact différé en fonction de la proportion de cette aliment dans le plat.

En analysant plus précisément nos besoins, nous avons remarqué un manque de plats trop important dans la base de données actuelle. Pour résoudre ce problème, nous avons changé une nouvelle fois de base de données, cette fois-ci comportant uniquement les ingrédients avec l'impact écologique associé par kilo d'aliment. Les données étaient fournies sous forme d'un fichier Excel comportant plusieurs onglets, dont un dédié uniquement aux ingrédients. Chaque ingrédient est associé vers un code Ciqua [4] qui sert d'identifiant unique.

Pour connaître le poids de chaque aliment dans un plat et pour remplir la table sql des plats nous avons choisi de faire confiance au utilisateur de l'application. Une page de l'application permet d'enregistrer la composition d'un plat (voir Chapitre 6.2.4). Connaissant le poids de chaque aliment et son impact écologique, il devient alors possible de calculer l'impact environnemental de chaque ingrédient au sein d'un plat, mais également l'impact écologique global du plat lui-même. Pour cela, on doit donc réalisé la somme de l'impact de chaque ingrédient dans le plat et faire un produit en croix pour le ramener à un kilogramme de nourriture.

#### 6.1.2 Modélisation

Pour l'application nous avons besoin d'une base de données regroupant d'une part les informations du CSV fournit par AGRIBALYSE ainsi que les nouvelles données tel que les plats ou les menus.

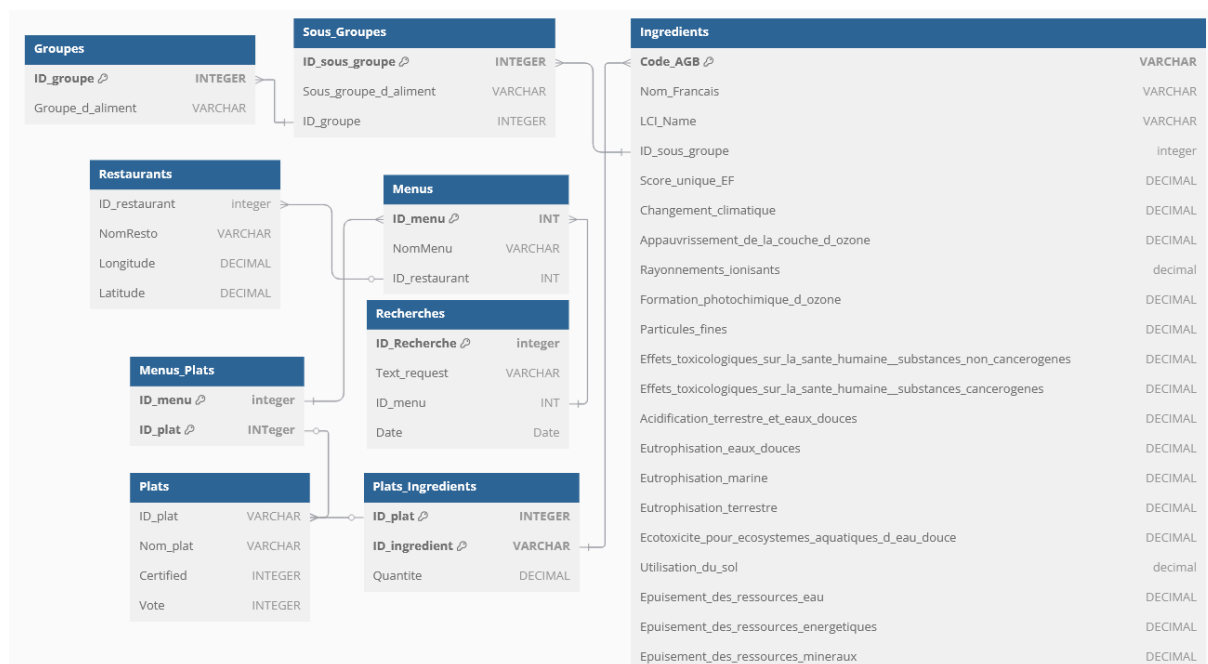


Fig. 6. – Schéma du modèle Entité Association de la base de données final

Nous avons donc défini un modèle Entité-Association (E-A) afin de représenter la structure de notre base de données. La Fig. 6 met en évidence les différentes tables que comporte la base ainsi que les liaisons entre elles.

