



MENU & ALLERGEN INFORMATION SYSTEM

Réalisé par :
Mathéo Henault
Perle M.S. Ndayizeye
Ismael Madou Gagi
Dimitri Delpech

Table des matières

2. Objectifs et besoins du client.....	4
2.1. Sources et structure des données.....	4
1.1. Présentation des données.....	4
1.2. Contenu de la base de données.....	5
3.1. Structure.....	5
3.2. Nettoyage de la base de données.....	6
• Nettoyage et préparation des données (cleaning_processing.py).....	6
3.3. Parcours Statistique.....	7
Analyse descriptive (descriptive analysis.py).....	7
3.3.1. Analyse nutritionnelle.....	8
1. Teneur en calcium.....	9
2. Teneur en fer.....	10
3. Teneur en protéines.....	10
4. Teneur en glucides.....	11
5. Teneur en lipides.....	11
6. Teneur en acides gras saturés.....	12
3.3.2. Conclusion de l'analyse des nutriments présents dans les plats.....	12
4. Architecture technique.....	13
4.1. Scripts et fonctionnalités.....	13
4.2. Structure logique.....	16
7.1. Points Clés :	20
7.2. Limites.....	20

I. Introduction

Dans un monde où l'expérience client devient un facteur clé de différenciation, le secteur de la restauration n'échappe pas à la règle. Aujourd'hui, les consommateurs ne cherchent plus seulement à bien manger : ils veulent des repas en accord avec leurs goûts, leurs convictions et leurs besoins spécifiques (allergies, régimes, préférences alimentaires).

C'est dans cette optique qu'un restaurateur souhaite transformer sa manière de servir ses clients. Son ambition ? Offrir une expérience culinaire sur-mesure grâce à une approche data-driven, innovante et interactive qui va également lui permettre de récolter des données sur le long terme.

Notre mission:

Nous avons été chargés de développer une interface intelligente capable de :

- Analyser en profondeur les caractéristiques des plats : ingrédients, allergènes, valeurs nutritionnelles.
- Proposer des recommandations personnalisées en temps réel pour chaque client.
- Optimiser la gestion des stocks et des préférences afin d'aligner efficacité opérationnelle et satisfaction client.

Grâce à cette solution, le restaurateur pourra non seulement fidéliser sa clientèle, mais aussi se positionner comme un acteur moderne et attentif aux besoins de chacun. Une nouvelle manière de consommer, plus saine, plus personnalisée... et plus humaine.

2. Objectifs et besoins du client

- Analyse des menus pour mettre en évidence les préférences des clients
- Développement d'un algorithme de recommandation suivant les restrictions et préférences
- Concevoir un modèle permettant de filtrer les plats en fonction des contraintes alimentaires.
- Proposer des représentations graphiques claires pour faciliter l'interprétation des données.
- Générer des rapports et des indicateurs exploitables pour améliorer la gestion des menus.
- Assurer une interface intuitive et accessible pour le restaurateur.
- Catégorisation selon les labels (fait maison, biologique, etc.)
- Les clients ont la possibilité de commander des plats avec certains ingrédients en moins selon leurs préférences.

Afin de répondre à ces objectifs, il faut tout d'abord bien comprendre les données à disposition. C'est pourquoi nous présentons ci-après les sources, la structure et le contenu des bases de données exploitées dans ce projet.

II. Description des données

Notre base de données est constituée de deux ensembles de données distincts :

- Les données du menu, qui répertorient les plats proposés par le restaurant.
- Les données des plats, qui contiennent des informations détaillées sur chaque plat, notamment leurs ingrédients.
- Nous avons ensuite défini les variables des deux tables constituant notre base de données. Ces informations ont été consignées dans le document "**Description des données.docx**" afin de garantir une compréhension claire et partagée de la structure de notre base.

2.1. Sources et structure des données

1.1. Présentation des données

Les données utilisées dans ce projet proviennent des menus officiels proposés par la ville d'Antibes Juan-les-Pins dans le cadre de sa restauration collective. Ces informations,

régulièrement actualisées, offrent une vision détaillée et transparente de l'offre culinaire municipale.

- **Licence** : Open Licence 2.0 (réutilisation libre avec mention de la source).
- **Mise à jour** : toutes les deux semaines.
- **Source** : « Ville d'Antibes, disponible sur data.gouv.fr, mise à jour du 31 janvier 2025».

1.2. Contenu de la base de données

Les jeux de données incluent :

1. **Les plats proposés** :
 - Entrées, plats principaux, accompagnements, desserts, pains et suppléments.
 - Classés par type de menu (scolaire, senior, etc.).
2. **Caractéristiques descriptives** :
 - **Nutriments** : Teneur en protéines, glucides, lipides, calcium, fer, etc.
 - **Allergènes** : Liste des produits allergènes (gluten, lait, œufs, etc.) pour chaque plat.
3. **Origine des produits** : indication de la provenance (local, bio, etc.).
4. **Métadonnées** :
 - Dates de validité des menus.
 - Identifiants uniques pour chaque plat.

Une fois les données identifiées et comprises, nous avons défini une méthodologie pour assurer un traitement fiable et une implémentation efficace.

III. Méthodologie

Pour assurer une collaboration fluide entre les membres de l'équipe, nous avons mis en place un dépôt Git accessible à l'ensemble des collaborateurs.

De plus, nous avons élaboré une roadmap afin de définir clairement les différentes missions à réaliser ainsi que les délais associés, permettant ainsi une meilleure organisation et gestion du projet.

3.1. Structure

Nous avons apporté des modifications aux noms des colonnes de la table afin de les uniformiser en anglais. De plus, nous avons supprimé les accents et les tirets, qui n'existent pas en anglais, afin d'assurer une meilleure compatibilité des variables. La version modifiée de cette base de données a été enregistrée dans le fichier data/processed.

Des modifications ont été apportées à la structure du dépôt git afin de centraliser l'exécution dans un fichier **main.py**. Cette approche permet d'organiser le déroulement du programme de manière séquentielle, en exécutant chaque étape une à une, garantissant ainsi une meilleure structuration et lisibilité du code.

Nous avons développé plusieurs scripts Python qui s'articulent autour de quatre grands axes :

- le traitement et l'analyse des données,
- l'interaction avec l'utilisateur,
- la validation et l'affichage des résultats.
- les tests unitaires pour assurer la fiabilité du système.

Cette organisation permet d'optimiser la gestion des repas en garantissant une expérience fluide et adaptée aux préférences et restrictions alimentaires des utilisateurs.

3.2. Nettoyage de la base de données

Avant toute interaction avec l'utilisateur, il est essentiel de disposer d'une base de données propre et exploitable. Cette phase comprend le nettoyage des données et l'analyse statistique pour mieux comprendre leur structure.

