

# **Cahier des Charges - Projet Informatique Agile**

## **SmartPantry – Gestion intelligente des produits alimentaires**

Version : 1.0

**Date :** 24 October 2025

**Équipe :**

- Marwane Abouhamou
  - Youssef Najimi
  - Abdellah Belgnaou

**Client/Sponsor :** Professeur Ait El Cadi

## **1. Description Générale du Projet**

### **Objectif :**

SmartPantry est une application mobile permettant aux utilisateurs de gérer les produits qu'ils achètent en scannant le code-barres ou le QR code des articles. L'application enregistre la date d'expiration et envoie des notifications avant que les produits n'expirent. Elle propose également des recettes personnalisées pour utiliser les ingrédients proches de la date limite, afin de réduire le gaspillage alimentaire.

## **2. Objectifs du Projet**

### **2.1 Objectifs Principaux**

- **Développer un site web et une application mobile fonctionnelle** pour Android et iOS d'ici 7 mois.
- **Permettre le scan des produits via code-barres/QR code** pour un enregistrement rapide et fiable des informations.
- **Gérer automatiquement les dates d'expiration** et envoyer des rappels personnalisés avant la péremption.
- **Intégrer un moteur de recherche de recettes** basées sur les produits à consommer.
- **Réduire le gaspillage alimentaire** en incitant les utilisateurs à consommer les produits avant expiration.
- **Proposer une interface intuitive et ergonomique**, simple d'utilisation même pour les non-technophiles.
- **Assurer la synchronisation des données** entre plusieurs appareils via le cloud (Firebase).
- **Garantir la sécurité et la confidentialité** des données utilisateur (authentification sécurisée).
- **Permettre la consultation de statistiques** (ex. : produits les plus gaspillés, nombre de produits consommés à temps).
- **Offrir la possibilité d'ajouter manuellement un produit** non reconnu par le code-barres.
- **Inclure un système de filtrage et de tri intelligent** (par catégorie, date d'expiration, fréquence d'utilisation).
- **Optimiser les performances** pour un temps de réponse inférieur à 2 secondes sur tous les appareils.
- Permettre la création d'un **compte familial**, permettant à plusieurs membres de partager et gérer le même inventaire.
- Permettre la gestion des rôles au sein d'un compte familial (propriétaire, éditeur, lecteur).
- Offrir la possibilité d'inviter d'autres utilisateurs via une notification.
- **Permettre l'utilisation hors ligne** (accès aux produits enregistrés même sans connexion Internet).
- **Inclure un mode "suggestion intelligente"** qui propose des recettes à partir des ingrédients disponibles dans le garde-manger.
- **Faciliter les mises à jour et la maintenance** du code via une architecture modulaire.

### **2.2 Objectifs Secondaires**

- **Former l'équipe à la méthodologie Agile (Scrum)** pour assurer une meilleure organisation du travail en sprints.
- **Intégrer des tests unitaires et d'interface** pour garantir la stabilité et la qualité du code.
- **Améliorer la collaboration entre les membres de l'équipe** à l'aide d'outils de gestion de projet (ex. Trello, Jira, Dicord).

- Mettre en place un système de gestion de version efficace (Git/GitHub) pour un suivi clair des développements.
- Documenter toutes les étapes du projet (techniques et fonctionnelles) pour faciliter la maintenance et l'évolution future.
- Assurer une expérience utilisateur fluide grâce à des tests d'ergonomie et de navigation.
- Développer une interface multilingue (français/anglais) pour une plus grande accessibilité.
- Optimiser la consommation de ressources (batterie, mémoire) afin de garantir la performance sur les appareils mobiles.
- Prévoir la possibilité d'intégrer des fonctionnalités futures, comme la reconnaissance automatique de texte (OCR) pour lire les dates d'expiration.
- Respecter les principes d'éco-conception logicielle, en réduisant le gaspillage de ressources numériques.
- Effectuer une phase de test utilisateur réelle avant le déploiement final.
- Assurer la compatibilité avec différentes tailles d'écran et plateformes (tablette, téléphone).
- Élaborer un plan de communication pour présenter le projet au client et aux utilisateurs potentiels.
- Créer un manuel utilisateur simplifié pour faciliter la prise en main de l'application.