Les données que nous récupérerons via le fichier Excel sont rangés dans la table ingrédients. Afin de trier les ingrédients par catégories on retrouve les tables groupes et sous-groupes. Chaque ingredient a une clé étrangère qui mène vers un sous groupes lequel possédant à son tour une clé étrangère menant vers la tables des groupes.

Ensuite, les plats sont composé de plusieurs ingrédients et un ingredient peut être présent dans plusieurs plat. On a donc une table intermediaire entre les plats et les ingrédients. Cette table est composé d'une clé étrangère menant vers l'ID des ingrédients

Après avoir modéliser les deux bases de données, on rédige leur structure dans un fichier .sql qui servira pour leurs implémentations (voir Chapitre 6.1.3).

### 6.1.3 Implémentation

Afin de convertir les données d'AGRIBALYSE dans notre base de données décrites dans la partie précédente, nous avons choisi de coder un programme Python. Le but du programme est de créer à partir de 0 la base de données.

Pour cela, on commence par créer le fichier qui contiendra la base de données. Ce fichier aura une extension .db et s'il était déjà existant, il est remplacé. Ensuite on importe le fichier .sql contenant l'agencement des tables décrit dans la partie Modélisation(voir Chapitre 6.1.2). Cela permet d'initialiser toutes les tables mais elles sont encore vides.

Ensuite, il faut remplir trois tables avec les données fournies par AGRIBALYSE. Les tables à remplir sont les groupes, les sous-groupes et les ingrédients. Les données sont dans un tableau Excel comportant plusieurs pages. On commence donc par extraire la bonne page du Excel. Après cela, on commence le remplissage des trois tables par celle des groupes à cause des contraintes de clé étrangère. Pour remplir la deuxième table, parce qu'on a besoin de la clé primaires de la tables des groupes qui viennent d'être ajouté à la base de données, on doit d'abord faire une requête à la base de données. On range les indices et le nom du groupe dans un dictionnaire. Avec les informations du tableau Excel, on associe

le noms des sous-groupes avec l'indice de son groupe correspondant. Enfin on ajoute le dictionnaire résultant à la base de données. On recommence ensuite la même opération afin d'associer les indices des sous-groupes avec les ingrédients.

Ce programme permet à la fois de créer la base de données pour l'application de l'utilisateur et pour le serveur. La seule différence entre les deux tables et qu'on utilise pas le même fichier .sql.

Avec ce programme Python, si la base de données d'AGRIBALYSE venait à être mise à jour, il suffirait d'exécuter le programme pour obtenir la nouvelle base. Une amélioration possible pour ce programme serait de récupérer les autres données des anciennes bases pour les rajouter aux nouvelles.

## 6.2 Application utilisateur

Dans cette partie, nous allons décrire et expliquer les différentes parties ainsi que les différentes fonctionnalités implémentées tout en montrant leur fonctionnement avec des images.

### 6.2.1 Analyse d'un menu

Cette fonctionnalité est visible sur la première page de notre application, elle consiste, dans un premier temps, à reconnaître le texte contenu sur une image choisie par l'utilisateur depuis sa galerie décrite par la figure Fig. 8. Pour cela l'utilisateur appuie sur le bouton **Choisir une image**. Par défaut, il y a une image d'un menu du restaurant administratif, (voir Fig. 7). Les images de sa galerie s'affichent puis il clique sur celle qu'il veut analyser. Si une fois les images de la galerie affichées, il (l'utilisateur) ne clique sur aucune image et referme l'affichage, un message d'alert s'affiche indiquant qu'aucune image n'a été sélectionnée (voir Fig. 9), sinon l'image est mise à jour (voir Fig. 10).

#### A revoir (update l'image)

L'utilisateur, peut rentrer le nom du restaurant dont il va analyser le menu ainsi que l'adresse s'il l'a. Ces deux informations ne sont pas obligatoires.