• Nettoyage et préparation des données (**cleaning_processing.py**)

Fonctionnalités :

1. **Importation des données** : Chargez deux fichiers Excel (antibes-menu-2025.xlsx et antibes-plats-2025.xlsx) depuis le dossier data/raw.

2. **Nettoyage des données** :

➤ **Menu** :

- Sélectionne des colonnes spécifiques et renomme les variables pour une meilleure lisibilité.
- Gère les valeurs manquantes (NaN) en les remplaçant par des valeurs par défaut ("Non renseigné" ou "Aucun").
- Uniformise les noms de colonnes (ex: menuRestaurantId → menurestaurant_id).

➤ **Plats** :

- Supprimer des colonnes inutiles (platProduitIonisation, platProduitAdditif).
- Nettoyer les valeurs manquantes pour les allergènes et nutriments.
- Renommer les colonnes (ex: platCode → dish_code).

3. **Exportation** : Sauvegarde les données nettoyées dans le dossier data/processed sous forme de fichiers CSV (menu.xlsx, dishes.xlsx).

Utilité :

Prépare les données brutes pour l'analyse en assurant leur cohérence et leur qualité.

Modification des données(modifdata)

Fonctionnalités :

1. Extraction des nutriments :

- Par la colonne product nutrient (chaîne de texte) pour extraire des valeurs numériques (ex: "1.2g Protides → 1.2).
- Convertit les unités (mg → g) et crée des colonnes séparées pour chaque nutriment (Protéines, Glucides, etc.).

2. Filtrage des données :

- Conserve uniquement les colonnes pertinentes (A à E) et ajoute les nutriments extraits.
- Remplace les valeurs manquantes par 0.

3. Exportation : Génère un fichier dishes cleaned.xlsx que l'on a nommée dishes afin de l'utiliser dans le code il se trouve dans le dossier data/processed.

Utilité :

Transforme les données nutritionnelles textuelles en format structuré pour l'analyse quantitative

3.3. Parcours Statistique

Voici comment notre travail était :

Analyse descriptive (descriptive analysis.py)

Fonctionnalités :

1. Analyse exploratoire :

- Compte les allergènes et nutriments par plat.
- Génère des visualisations :
- **Fréquence des plats** : barplot avec légende pour mapper les noms à des IDs.
- **Allergènes par plat** : barplot simplifié avec numérotation.
- Tables croisées (pd.crosstab) pour explorer les relations entre variables (ex: meal_type vs dish_type).

2. Optimisation visuelle :

- Utilisation de seaborn pour des graphiques clairs.
- Adaptation des légendes pour éviter la surcharge d'information.

Utilité : Identifie les tendances (ex: plats/allergènes fréquents) et facilite la prise de décision.

• Visualisations et interprétations statistiques

L'étape suivante consiste à intégrer l'utilisateur au cœur du système. L'interaction utilisateur est un élément clé pour garantir une expérience personnalisée et intuitive.

L'utilisateur interagit avec le programme en renseignant ses allergies et en sélectionnant son repas. Cette phase repose sur plusieurs modules qui assurent une expérience fluide et personnalisée.

3.3.1. Analyse nutritionnelle

1. Accès aux statistiques

Depuis le menu principal, l'utilisateur peut choisir l'option "**stats**" afin de consulter des informations nutritionnelles spécifiques.

2. Choix des éléments à analyser

L'utilisateur est invité à sélectionner les éléments nutritionnels qu'il souhaite consulter. Les options disponibles sont : **Calcium, Fer, protéines, Glucides, Lipides, Acides gras saturés**. Il peut en choisir un ou plusieurs.

3. Affichage des résultats

Une fois la sélection effectuée, les statistiques correspondantes sont affichées à l'utilisateur.

4. Choix de continuer

Après avoir visualisé les statistiques :

- L'utilisateur peut **consulter d'autres données nutritionnelles** en revenant à l'étape précédente.
- Ou bien **revenir au menu principal** pour poursuivre sa commande ou quitter l'application.

Une fois les choix de l'utilisateur effectués, nous avons mis en place des analyses nutritionnelles pour évaluer la qualité des plats proposés et fournir des recommandations adaptées.

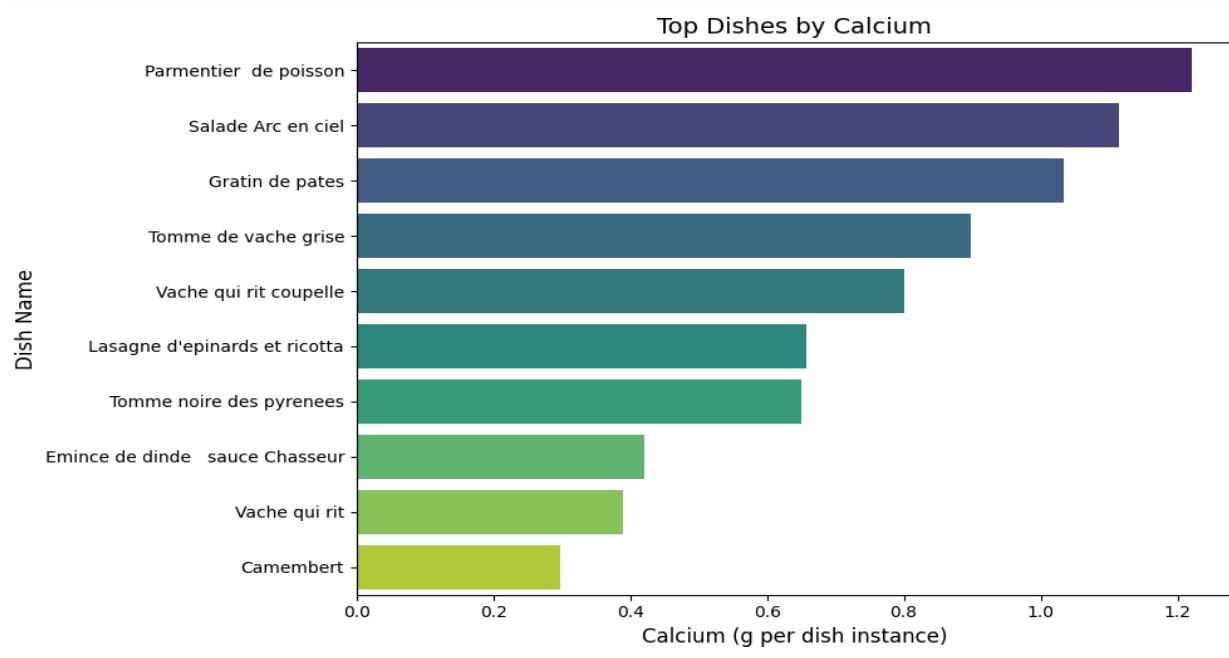
L'objectif de cette analyse est d'examiner la répartition des nutriments (calcium, fer, protéines, glucides, lipides et lipides saturés) dans différents plats. En identifiant les plats les plus riches en chaque nutriment, nous pouvons mieux comprendre leur contribution nutritionnelle et leur impact potentiel sur une alimentation équilibrée.