## 3. Contexte et Besoins

### 3.1 Problème à Résoudre

Dans la vie quotidienne, de nombreuses personnes achètent des produits alimentaires en grande quantité sans surveiller attentivement leurs dates de péremption.

Résultat : une partie de ces produits finit souvent à la poubelle, provoquant du **gaspillage alimentaire, une perte financière et un impact environnemental négatif.**

De plus, il n'existe pas d'outil simple, centralisé et accessible permettant de :

- suivre facilement les produits stockés à la maison,
- recevoir des rappels avant expiration,
- et obtenir des idées de recettes adaptées aux aliments disponibles.

Le projet **SmartPantry** vise à résoudre ce problème en proposant une **application mobile intelligente** qui aide les utilisateurs à mieux gérer leurs denrées, à éviter le gaspillage, et à optimiser leur consommation.

L'application permettra :

- le **scan automatique des produits** via code-barres ou QR code,
- la **gestion des dates d'expiration,**
- la **notification avant péremption,**
- et la **suggestion de recettes** pour utiliser les produits à temps.

Une version **compte familial** permettra aussi à plusieurs utilisateurs de gérer ensemble le même garde-manger numérique.

### 3.2 Parties Prenantes

[Liste des parties prenantes et leurs rôles.]

Partie prenante	Rôle / Fonction	Responsible (Responsable)	Accountable (Autorité)	Consulted (Consulté)	Informed (Informé)
Marwane Abouhamou	Product Owner	Définit les priorités, gère le backlog, valide les fonctionnalités clés.	✓	✓	✓
Youssef Najimi	Développeur Back-end	Développe la logique serveur, gère la base de données et les API.	✓	✓	✓
Abdellah Belgnaou	Développeur Front-end / UI-UX	Conçoit l'interface, intègre les fonctionnalités côté client et assure la fluidité d'utilisation.	✓	✓	✓
Professeur Ait El Cadi	Client / Encadrant	Valide les livrables, oriente les choix techniques et méthodologiques.		✓	✓
Utilisateurs finaux	Consommateurs / Familles	Fournissent des retours d'expérience sur l'application et ses fonctionnalités.		✓	✓
Services externes (APIs et outils)	Firebase, Spoonacular, etc.	Fournissent le stockage cloud, les données de recettes et la messagerie de notifications.			✓

## 4. Portée du Projet

### 4.1 Inclusions

- **Scan des produits** via code-barres ou QR code pour enregistrer automatiquement les informations du produit (nom, catégorie, date d'expiration).
- **Ajout manuel des produits** lorsque le code-barres n'est pas reconnu ou absent.
- **Gestion automatique des dates d'expiration** avec affichage des jours restants avant péremption.
- **Envoi de notifications** pour avertir l'utilisateur avant la date limite de consommation.
- **Affichage d'un tableau de bord (dashboard)** regroupant tous les produits classés par état

(frais, bientôt expiré, expiré).

- **Proposition de recettes personnalisées** à partir des produits proches de la date d'expiration.
- **Création et gestion de compte utilisateur**, avec possibilité de **mise à niveau vers un compte familial**.
- **Partage de profil familial** permettant à plusieurs membres d'accéder au même garde-manger numérique.
- **Gestion des rôles au sein du compte familial** (propriétaire, éditeur, lecteur).
- **Sauvegarde et synchronisation des données** dans le cloud (Firebase).
- **Consultation des statistiques d'utilisation**, comme le nombre de produits consommés à temps ou gaspillés.
- **Interface multilingue (français/anglais)** pour élargir l'accessibilité.
- **Mode hors ligne** pour consulter les produits déjà enregistrés sans connexion Internet.
- **Authentification sécurisée** et protection des données personnelles des utilisateurs.
- **Compatibilité multi-plateforme** (Android et iOS).