Fig. 7. – page scan

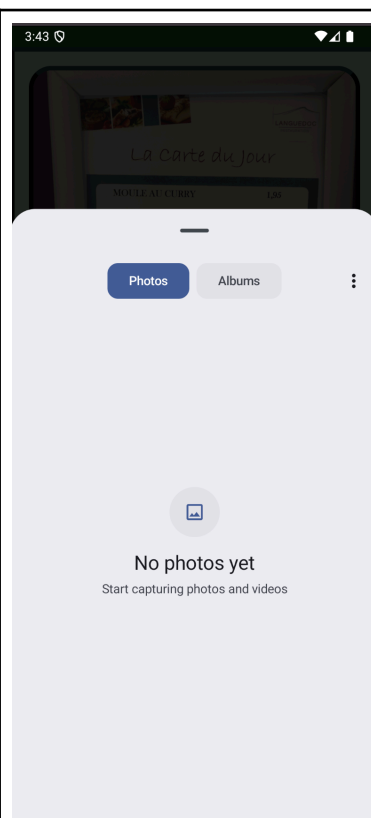


Fig. 8. – Choisir une photo

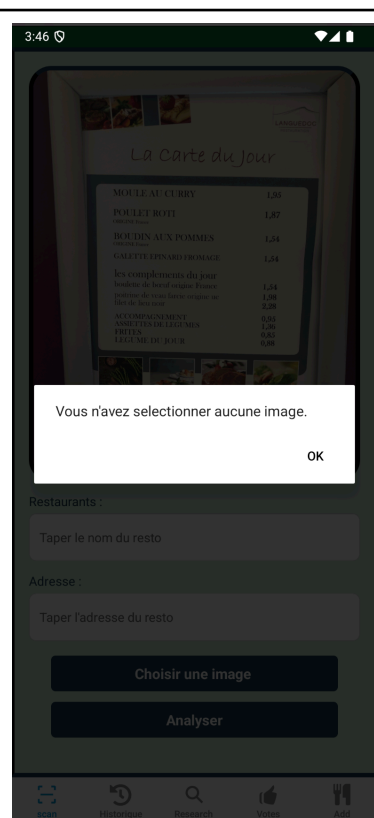


Fig. 9. – image non selectionné

Tableau 1. – Analyse menu

Une fois ceci fait, il peut cliquer sur le bouton **Analyser**. On effectue l'analyse textuelle de l'image et on recupere le nom des plats du menu ou du moins le texte reconnu. Une fois les noms recuperés, on verifie pour chaque plat, s'il existe dans notre base de données pour pouvoir ensuite recuperer la liste des ingredients, leurs quantités dans le plat et effectuer le calcul de score. En fonction du score du plat, on lui attribue une couleur qui sera la même que celle de la pastille qui sera affiche pour ce plat, comme l'indique le Tableau 2.

Le score d'un plat est la somme des Score unique EF [5] (EF = Environmental Footprint) = Somme (impact catégorie \* facteur de pondération) de chacun de ses ingredients.

Intervale de score	Couleur	Niveau d'Impact
score $\geq 0$ et score $\leq 1$	Verte	Faible
score $> 1$ et score $\leq 5$	Orange	Moyen
score $> 5$	Rouge	Elevé
Plat non reconnu	Noire	-