Nous avons réalisé plusieurs visualisations qui mettent en évidence les plats les plus riches pour chaque aliment. Voici une analyse détaillée de chaque graphique.

⚠ Note méthodologique : Afin de garantir une analyse plus fiable, nous avons filtré certaines valeurs nutritionnelles jugées incohérentes. Par exemple, certaines entrées indiquent 45 g de poivre ou plus de 100g de lipides pour un seul **ingrédient** de plat, ce qui n'est pas réaliste. Des ingrédients non pertinents pour l'analyse (poivre, sel, eau, thym, pâte à tartiner, arômes...) ont été exclus, et des seuils ont été appliqués (ex : max 30g de protéines, 1.5g de calcium...). Malgré cela, la qualité des données dépend encore de la précision des valeurs renseignées dans la base de données. Un travail complémentaire reste donc nécessaire pour fiabiliser les quantités nutritionnelles des ingrédients.

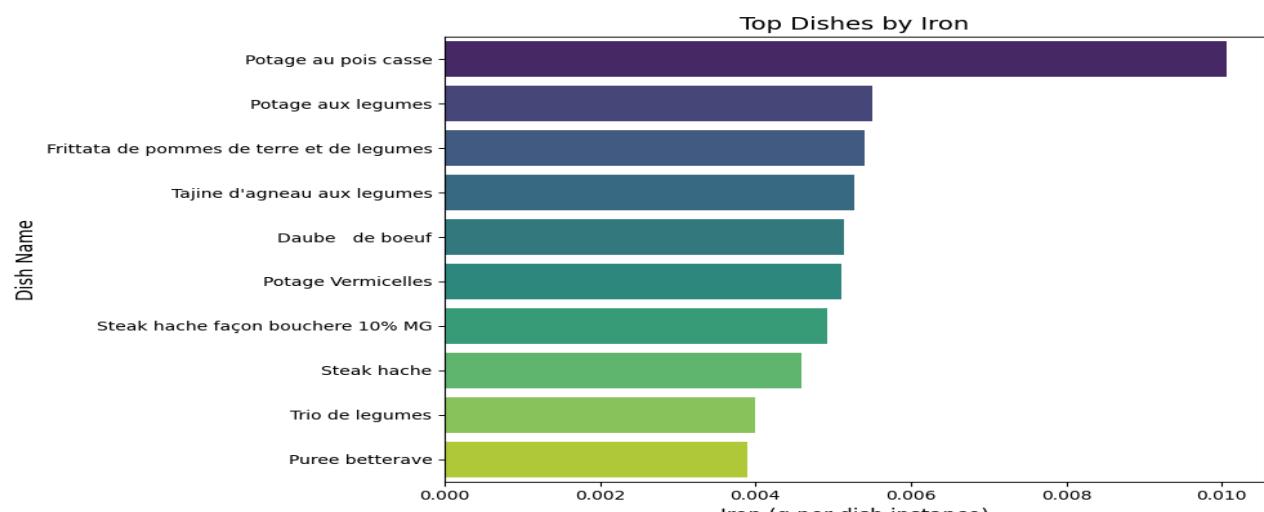
- **Graphiques et interprétations**

1. Teneur en calcium



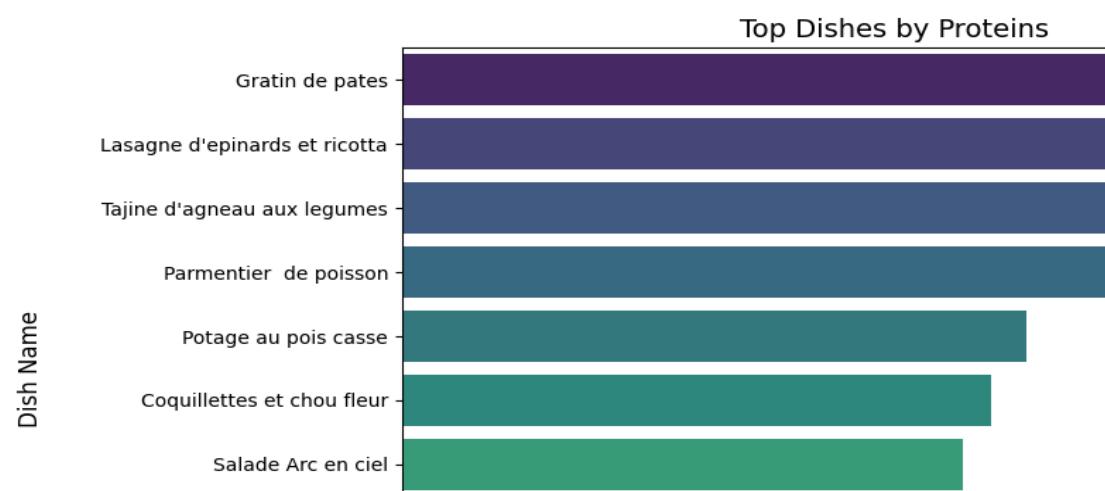
Le **Parmentier de poisson**, la **Salade Arc en ciel** et le **Gratin de pâtes** se classent en tête des plats les plus riches en calcium. La présence d'ingrédients laitiers comme l'emmental, la tomme ou la vache qui rit justifie ces apports. Le calcium, essentiel à la santé osseuse, est ici apporté par des sources alimentaires bien connues. Les valeurs semblent maintenant cohérentes avec les portions observables dans un plat standard. Ces plats peuvent ainsi contribuer de façon significative aux besoins quotidiens en calcium, notamment chez les enfants et les adolescents.

2. Teneur en fer



Les plats les plus riches en fer sont le **Potage au pois cassé**, le **Potage aux légumes** et la **Frittata de pommes de terre et de légumes**. Ces plats font appel à des légumineuses ou des légumes à feuilles, comme les pois cassés ou les épinards, connus pour leur teneur en fer. On observe aussi la présence de plats carnés (ex : steak haché), ce qui contribue à des apports en fer héminique, mieux absorbés par l'organisme. Ces résultats mettent en lumière l'importance de varier les sources alimentaires pour optimiser l'apport en fer dans les repas proposés.

