

Projet Innovation

Etat de l'art

Equipe-Q : Eatrack

Soufiane Aourinmouche, Hugo Croenne, Alix Humbert, Mohamed Younes

Critères de comparaison :

Un tableau comparatif pour les fonctionnalités :

<https://docs.google.com/document/d/1ijGnPt6zbqTBbeiQHS0N5cXkeE8zZ4MleYjidfr9QZ8/edit?usp=sharing>

Toutes les applications concurrentes mentionnées dans le tableau utilisent un filtrage de recettes (en fonction des aliments du frigo, préférences et exigences). **Eatrack** s'appuie sur un vrai système de recommandation utilisant des méthodes scientifiques (*voir slide 2*) pour analyser le profil de l'utilisateur et son historique et celui des personnes avec un profil qui lui ressemble.

Eatrack propose également de configurer son système de recommandation, elle traite alors plusieurs contextes possibles : Dans le contexte d'une famille, il maximise l'utilisation des aliments du frigo pour éviter le gaspillage. Dans le contexte d'un groupe d'amis, il maximise la prise en compte des préférences du groupe. Dans le contexte d'un picnic, il ne recommande que des recettes adaptés à ce contexte. Plusieurs contextes peuvent être ajoutés

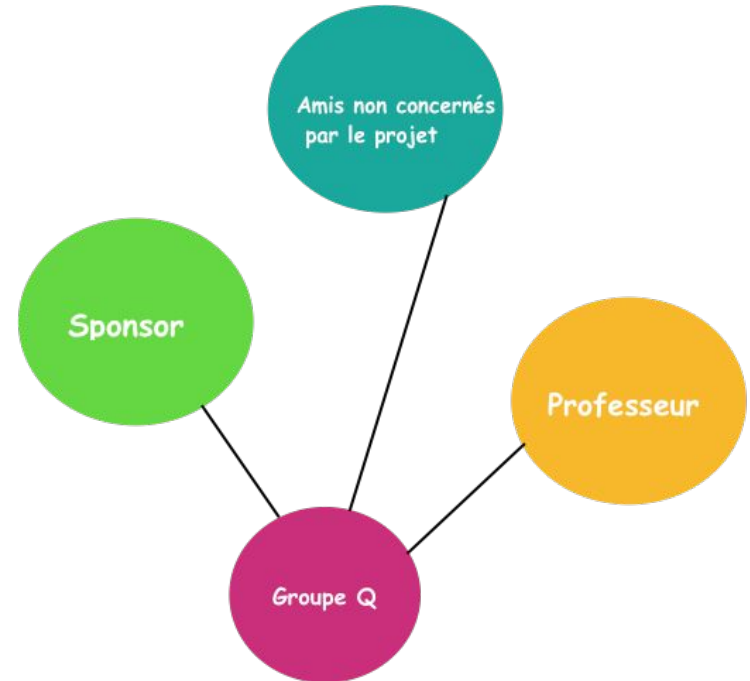
Pour la partie monétisation, ces applications utilisent souvent des publicités qui peuvent être envahissantes et nuisent à l'expérience utilisateur. Nous pensons monétiser notre projet de manière invisible, en cherchant des partenariats avec des marques (comme Charal, Panzani, Uncle Ben's ...), ainsi dans notre système de recommandations, nous mettons en priorité les recettes contenant des produits que proposent nos marques partenaires (en cohésion avec les préférences et le profile de l'utilisateur bien sûr).

Mots clés pour la recherche :



Avis d'experts :

Ce genre d'application est utilisé au quotidien par tout le monde. Les membres du groupe-Q, les professeurs, le sponsor et nos connaissances extérieures ont donné leurs avis pour nous aiguiller.



Technologie nécessaires

Pour que notre projet soit extensible, nous avons besoin d'une architecture logicielle qui permet la modification/intégration de modules sans impacter le reste.

Pour pouvoir proposer des recommandations pertinentes en fonction de l'utilisateur, ses préférences, allergies, recettes testées auparavant et celles testées par les utilisateurs ayant un profil similaire, nous avons besoin d'un système de recommandation basé sur des modèles mathématiques.

Afin de renseigner à l'application les aliments du frigo, nous utilisons l'appareil photo du téléphone, nous avons donc recours au Machine Learning pour la reconnaissance d'images.



Critères de comparaison de ces technologies

Architecture modulaire : Nous utilisons une architecture micro-services, qui nous permet d'intégrer/modifier des services sans impacter le reste. Elle nous permet également d'utiliser des services implémentés avec de différents langages (service en Python pour le système de recommandation, .Net pour les web-services, **Java/Android** pour le client).

Système de recommandation et Valorisation de données : Il s'agit du point le plus important dans notre projet. En effet, les concurrents utilisent un filtrage des recettes en fonctions des ingrédients du frigo, les préférences et les exigences. **Eatrack** s'appuie sur un vrai système de recommandation basé sur des modèles mathématiques implémentés en Python dans un service externe. Nous avons utilisé une librairie "**Surprise**" parce qu'elle propose toutes les fonctionnalités nécessaires au système de recommandation dont notre projet a besoin. Cette librairie est facilement configurable, ce qui permet de différencier les différents contextes d'utilisations de l'application. De plus celle-ci est implémentée en Python, langage étudié dans notre parcours scolaire, open-source et propose des tutoriels pour la prendre en main.

Reconnaissance des aliments : Nous comptons utiliser Keras plutôt que **Tensorflow**. En effet, pour construire un projet "from-scratch", **Keras** est recommandé puisqu'il est facile à prendre en main (*pour un projet de 3 semaines*). Cependant **Tensorflow** étant plutôt orienté Deep Learning, il ne répond pas précisément à notre besoin.

Critères d'évaluation :

Satisfaction de l'utilisateur (validation de la recette recommandée) pour le système de recommandation.
Ajout d'un nouveau service sans impacter l'ensemble de composants pour l'architecture logicielle.
Validation/Invalidation de l'aliment reconnu par l'application pour la reconnaissance des aliments.

Système de recommandation : A chaque fois que nous recommandons une recette à un utilisateur, il pourra attribuer une note à cette recommandation ce qui nous permettra d'évaluer notre système de recommandation. Une analyse de ces notes reçues pourra nous aider à trouver les anomalies du système : les cas extrêmes ou rares.

Dans un premier temps, des tests fonctionnels et unitaires seront mis en places afin de valider notre système.

Architecture : L'ajout d'un service de localisation pour recommander des aliments proposés par les supermarchés à proximité ne nécessiterait pas la modification des autres services et composants déjà existants.

Reconnaissance : Lorsque l'utilisateur prend en photo un aliment du frigo, nous proposons en retour 3 aliments possibles (3 output de notre système de reconnaissance des aliments). Il pourra sélectionner celui qui correspond à son aliment, ou alors sélectionner "Aucun". Cela nous permettra d'évaluer notre système de reconnaissance des aliments.