Tableau 2. – Tableau descriptif de l'attribution des couleurs

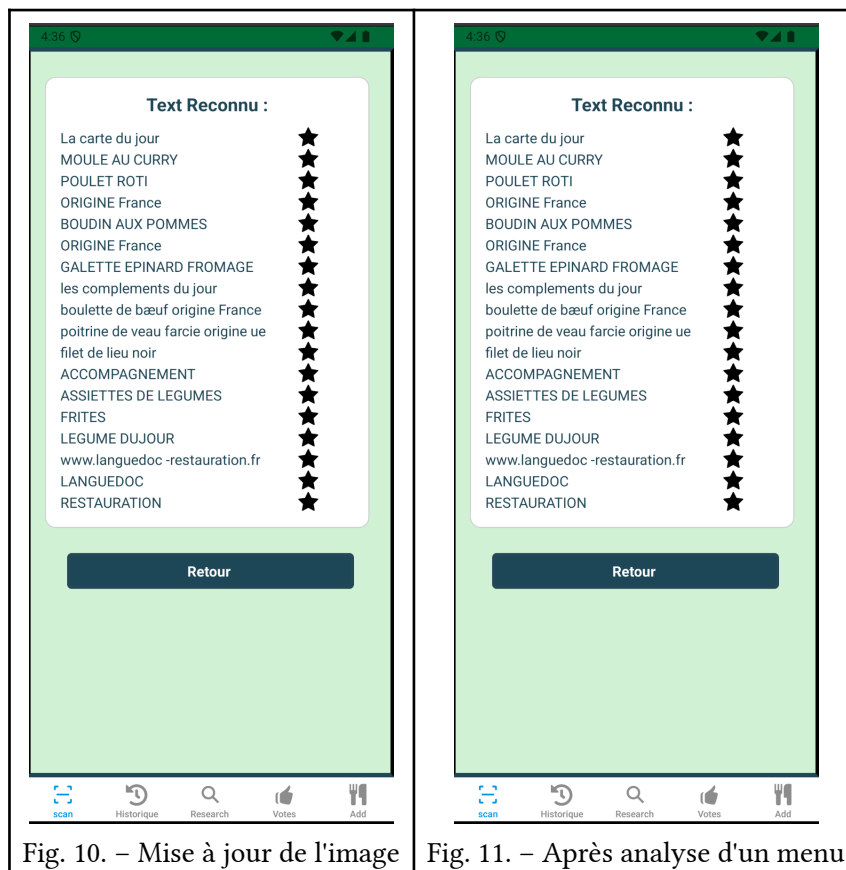


Fig. 10. – Mise à jour de l'image

Fig. 11. – Après analyse d'un menu

Tableau 3. – Update image et resultat de l'analyse

Avant l’affichage du texte ou des plats reconnu(s), les informations renseignées sont stockées dans la base de données locale de l’application. Ces données sont aussi envoyées au serveur si l’utilisateur est connecté à internet et donc au serveur ou attend qu’il le soit pour les envoyées. Cette fonctionnalité est implementée par l’algorithme dans la partie **algos**.

Après l’analyse, on affiche la liste des plats avec une pastille en forme d’étoile juste devant le nom ayant une couleur descriptif de l’impact ecologique du plat comme l’indique la Fig. 11. Quand un plat n’existe pas dans notre base de données, l’étoile est de couleur noire.

On peut cliquer sur chaque ligne ou plat. Et si le plat existe dans notre base de données, on est reconduit sur une autre page affichant les informations du plat ainsi que les différents ingrédients qui le composent avec leur pourcentage comme décrit la Fig. 12. Si Le plat n’existe pas, alors le message : « Ce plat n’existe pas dans nos données. Vous pouvez l’ajouter en allant sur la page d’ajout » dont nous parlons plus tard dans la partie Chapitre 6.2.4, est affiché (voir Fig. 13).