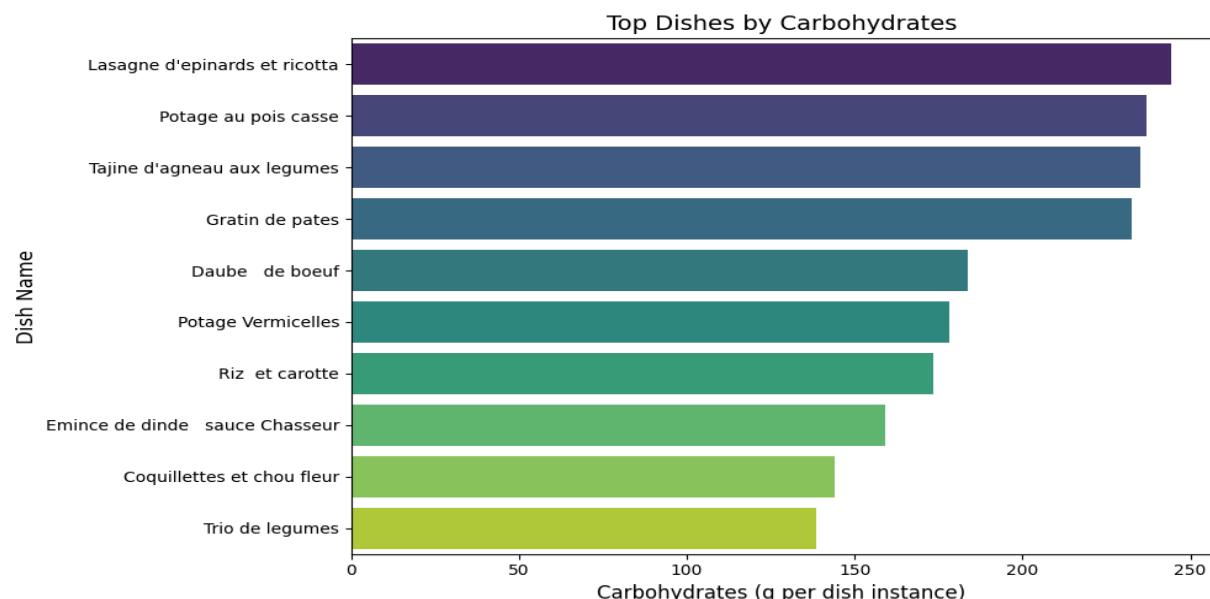
3. Teneur en protéines



Les plats les plus protéinés sont le **Gratin de pâtes**, les **lasagnes d'épinards et ricotta** et le **Tajine d'agneau aux légumes**. Ces recettes contiennent des ingrédients riches en protéines comme le fromage, la viande ou les œufs. Les niveaux observés sont élevés mais cohérents

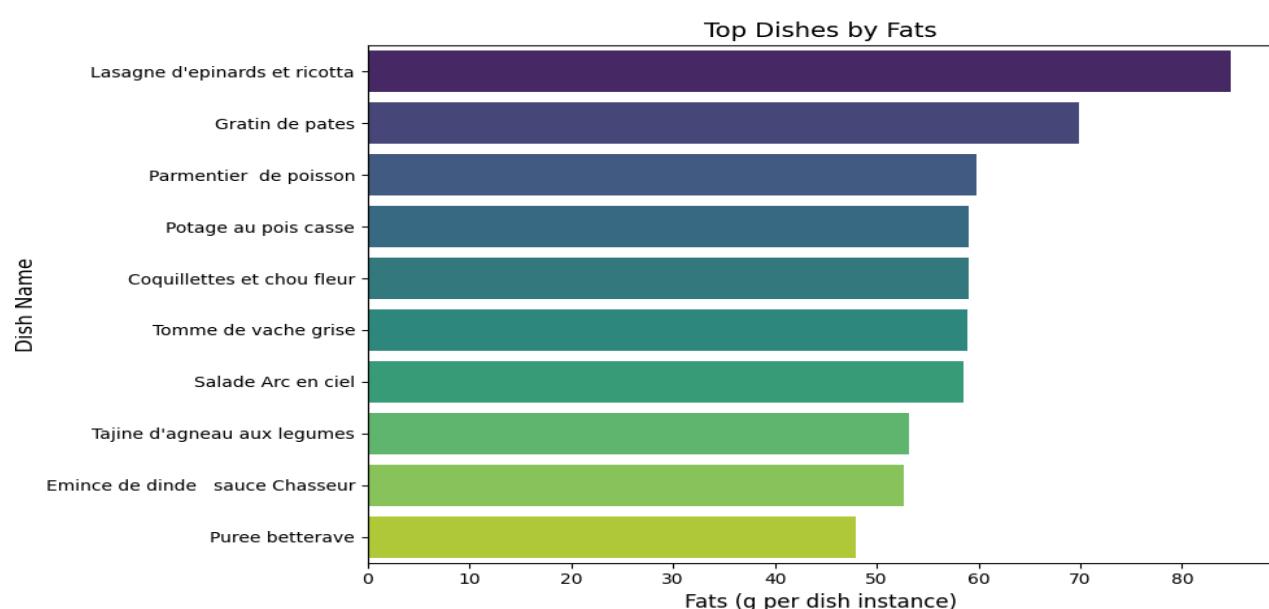
avec les quantités d'ingrédients principaux. Ces plats s'avèrent particulièrement adaptés aux besoins accrus en protéines, par exemple pour des adolescents ou des sportifs.

4. Teneur en glucides



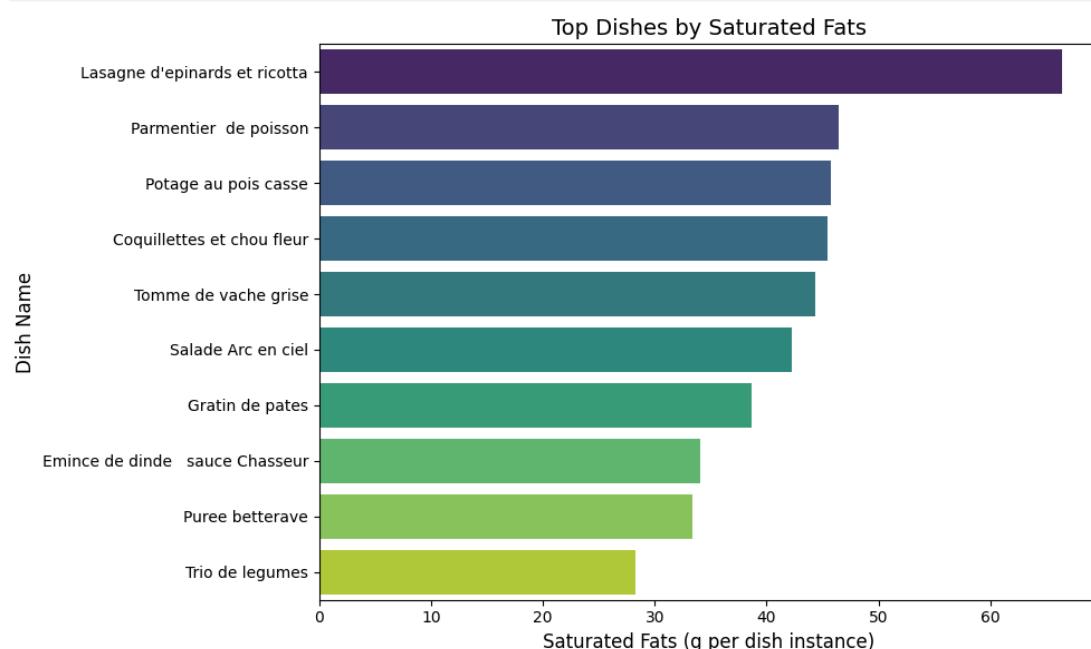
La **Lasagne d'épinards et ricotta**, le **Potage au pois cassé** et le **Tajine d'agneau aux légumes** arrivent en tête des plats les plus riches en glucides. Ces résultats s'expliquent par la présence abondante de féculents (pâtes, pommes de terre, pois, vermicelles...). Les apports restent logiques pour des plats complets comportant une base céréalière ou légumineuse. Ces plats constituent donc une bonne source d'énergie, adaptée à des repas principaux.

