

# Cahier des charges - Cookbook

## Table des matières:

1. Description du projet
2. Fonctionnalités
3. Spécifications techniques
4. Wireframes de base

## Description du projet

---

Une application web de repertoire de recettes de cuisine permettant à l'utilisateur de stocker ses recettes de cuisine, les organiser, lancer une recherche et les consulter.

Additionnellement, l'application enregistrera les ingrédients automatiquement à une liste *Placard* qui reprendra tous les aliments qui seraient nécessaires à la réalisation d'une/des recettes. L'utilisateur aura le choix d'indiquer ce qu'il possède, ou pas.

L'aliment sera ajouté ensuite à une liste *Course* pour que l'utilisateur puisse mieux visualiser ses achats potentiels. Cette liste de course pourra être réorganisée en plusieurs (sous) listes en fonction des besoins.

## Fonctionnalités

---

### Sélection et visualisation d'une recette

L'utilisateur accède à une recette depuis la page d'accueil ou la liste de dossiers. Après avoir lancé une recherche ou pas (voir fonctionnalité plus bas).

La page d'une recette affichera les informations suivantes: le nom de la recette, une description de la recette, la liste des ingrédients et leur quantité, les étapes de la préparation du plat, le temps de préparation, le nombre de personnes, une cote personnelle et la/les photo(s) du plat.

## Modifier/Supprimer une recette

Ces actions sont accessibles depuis la page de la recette ou la liste des dossiers. Une redirection sera faite vers un formulaire de modification ou une fenêtre modale demandera la confirmation de la suppression.

## Pin/Unpin une recette

L'utilisateur pourra depuis la page de la recette ou la liste de dossiers mettre en évidence une recette (pin) ou l'inverse (unpin). L'utilisateur pourra aussi choisir la taille du pin depuis la page d'accueil en affichant les options de la recette en question.

## Création d'un dossier

L'utilisateur depuis la page de dossiers peut créer un nouveau dossier en sélectionnant l'icone dans la bar latérale.

Optionnellement un dossier peut être créer lors de la création d'une recette pour la stocker directement dans le dit dossier. La réorganisation des recettes se fait en cochant une recette et en choisissant le dossier dans lequel on veut déplacer la recette. Plusieurs recettes peuvent être déplacées en une seule fois.

## Création d'une recette

L'utilisateur accède au formulaire de création depuis la bar latérale sur n'importe quelle page en cliquant sur l'icone correspondante.

L'utilisateur devra fournir les informations suivantes: le nom de la recette, une description, les ingrédients nécessaires et leur quantité, le nombre de personnes, une cote personnelle (max. 5), le temps de préparation et les étapes de préparation. L'image est optionnelle et pourra être rajoutée à travers le formulaire de modification par après. Les nouveaux ingrédients pas encore connus de l'application seront insérés à la liste *Placard* (voir fonctionnalité qui suit).

## Ajout automatique des ingrédients (produits)

L'application propose lors de la création d'une recette (voir fonctionnalité plus haut) les produits déjà répertoriés dans la liste *Placard* et ajoutera ensuite automatiquement les nouveaux ingrédients en tant que produits à l'insertion d'une nouvelle recette.

## Ajout des ingrédients à la liste de course

L'utilisateur coche dans la liste *Placard* les produits qui ne sont plus disponibles dans sa cuisine et ceux-ci sont ajoutés à une liste générée automatiquement qui comprend tous les produits sans distinctions. L'utilisateur pourra ensuite les trier en créant ses propres listes de course (voir fonctionnalité qui suit). Un produit peut se retrouver dans plusieurs listes différentes.

## Création d'une liste de course (et organisation des produits)

L'utilisateur peut créer une nouvelle liste en cliquant sur l'icone depuis la bar secondaire seulement accessible sur la page *Placard*. L'utilisateur ajoute des produits (voir fonctionnalité plus haut) dans une liste au choix depuis les options du produit. Un produit peut se retrouver dans plusieurs listes mais il sera unique dans une liste donnée.

L'utilisateur peut également joindre une note au produit qui est unique à ce produit dans une liste spécifique. Le même produit pourra avoir une note différente dans une autre liste dans laquelle il se trouverait également.

## Recherche dans l'application

Par nom de recettes, nom de dossier, cotation (passer un nombre sans unité horaire, ex : 3 sachant que le maximum est 5), ingrédients, temps de préparation (ex : 30 min) et nom de liste de course. Des mots-clé sont joints à la recherche pour mieux filtrer les produits dans la bar de recherche même.

## Note sur la page d'accueil

La page d'accueil affichera les recettes mises en évidence sous formes de cartes de tailles différentes choisies par l'utilisateur. Les informations affichées se limiteront à l'image, le temps de préparation, la cote personnelle et le nombre de personnes.