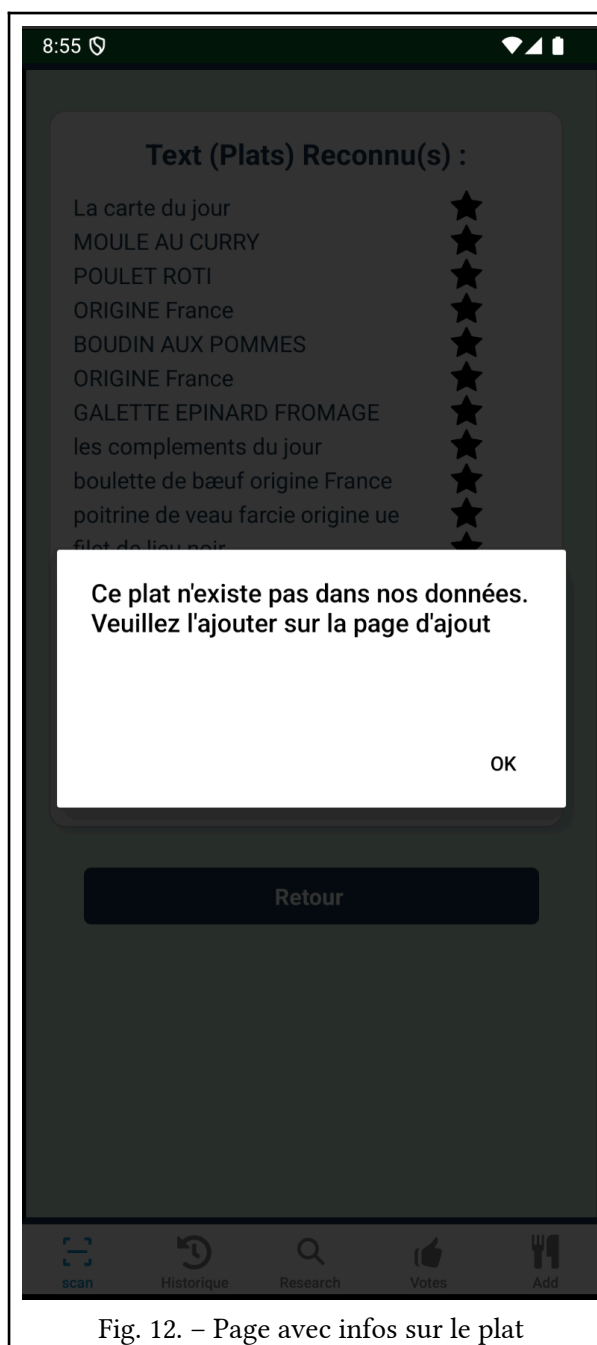


Fig. 12. – Page avec infos sur le plat

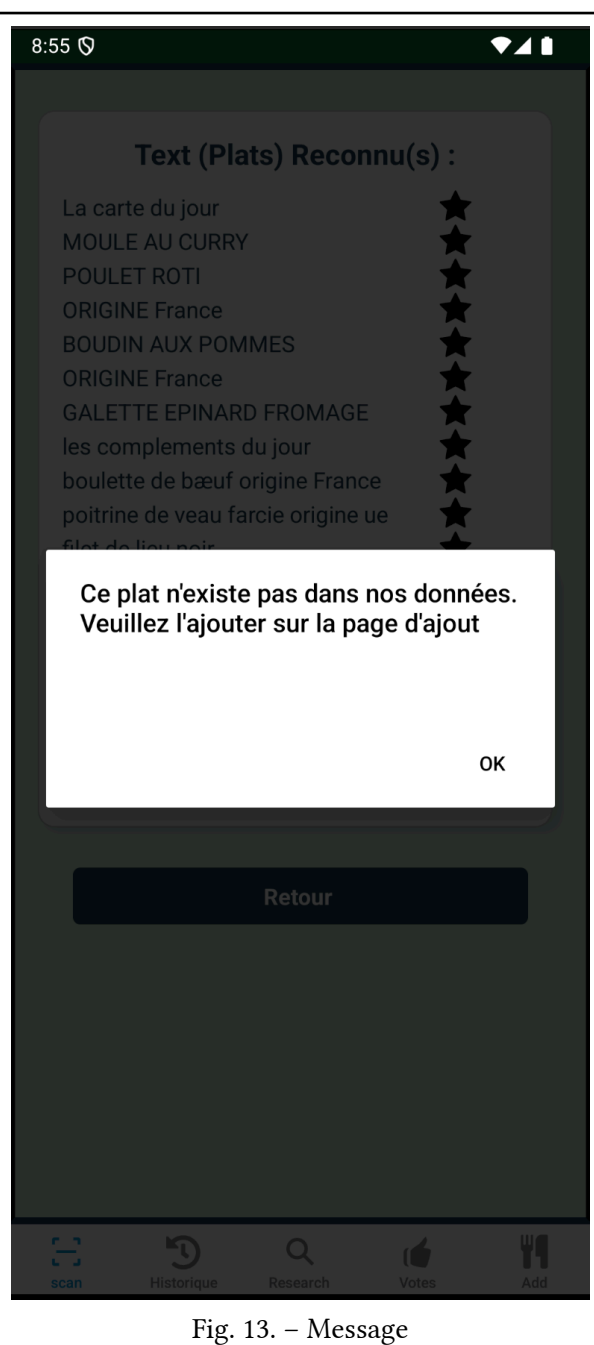


Fig. 13. – Message

Tableau 4. – Page de details d'un plat et alert si le plat n'existe pas dans la base de données



### 6.2.2 Historique

Sur cette page, on retrouve tous les menus que l'utilisateur a scanné. Pour cela on recupere stockées dans la base de données locale de l'application (voir Fig. 14). On peut aussi les trier par date, du plus anciens au plus recents ou et vice versa (voir Fig. 16 et Fig. 15). En cliquant sur un menu, on est redirigé vers la page de détails du plat (voir Fig. 12).