5. Teneur en lipides



Les plats les plus riches en lipides sont les **lasagnes d'épinards et ricotta**, le **Gratin de pâtes** et le **Parmentier de poisson**. Ces plats sont généralement préparés avec des sauces crémeuses, du fromage et parfois des matières grasses ajoutées, expliquent ces teneurs. Les quantités mesurées sont élevées mais restent crédibles après nettoyage des données aberrantes. Il convient toutefois de surveiller leur fréquence de consommation, notamment dans un cadre scolaire ou collectif.

6. Teneur en acides gras saturés



Les plats les plus concentrés en acides gras saturés sont les **lasagnes d'épinards et ricotta**, le **Parmentier de poisson** et le **Potage au pois cassé**. Cela reflète la présence d'ingrédients comme les produits laitiers ou certaines viandes. Les graisses saturées étant à consommer avec modération, ces plats doivent être équilibrés avec d'autres options plus légères. Une meilleure information nutritionnelle pourrait aider à orienter les choix vers des alternatives moins riches..

3.3.2. Conclusion de l'analyse des nutriments présents dans les plats

Cette analyse nous montre que chaque type de plat apporte une contribution nutritionnelle différente. Les plats à base de produits laitiers et légumineuses se révèlent particulièrement intéressants pour couvrir les apports en calcium et en fer. Les plats carnés et ceux à base de poisson apparaissent comme de bonnes sources de protéines et de lipides, souvent nécessaires pour l'équilibre énergétique global. Enfin, les plats riches en féculents ou légumes amylacés sont les plus représentés dans les apports en glucides, fournissant l'énergie indispensable pour une journée active.

Afin de concrétiser cette logique fonctionnelle, nous avons conçu une architecture technique modulaire et évolutive, décrite dans la section suivante.

4. Architecture technique

4.1. Scripts et fonctionnalités

1. Gestion des allergènes (allergies.py) :

- Demande à l'utilisateur s'il a des allergies via une interface interactive.
- Affiche la liste des allergènes disponibles dans les plats et permet une sélection multiple.

2. Filtrage des plats :

- Filtre les plats contenant les allergènes sélectionnés à l'aide de la fonction filter_dishes_by_allergens.

Utilité :

Personnalise les recommandations de plats en fonction des contraintes alimentaires de l'utilisateur.

Pour illustrer concrètement le fonctionnement de notre solution, nous décrivons ci-dessous le parcours type d'un utilisateur interagissant avec le système, selon trois scénarios distincts : snack, repas complet ou consultation de statistiques.

- Gestion du menu et des choix (menu.py)

Fonctionnalités :

1. Interface utilisateur :

- Propose un menu interactif avec 6 catégories : entrée, plat principal, garniture, dessert, pain, supplément.
- Permet de choisir des plats, voir leurs ingrédients, et d'en retirer certains si besoin.

2. Gestion des commandes :

- Stocke les choix de l'utilisateur dans une liste globale chosen_products.
- Affiche un récapitulatif avec des icônes pour chaque catégorie (ex: 🍔 pour le plat principal).

3. Options supplémentaires :

- Accès aux statistiques nutritionnelles (stats) ou sortie du programme (stop).

Utilité :

Sert d'interface principale pour la sélection des plats et la personnalisation des commandes.

- Nutriments (nutrients.py)

Fonctionnalités :

1. Analyse nutritionnelle :

- Calcule les apports moyens en nutriments (protéines, lipides, etc.) par plat.

- Filtre les valeurs aberrantes (ex : Glucides > 80g considéré comme invalide).

2. Visualisation :

- Génère des graphiques (barplots) des plats les plus riches pour un nutriment donné via matplotlib/seaborn.

3. Statistiques personnalisées :

- Affiche les nutriments des plats choisis par l'utilisateur (display_chosen_products_nutrients).

Utilité :

Fournit des insights nutritionnels et aide à faire des choix alimentaires éclairés

• Recapitulatif (recap.py)

Fonctionnalités :

1. Récapitulatif :

- Affiche un résumé des plats commandés avec des icônes associées à chaque catégorie.

2. Interaction :

- Propose de continuer ou de quitter le programme.

3. Messages utilisateur :

- Affiche un message de bienvenue et un message de sortie personnalisé.

Utilité :

Améliorer l'expérience utilisateur en synthétisant les choix et en permettant une navigation claire.

• Script shared_data.py

Fonctionnalités :

• Variable globale :

Définit une liste chosen_products pour stocker les plats sélectionnés par l'utilisateur.

Utilité : Permet le partage de données entre les scripts (ex: menu.py, validation.py).

• Script validation.py

Fonctionnalités :

1. Validation interactive :

- validate_snack() et validate_product() :

- Enregistrent les choix de l'utilisateur dans chosen_products.

- Proposent d'ajouter un autre produit ou d'afficher le récapitulatif (print_recap).
Utilité : garantit que les sélections sont correctement enregistrées avant confirmation.

