Cahier des Charges – Fridge Manager

À destination de l’équipe de développement

## 🔧 Aperçu général

L'application Fridge Manager est une solution web complète visant à aider les utilisateurs à :

- Gérer les produits alimentaires d’un frigo

- Suivre les dates de péremption

- Générer automatiquement des listes de courses

- Être notifiés en cas de produits bientôt périmés

Le back-end est développé en FastAPI, et le front-end est prévu avec React, Vue, Angular ou Flutter.

## 🔐 1. Authentification & Sécurité (Back-end)

- Connexion via email/mot de passe avec token JWT

- Protection des routes via Authorization: Bearer <token>

- Accès au profil utilisateur avec /me

- Mise à jour du profil via /users/me

- Hashage des mots de passe avec passlib + bcrypt

## 📦 2. Gestion de l’inventaire (Produits)

\*\*Front-End\*\*

- Liste des produits affichée dans un tableau de bord

- Filtres : catégorie, emplacement, date de péremption

- Indicateur rouge pour les produits bientôt périmés

- Bouton d’ajout de produit

\*\*Back-End\*\*

- CRUD complet avec /items/

- Champs : nom, quantité, unité, catégorie, emplacement, date de péremption

- Recherche des produits expirant bientôt via /items/soon-expired

## 🛍️ 3. Listes de Courses Intelligentes

\*\*Front-End\*\*

- Création de listes

- Ajout / suppression d’articles

- Cocher un article (✓ acheté)

- Génération automatique depuis les produits manquants ou bientôt périmés

- Renommage / suppression d'une liste

\*\*Back-End\*\*

- Endpoints : /lists/, /lists/{list\_id}/items/batch, /lists/items/{item\_id}

- Prise en charge de plusieurs listes par utilisateur

## 📣 4. Notifications

\*\*Front-End\*\*

- Affichage des produits bientôt périmés

- Historique des notifications

\*\*Back-End\*\*

- Notifications automatiques par email (produits expirant sous 48h)

- Planification via apscheduler

- Historique via /notifications/sent

## 👤 5. Profil Utilisateur

\*\*Front-End\*\*

- Visualisation & modification du profil

- Changement de mot de passe

- Déconnexion (suppression du token)

\*\*Back-End\*\*

- /me → Récupérer l’utilisateur connecté

- /users/me → Modifier ses informations

## 🎨 6. Design et Interface Utilisateur

- Interface claire et minimaliste

- Statuts colorés : ✅ Vert (OK), ⚠️ Jaune (bientôt périmé), ❌ Rouge (périmé)

- Icônes pour : frigo/congélateur, articles achetés/non achetés

- Responsive design : mobile-first

## ⚙️ 7. Stack Technologique

- Back-End : FastAPI, SQLAlchemy, Apscheduler, JWT

- Front-End : React / Vue / Angular / Flutter

- UI : Material UI / Tailwind / Bootstrap

- API : Axios / Fetch

- Stockage local : localStorage ou sessionStorage

## 🧪 8. Testabilité

- Compatible navigateurs : Chrome, Firefox, Safari

- Interface testable en mobile et desktop

- Tests unitaires recommandés

## 🔄 9. Points techniques résolus

- Problème SMTP → résolu avec App Password Gmail

- Erreurs 422/500 → fixées avec Query() et joinedload()

- Relations entre listes et items → chargées correctement

- Support multi-utilisateur intégré

## 📤 10. Intégration & Déploiement

- API Base URL : https://api.fridge-manager.com/v1

- Gestion des erreurs 401 → redirection vers /login

- Swagger disponible pour documentation

## 🚧 11. Prochaines étapes

- Intégration complète front-back

- CI/CD (GitHub Actions, etc.)

- Génération OpenAPI complète

- Export PDF/Word pour présentation