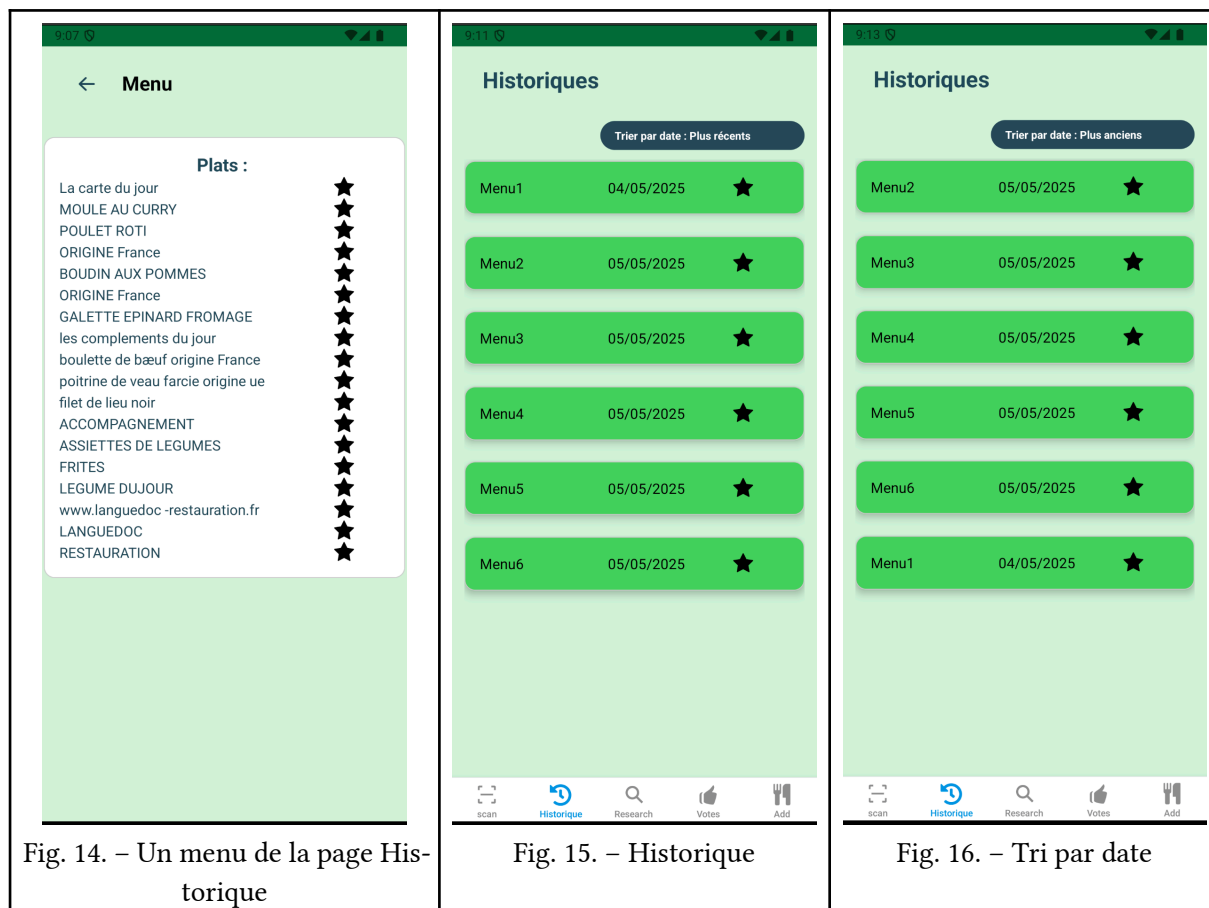


Tableau 5. – Historique des plats scannés

### 6.2.3 Recherche

Cette page est constuié de 3 parties.

La première partie est un champ de recherche qui permet de rechercher un plat dans la base de données. Il suffit de commencer par taper le nom du plat et une liste déroulante s'affiche avec les plats existant dans notre base de données et qui correspondent à la recherche (voir Fig. 17). En cliquant sur un plat, on est redirigé vers la page de détails du plat (voir Fig. 12).

La deuxième partie (Favoris) est une liste de plats existant dans notre base de données et qui sont les plus recherchés par les utilisateurs (voir Fig. 18). En cliquant sur un plat, on est redirigé vers la page de détails du plat (voir Fig. 12).

La troisième partie (Suggestions) est une liste de plats existant dans notre base de donnée et qui sont les moins polluants (voir Fig. 18). En cliquant sur un plat, on est redirigé vers la page de détails du plat (voir Fig. 12).