## Spécifications techniques

---

L'API utilisé dans le backend sera faites en PHP en utilisant le framework Symfony. Celui-ci servira d'intermédiaire entre la base de données et l'interface utilisateur qui elle sera programmée avec React et Tailwind.

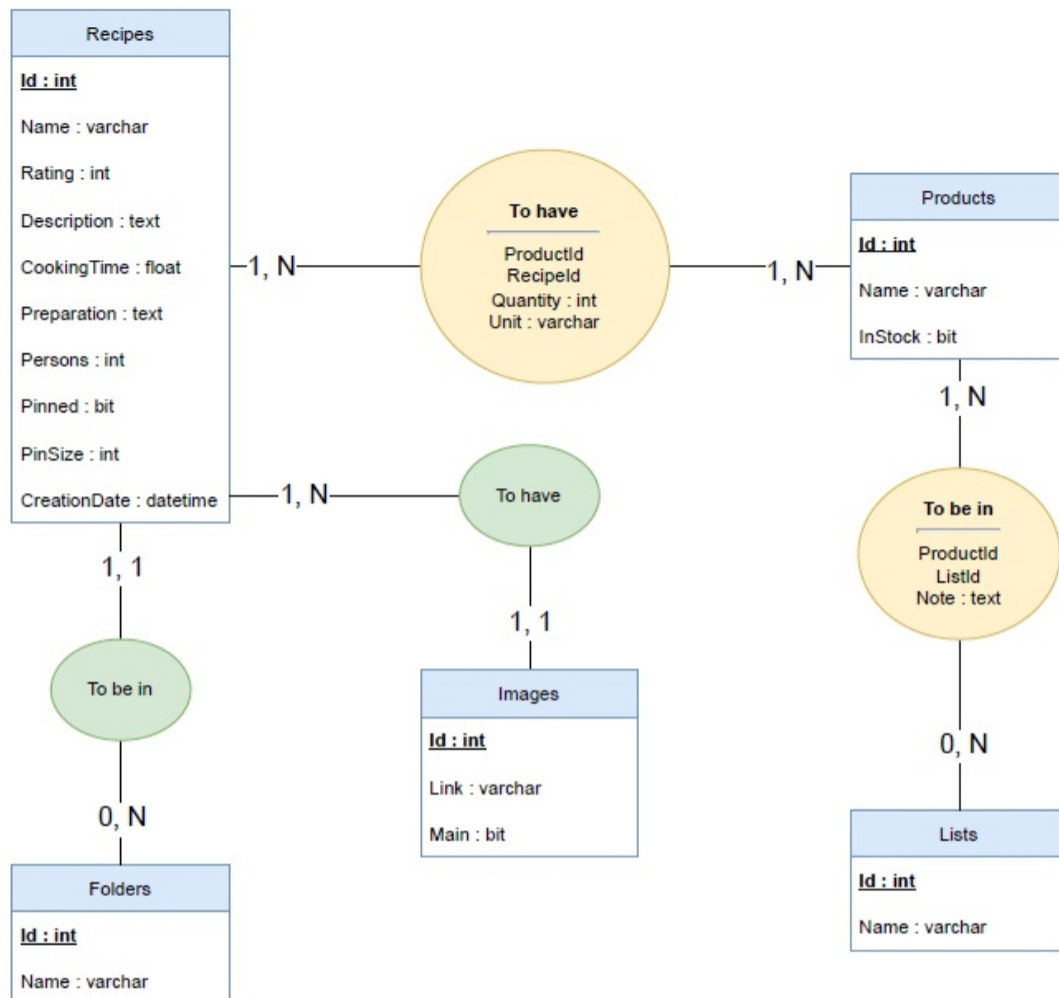
Le choix de cette combinaison vient du fait que l'intégration d'ElasticSearch à mon project se fait plus aisément par des modules React spécialement dédiés.

ElasticSearch étant en lui-même une API qui importerait et indexerait les éléments choisis dans la base de données pour offrir une expérience de recherche dans l'application plus performante et efficace. La fonctionnalité de recherche se baserait alors sur Elastic App Search et son module Search UI à usage dans React.

Les images seront renommées et stockées dans un dossier *Images*. Seulement le lien relatif sera repertorié dans la base de données.

## Base de données

DBMS: MySQL. Comprendra les tables suivantes (en anglais):



L'application (utilisateur) accède à la base de données seulement pour consulter et ajouter des éléments aux tables.

Au niveau du backend en Symfony, l'ORM Doctrine sera utilisé et le schéma ci-dessus est donc à titre de référence pour toujours bien abstraire la base de données des différentes couches au niveau applicatif. Les tables auront donc leur équivalence en modèles Doctrine.