## 4.2 Exclusions

- **Pas de fonctionnalité d'achat ou de commande en ligne** : l'application ne permet pas de faire des courses ou de passer des commandes auprès de magasins.
- **Pas d'intégration avec des capteurs physiques** (balances connectées, réfrigérateurs intelligents, etc.).
- **Pas de reconnaissance automatique d'images** pour identifier visuellement les produits (fonction OCR prévue pour une version future).
- **Pas de mode de paiement ou de gestion budgétaire** lié aux produits achetés.
- **Pas de fonctionnalité de géolocalisation** des magasins ou promotions locales.
- **Pas d'enregistrement vocal ou commande vocale intégrée** (non prioritaire dans cette version).
- **Pas de gestion multi-profil complexes** (autres que la version familiale).
- **Pas de support technique en temps réel ou chat intégré**.
- **Pas de publicité ni de contenu sponsorisé** dans l'application initiale.
- **Pas d'intégration directe avec des plateformes de recettes externes nécessitant un abonnement payant**.

## 4.3 Contraintes

- **Budget nul**, le projet étant réalisé dans un cadre académique sans financement externe.
- **Délai de réalisation limité à 28 semaines**, incluant l'analyse, le développement, les tests et la présentation finale.
- **Ressources humaines limitées** à trois membres de l'équipe (Marwane Abouhamou, Youssef Najimi, Abdellah Belgnaou).
- **Technologies imposées** :
  - **React Native pour l'application mobile (Android et iOS)**.
  - **React.js pour le site web**.
  - **Firebase pour la base de données, l'authentification et la synchronisation en temps réel**.  
**Portée fonctionnelle définie** : seules les fonctionnalités essentielles doivent être développées dans la première version (MVP).
- **Dépendance à des services tiers** (APIs pour les recettes et la reconnaissance de codes-barres).
- **Absence de budget pour hébergement professionnel** : le site web doit être hébergé sur des plateformes gratuites (ex. Firebase Hosting ou Netlify).
- **Disponibilité du temps de travail restreinte** en raison des autres projets universitaires des membres.
- **Respect des règles de protection des données (RGPD)** et des politiques de sécurité Firebase.
- **Exigence de qualité et de stabilité** du produit avant la soutenance finale.
- **Langues prises en charge** : français et anglais uniquement dans la version initiale.

## 5. Exigences

## 5.1 Fonctionnelles (User Stories)

User Story	Priorité	Estimation (points)	Critères d'Acceptation
En tant qu'utilisateur, je veux <b>créer un compte personnel</b> afin d'accéder à mes produits et à mes paramètres.	Haute	3	L'inscription nécessite un email valide et une confirmation.
En tant qu'utilisateur, je veux <b>scanner le code-barres d'un produit</b> afin d'enregistrer automatiquement son nom et sa date d'expiration.	Haute	5	Le produit s'ajoute à la base avec les bonnes informations.
En tant qu'utilisateur, je veux <b>ajouter manuellement un produit</b> afin de gérer les articles sans code-barres.	Moyenne	3	Le produit apparaît dans la liste après validation.
En tant qu'utilisateur, je veux <b>recevoir une notification avant la date d'expiration</b> afin d'éviter le gaspillage.	Haute	4	Une alerte s'affiche X jours avant la date limite.
En tant qu'utilisateur, je veux <b>voir mes produits triés par statut</b> (frais, bientôt expiré, expiré) afin de prioriser leur utilisation.	Haute	4	Les produits sont correctement triés et colorés.
En tant qu'utilisateur, je veux <b>rechercher une recette</b> à partir de mes produits disponibles afin d'utiliser ceux proches de la péremption.	Haute	5	L'application affiche des recettes contenant les ingrédients correspondants.
En tant qu'utilisateur, je veux <b>mettre à niveau mon compte vers un compte familial</b> afin d'inviter d'autres membres à gérer les produits.	Haute	8	Les membres invités peuvent voir et modifier les produits partagés.
En tant que membre d'un compte familial, je veux <b>modifier ou supprimer des produits</b> afin de maintenir la base à jour.	Moyenne	5	Les changements sont visibles par tous les membres du compte.
En tant qu'utilisateur, je veux <b>accéder à mon compte depuis le site web ou l'application mobile</b> afin de gérer mes produits depuis plusieurs appareils.	Haute	5	Les données sont synchronisées entre le site web et l'application mobile.
En tant qu'utilisateur, je veux <b>me connecter avec mon email et mot de passe</b> afin de sécuriser mon accès.	Haute	3	Connexion réussie après vérification des identifiants.
En tant qu'utilisateur, je veux <b>supprimer un produit de ma liste</b> afin de garder un inventaire propre et à jour.	Moyenne	2	Le produit disparaît immédiatement de la liste.
En tant qu'utilisateur, je veux <b>voir des statistiques simples</b> (produits consommés, gaspillés, etc.) afin d'évaluer mes habitudes.	Basse	3	Les statistiques se mettent à jour automatiquement selon les données.