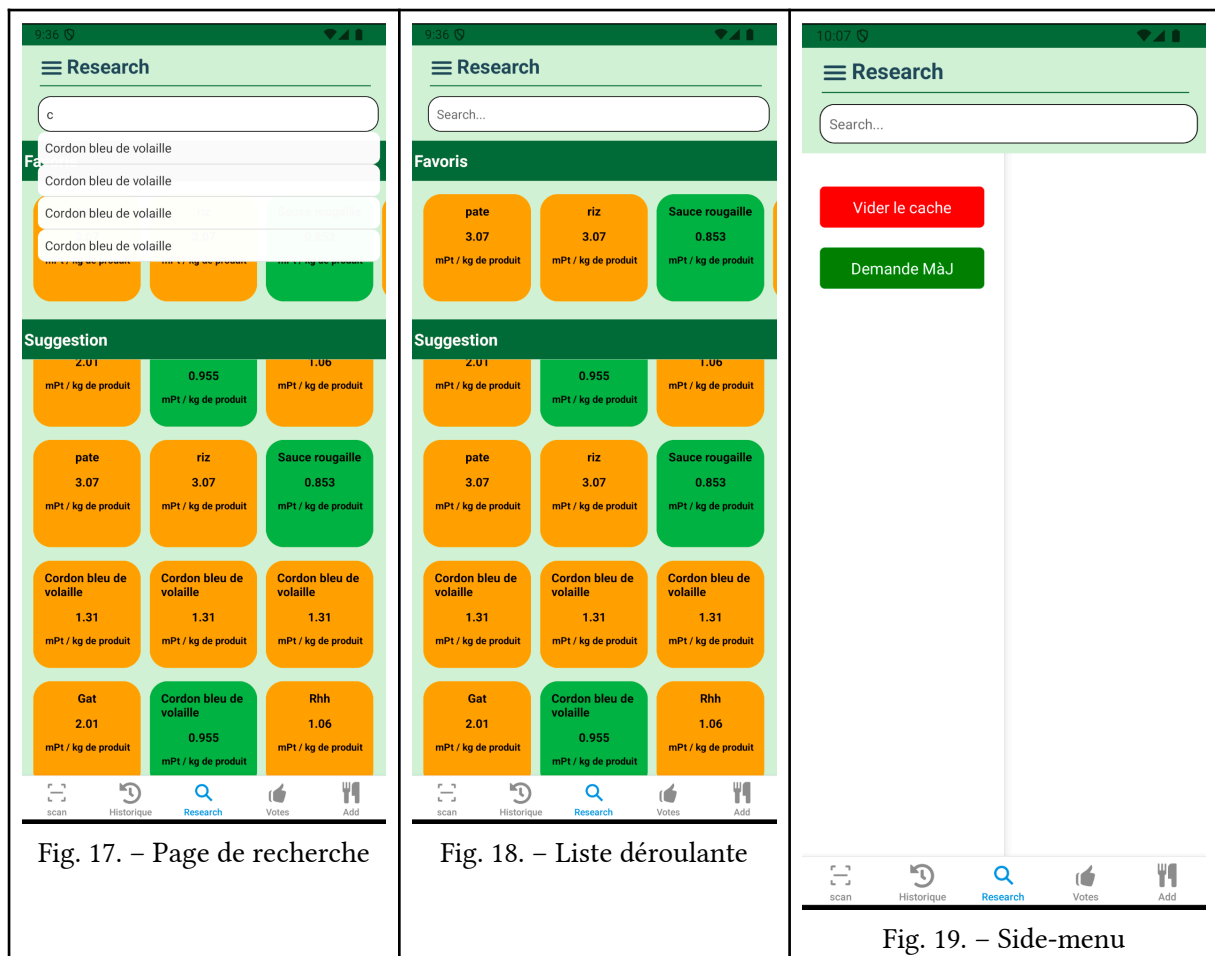


Tableau 6. – Page de recherche et liste déroulante et side-menu

On a aussi tout en haut à gauche de la page, un bouton de menu qui permet d'afficher un side-menu. Ce menu permet d'avoir accès deux fonctionnalités de l'application. La première est la possibilité de vider le cache de l'application grace au bouton **Vider le cache**. En effet, l'application stocke les données permettant de faire l'affiche de cette page, dans le cache de l'appareil, pour permettre la fluidité de l'affichage et d'éviter de nombreuses requetes à la base de données. Il est donc possible de vider le cache pour libérer de la place sur l'appareil. La deuxième fonctionnalité est la possibilité de faire une demande de mise à jour des données de l'application manuellement grace à un bouton **Demande de MàJ**, même si la mise à jour se fait automatiquement une fois par jour, lorsque l'utilisateur est connecté à internet. En cliquant sur ce bouton, on envoie une requête au serveur pour lui demander de mettre à jour les données de l'application. Le serveur va alors envoyer une réponse à l'application avec les données qui manquent s'il y en a. Sinon, juste une réponse pour dire que tout est à jour. Cette fonctionnalité est implémentée par l'algorithme dans la partie **algos**.

#### 6.2.4 Ajout de plat et vote

#### 6.2.5 Design

Nous avons commencé par faire des croquis de l'application sur figma afin de réfléchir à quelle fonctionnalité nous allions implémenté. ...