- **Script visualization.py**

Fonctionnalités :

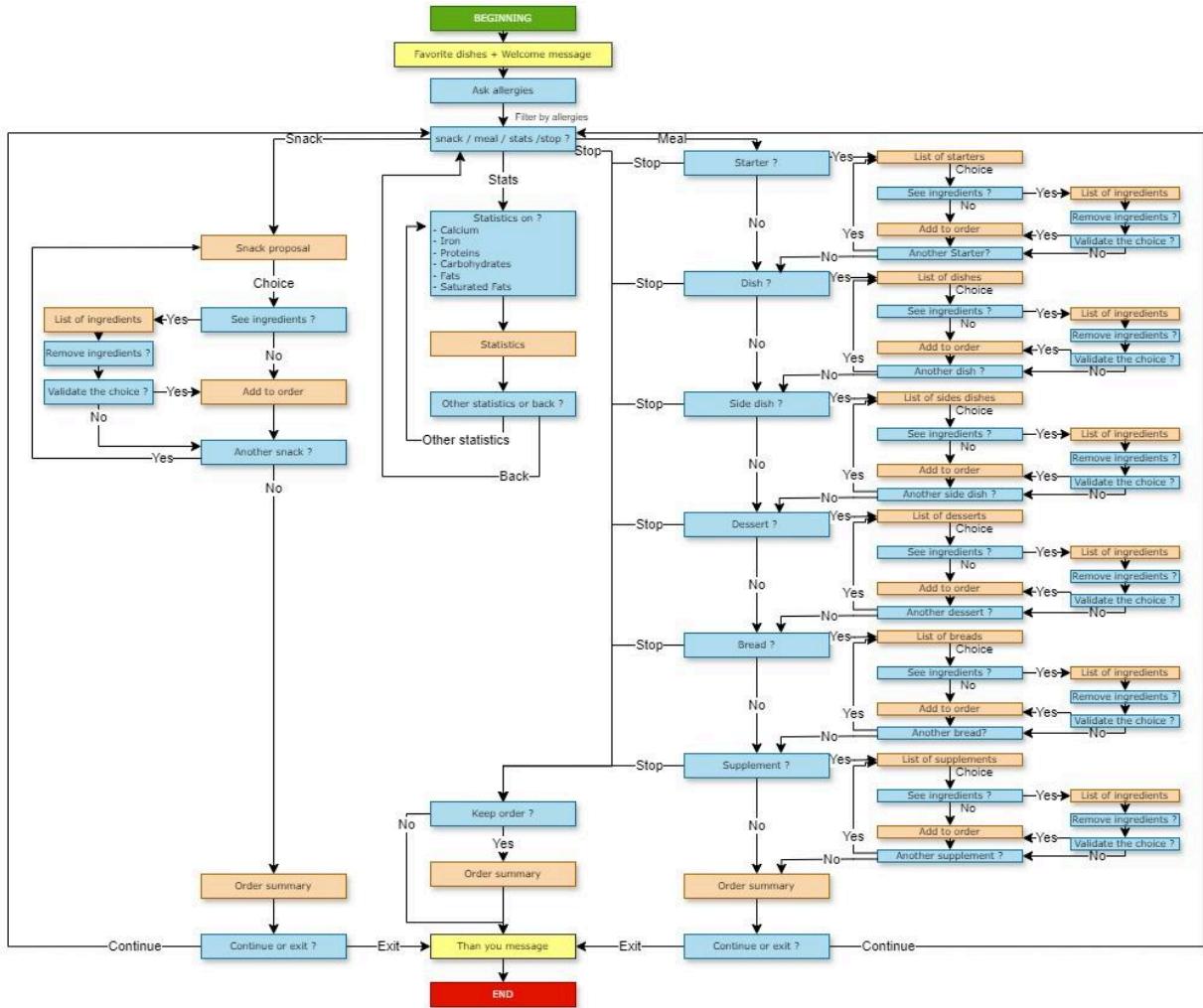
- Affichage d'images :
display_graph_image() charge et affiche une image (ex: graphique des plats populaires) depuis un chemin relatif.
Utilité : Visualisation des résultats d'analyse (ex: statistiques de commandes).

- **Script principal (main.py)**

Il orchestre le programme en exécutant d'abord le module de gestion des allergies, puis en lançant la sélection des repas en fonction des préférences de l'utilisateur.

Cette architecture permet d'assurer un traitement efficace des données, une expérience utilisateur optimisée, une validation rigoureuse des choix et une vérification fiable du système via des tests unitaires. L'ensemble garantit une solution robuste, adaptée aux préférences et aux contraintes alimentaires des utilisateurs.

4.2. Structure logique



Voici le logigramme représentant l'architecture du code Python de notre outil. Lors du lancement du programme, l'utilisateur accède dans un premier temps à une liste des plats les plus commandés, ce qui met en évidence les plats populaires auprès des clients.

Un message de bienvenue s'affiche ensuite, suivi d'une première question : avez-vous des allergies ? La réponse à cette question permettra de filtrer les plats proposés en fonction des restrictions de l'utilisateur.

Puis, l'application pose une nouvelle question : souhaitez-vous consulter des statistiques sur certains plats si on a des besoins spécifiques cela peut-être intéressant, ou préférez-vous simplement choisir un snack ou un menu complet ?

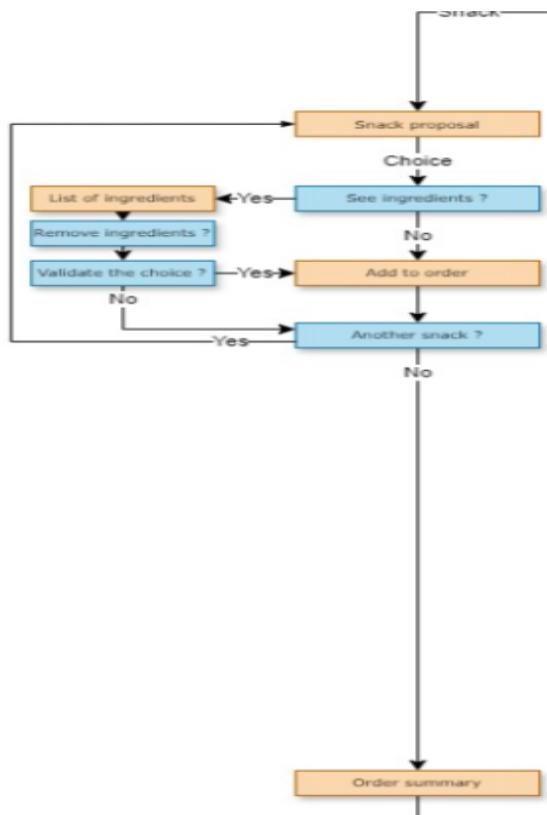
À partir de cette étape, le logigramme se divise en trois branches distinctes, correspondant aux différents types de parcours utilisateur.

4.2.1. Parcours Snack et Repas Complet

L'utilisateur est accueilli et invité à indiquer s'il a des allergies.

Les snacks contenant des allergènes signalés sont automatiquement exclus de la liste des propositions.

1. Sélection d'un snack



Une **proposition de snack** est faite.

L'utilisateur peut consulter la **liste des ingrédients** avant de prendre sa décision.

Une fois son choix fait, il peut **valider** et l'ajouter à la commande.

2. Ajout d'autres snacks (optionnel)

Après validation, le système demande si l'utilisateur veut **ajouter un autre snack**.