## 5.2 Non-Fonctionnelles

- **Performance** : le temps de réponse de l'application et du site web doit être inférieur à 2 secondes.
- **Sécurité** : authentification sécurisée (Firebase Auth), chiffrement des données utilisateurs.
- **Synchronisation** : toutes les données doivent être mises à jour en temps réel entre les appareils et le cloud.
- **Disponibilité** : l'application doit être fonctionnelle 24h/24 et 7j/7 (hors maintenance).
- **Compatibilité** :
  - Mobile : Android 10+ et iOS 13+.
  - Web : Chrome, Edge, Firefox, Safari.
- **Ergonomie** : interface claire, intuitive et accessible à tout public.
- **Multilingue** : application disponible en français et en anglais.
- **Accessibilité** : polices lisibles, couleurs contrastées, navigation fluide.
- **Scalabilité** : architecture modulaire pour faciliter l'ajout de nouvelles fonctionnalités (OCR, IA, etc.).
- **Confidentialité** : conformité avec le RGPD, pas de partage de données à des tiers.
- **Mode hors ligne** : accès possible aux produits enregistrés sans connexion Internet, avec synchronisation automatique au retour en ligne.
- **Maintenance** : facilité de mise à jour du code et documentation claire pour futurs contributeurs.

## 6. Méthodologie Agile

### 6.1 Framework

Le projet **SmartPantry** adopte le **framework Agile Scrum**, avec des **sprints de 3 semaines**.

Chaque sprint comprend la planification, le développement, les tests, la revue et la rétrospective.

L'objectif est de livrer à la fin de chaque sprint une version fonctionnelle et testée du produit (application mobile + site web).

### 6.2 Rôles

- **Product Owner : Marwane Abouhamou**  
Responsable de la définition des besoins et des priorités du produit. Il gère le backlog, communique avec le client (professeur encadrant) et valide les livrables.
- **Scrum Master : Youssef Najimi**  
Garant du bon déroulement de la méthodologie Scrum. Il anime les réunions Agile, aide à lever les obstacles et veille au respect du planning et des processus de l'équipe.

- **Équipe de Développement :**
  - **Abdellah Belgnaou** – Développeur Front-end / UI-UX : conception de l'interface et intégration graphique.
  - **Youssef Najimi** – Développeur Back-end : création et gestion de la base de données, intégration Firebase, API et sécurité.
  - **Marwane Abouhamou** – Coordination fonctionnelle et intégration mobile/web.

## 6.3 Cérémonies

- **Sprint Planning :**

Organisée au début de chaque sprint (toutes les 2 semaines). Durée : environ **2 heures**.  
Objectif : définir les tâches à réaliser et les objectifs à atteindre pour le sprint.
- **Daily Stand-up :**

Réunion courte quotidienne (15 minutes) chaque matin.  
Format : chaque membre partage ce qu'il a fait, ce qu'il va faire et les éventuels blocages.
- **Sprint Review :**

Tenue à la fin de chaque sprint (toutes les 2 semaines).  
Présentation au professeur encadrant du produit incrémenté, démonstration des fonctionnalités développées et collecte de retours.
- **Rétrospective :**

Immédiatement après la Sprint Review (durée : 30 minutes).  
Objectif : identifier les points positifs, les difficultés rencontrées et les axes d'amélioration pour le sprint suivant.