### 6.3 Fonctionnalités non implémenté

Malgré que l'application soit fonctionnel, plusieurs fonctionnalités non pas pu être implémenté. Ces fonctionnalités non pas été implémenté par manque de temps, ou par lacune technique. Nous allons

présenter les pistes les plus intéressantes sur lesquels nous pourrions travailler pour une future version de l'application.

Dans la première page du menu, nous avons envisagé que l'utilisateur puisse choisir entre prendre une photo directement dans l'application ou la sélectionner dans la galerie de son appareil.

Ensuite, il serait vraiment intéressant que les utilisateurs puissent se créer un compte. Cette ajout serait très important avant la publication de l'application pour pouvoir modérer les ajouts de plat ou pour limiter le nombre de vote par plat par utilisateur. Les autres fonctionnalités seraient toujours accessibles sans se créer de compte.

Caméra Localisation Compte user Amélioration des suggestions

## **6.4 Statistiques**

## 7 Conclusion

Pour conclure, l'application Crok'eco répond efficacement aux objectifs fixés en permettant aux utilisateurs de s'informer sur l'impact écologique des plats qu'ils consomment. Grâce à son approche collaborative et à l'utilisation de bases de données fiables comme AGRIBALYSE, elle offre une solution innovante et accessible pour sensibiliser le public à l'empreinte carbone de leur alimentation.

Cependant, certaines fonctionnalités prévues n'ont pas pu être implémentées, comme l'intégration de la caméra ou la gestion des comptes utilisateurs. Ces éléments représentent des pistes d'amélioration pour les futures versions de l'application. Malgré ces limitations, le projet a permis de développer des compétences techniques variées, allant de la gestion de bases de données à l'utilisation de frameworks modernes comme Expo.

Enfin, ce projet nous a permis de mettre en pratique nos connaissances acquises en Licence Informatique mais aussi de développer de nouvelles compétences essentielles pour la suite de notre parcours.

## 8 Bibliographie

### Bibliographie

- [1] INSEE, « Empreinte carbone de l'alimentation ».. <https://www.insee.fr/fr/statistiques/7728883?sommaire=7728903>
- [2] INSEE, « Évolution de la consommation des ménages selon les territoires de 2009 à 2019 ».. <https://www.insee.fr/fr/statistiques/7728873?sommaire=7728903>
- [3] ADEME, « Base Empreinte ».. <https://base-empreinte.ademe.fr/>
- [4] CIQUAL, « Table de composition nutritionnelle des aliments ».. <https://ciqual.anses.fr/>
- [5] Commission, E., « Environmental Footprint Methods ».. <https://ec.europa.eu/environment/eussd/smgp/index.htm>

### Tables des figures

Fig. 1: Emission de kg de CO2 en fonction du type de repas

Source: ADEME [3] .....	5
Fig. 2: Diagramme de Gantt .....	7
Fig. 3: Architecture de l'application .....	9
Fig. 4: Diagramme de séquence .....	10
Fig. 5: Logo AGRIBALYSE .....	11
Fig. 6: Schéma du modèle Entité Association de la base de données final .....	12
Tableau 1: Analyse menu .....	14
Fig. 7: page scan .....	14
Fig. 8: Choisir une photo .....	14
Fig. 9: image non sélectionné .....	14
Tableau 2: Tableau descriptif de l'attribution des couleurs .....	14
Tableau 3: Update image et resultat de l'analyse .....	15
Fig. 10: Mise à jour de l'image .....	15
Fig. 11: Après analyse d'un menu .....	15
Tableau 4: Page de details d'un plat et alert si le plat n'existe pas dans la base de données .....	16
Fig. 12: Page avec infos sur le plat .....	16
Fig. 13: Message .....	16
Tableau 5: Historique des plats scannés .....	17
Fig. 14: Un menu de la page Historique .....	17
Fig. 15: Historique .....	17
Fig. 16: Tri par date .....	17
Tableau 6: Page de recherche et liste déroulante et side-menu .....	18
Fig. 17: Page de recherche .....	18
Fig. 18: Liste déroulante .....	18
Fig. 19: Side-menu .....	18

enlever le numero de la page d'entete

CORRIGER L'ORTHOGRAPHE

si on en a besoin<sup>1</sup>

partie a finir :

1. statistiques
2. fonctionnalités non implémenté
3. design
4. recherche
5. ajout de plat et vote

---

<sup>1</sup>test