S'il répond **oui**, il revient à l'étape précédente et peut sélectionner un autre snack.

S'il répond **non**, le processus passe au **récapitulatif de commande**.

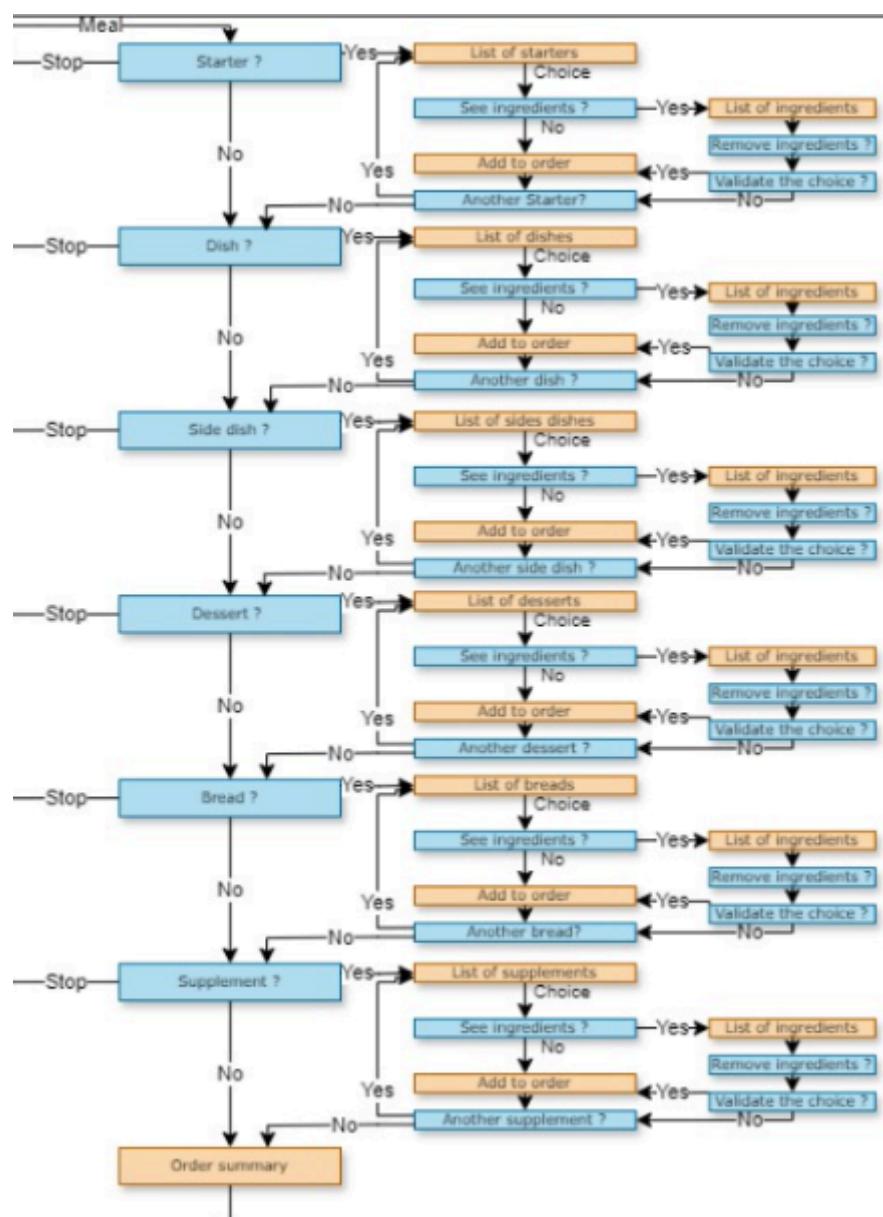
3. Finalisation ou continuation

Une fois la sélection terminée, un **récapitulatif de la commande** s'affiche.

L'utilisateur peut choisir de **finaliser sa commande** (ce qui déclenche un message de remerciement) ou **continuer** pour ajouter d'autres éléments.

Si l'utilisateur choisit d'arrêter, le programme met fin à l'exécution immédiatement.

4. Parcours "Repas Complet" (Meal)



Choix des éléments du repas

L'utilisateur construit son repas en plusieurs étapes successives :

- Entrée (Starter) 

L'utilisateur peut choisir une entrée parmi une liste filtrée.

Il peut voir les ingrédients avant de sélectionner son choix et retirer certains ingrédients si nécessaire.

Une fois validé, il peut ajouter une autre entrée ou passer à l'étape suivante.

- Plat principal (Dish) 

Une liste de plats principaux est proposée. L'utilisateur peut consulter les ingrédients et supprimer ceux qu'il ne souhaite pas avant de valider son choix. Il peut ajouter un autre plat principal ou passer à l'étape suivante.

- Accompagnements (Side Dish) 

L'utilisateur peut choisir un ou plusieurs accompagnements. Il peut voir les ingrédients, en retirer si nécessaire, puis ajouter les plats validés à sa commande. S'il le souhaite, il peut ajouter un autre accompagnement ou passer à l'étape suivante.

- Dessert (Dessert) 

L'utilisateur peut sélectionner un ou plusieurs desserts.

La consultation des ingrédients est possible avant validation, et il peut retirer certains ingrédients si besoin.

Une fois son choix fait, il peut ajouter un autre dessert ou passer à l'étape suivante.

- Pain (Bread) 

Une sélection de différents types de pain est proposée. L'utilisateur peut voir la liste des ingrédients, supprimer ceux qu'il ne souhaite pas, avant validation.

Il peut ajouter un autre pain ou passer à l'étape suivante.

- Suppléments (Supplément) 

Des suppléments (comme des sauces, soupes ou condiments) sont disponibles.

L'utilisateur peut consulter les ingrédients et enlever ceux qu'il ne veut pas avant de les ajouter. Il peut ajouter plusieurs suppléments ou passer à l'étape finale.

7. Conclusion

Ce projet est une application interactive de recommandation de plats, conçue pour améliorer l'expérience client dans un restaurant en tenant compte des allergies, des préférences alimentaires et des apports nutritionnels.

7.1. Points Clés :

➤ **Nettoyage et préparation des données :**

Standardisation des fichiers Excel, gestion des valeurs manquantes et conversion des nutriments en données exploitables.

➤ **Interface intuitive :**

- Sélection de plats par catégorie (entrée, plat principal, dessert, etc.), avec possibilité de retirer des ingrédients.
- Filtrage automatique des allergènes pour une sécurité alimentaire optimale.

➤ **Analyse nutritionnelle :**

- Visualisation des nutriments (protéines, glucides, etc.) pour aider les clients à faire des choix éclairés.
- Statistiques descriptives sur les plats les plus populaires et les allergènes les plus courants.

➤ **Modularité et robustesse :**

- Architecture modulaire permettant des améliorations futures (ex : intégration d'un système de feedback).
- Gestion des erreurs (fichiers manquants, saisies invalides) pour une expérience fluide.

7.2. Limites

Ce projet est une application interactive de recommandation de plats, conçue pour améliorer l'expérience client dans un restaurant en tenant compte des allergies, des préférences alimentaires et des apports nutritionnels. Cependant, il présente des limites.

7.2.1. Problèmes rencontrés :

Les données nutritionnelles présentées dans la base ne sont pas toujours correctement renseignées, et certaines valeurs semblent incohérentes ou manifestement erronées. Cela peut provenir d'une mauvaise saisie ou d'une approximation lors de l'élaboration des fiches ingrédients. Ainsi, une validation ou une correction manuelle de la base serait nécessaire pour garantir une évaluation encore plus précise et fiable des apports nutritionnels des plats proposés.

- ***Manque de données exhaustives pour les labels***

Initialement, nous souhaitions attribuer des labels aux plats choisis par les clients afin de mieux catégoriser leurs préférences et de leur offrir des suggestions plus personnalisées. Cependant, nous avons abandonné cette fonctionnalité en raison du manque de données exhaustives. Certaines entrées étaient incomplètes ou incohérentes, ce qui rendait difficile l'attribution fiable des labels. Une amélioration future pourrait consister à structurer davantage les données et à intégrer une base de labels plus complète et fiable.

- ***Fiabilité des informations nutritionnelles***

Un autre problème rencontré concerne les valeurs nutritionnelles des plats. Certaines fiches indiquent un nombre de nutriments anormalement élevé, ce qui remet en question la fiabilité des données. Cela peut fausser les recommandations faites aux clients, en particulier pour ceux ayant des exigences nutritionnelles spécifiques. Une amélioration possible serait de normaliser les informations nutritionnelles en effectuant un contrôle qualité sur les données sources ou en s'appuyant sur des bases de données nutritionnelles validées.

- ***Intégration d'un modèle d'apprentissage automatisé***

Au niveau du code, il serait intéressant d'utiliser des modèles d'apprentissage qui permettent de suggérer des plats aux clients sur la base de leurs choix passés.

7.3. Perspectives d'Amélioration

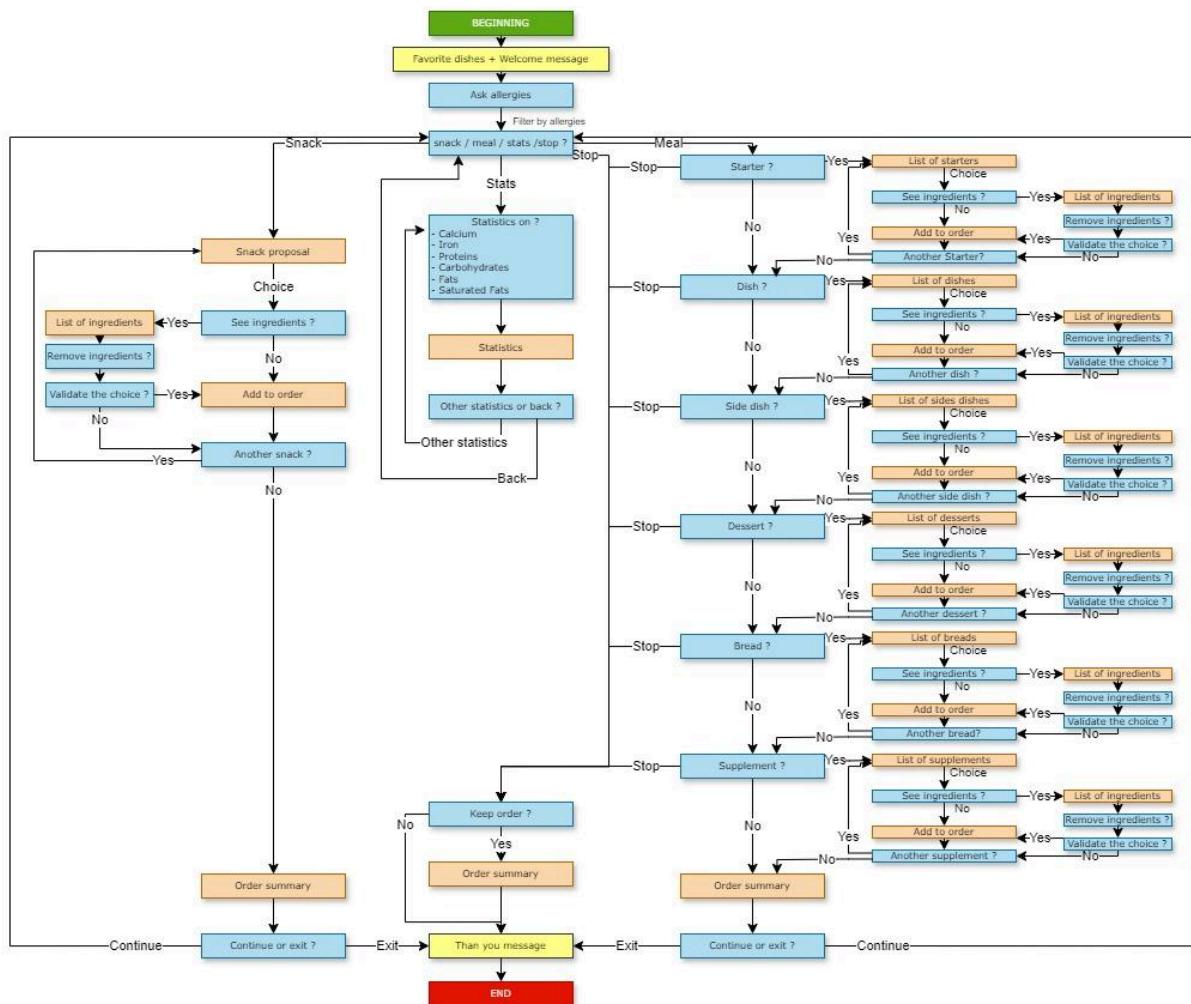
- ◆ **Base de données** : remplacer les fichiers Excel par une base SQL/NoSQL pour gérer un plus grand volume de données.
- ◆ **Recommandations personnalisées** : Utiliser l'historique des commandes pour suggérer des plats adaptés.
- ◆ **Version web/mobile** : Développer une interface accessible via navigateur ou application.

7.4. Bilan

Ce programme allie traitement des données, interactivité et analyse nutritionnelle pour offrir une solution complète et adaptable. Il peut être déployé en l'état dans un restaurant ou étendu pour des usages plus avancés (restauration collective, applications santé).

8. Annexes

- Logigramme détaillé



• Roadmap du projet