## **7. Planning**

### **— Phase 1 : Étude et Conception (Semaines 1 à 4)**

- Activités :**

- Analyse des besoins fonctionnels et techniques.
- Recherche sur les APIs (Firebase, Spoonacular, etc.).
- Conception de l'architecture globale (mobile, web, base de données).
- Création des maquettes graphiques (Figma).

- Livrables :**

- Cahier des charges complet.
  - Maquettes validées et architecture logicielle définie.
- 

### **— Sprint 1 : Initialisation et Configuration (Semaines 5 à 8)**

- Objectifs :**

- Mise en place des environnements de développement (React Native, React.js, Firebase).
- Configuration de la base de données et du stockage cloud.
- Création des premières interfaces d'accueil et d'inscription.

- Livrable :**

- Application et site web initialisés avec un système d'inscription fonctionnel.
- 

### **— Sprint 2 : Gestion des Produits (Semaines 9 à 12)**

- Objectifs :**

- Développement de la fonctionnalité de scan (code-barres / QR code).
- Ajout manuel des produits.
- Enregistrement et affichage des produits dans la base de données.

- Livrable :**

- Première version fonctionnelle du module “gestion des produits”.
- 

### **— Sprint 3 : Dates et Notifications (Semaines 13 à 16)**

- Objectifs :**

- Intégration de la gestion des dates d'expiration.

- Développement du système de notifications automatiques avant péréemption.
  - Tableau de bord avec tri par statut (frais, bientôt expiré, expiré).
  - **Livrable :**
    - Application envoyant des rappels automatiques aux utilisateurs.
- 

— **Sprint 4 : Recettes et Suggestions (Semaines 17 à 20)**

- **Objectifs :**
    - Connexion à l'API Spoonacular pour afficher des recettes adaptées aux produits.
    - Intégration d'un moteur de recherche et filtres (type de plat, allergènes, etc.).
    - Test de la cohérence entre produits et suggestions.
  - **Livrable :**
    - Module “recettes intelligentes” pleinement opérationnel.
- 

— **Sprint 5 : Comptes Familiaux et Collaboration (Semaines 21 à 24)**

- **Objectifs :**
    - Ajout de la mise à niveau vers un compte familial.
    - Système d'invitation et gestion des rôles (propriétaire, éditeur, lecteur).
    - Synchronisation complète entre les membres d'un même compte.
  - **Livrable :**
    - Fonction “compte familial” disponible et testée.
- 

— **Sprint 6 : Tests, Sécurité et Optimisation (Semaines 25 à 27)**

- **Objectifs :**
    - Réalisation de tests unitaires et d'intégration.
    - Vérification de la sécurité des données (authentification Firebase, RGPD).
    - Amélioration des performances et correction des bugs.
  - **Livrable :**
    - Version stable, rapide et sécurisée prête pour déploiement.
- 

— **Phase Finale : Déploiement et Soutenance (Semaine 28)**

- Activités :**
  - Déploiement du site web (Firebase Hosting / Netlify).
  - Génération de l'APK de l'application mobile.
  - Rédaction de la documentation technique et du guide utilisateur.
  - Préparation de la soutenance (présentation, démo).
- Livrables :**
  - Application mobile et site web en production.
  - Documentation complète.
  - Présentation finale du projet.

## 8. Gestion des Risques

Risque	Probabilité	Impact	Mitigation (Mesures préventives)
<b>Retard technique dans le développement</b>	Moyenne	Haute	Suivi régulier du planning, bonne répartition des tâches et communication entre les membres.
<b>Problèmes d'intégration entre la version mobile et web</b>	Moyenne	Moyenne	Tests fréquents de synchronisation et utilisation d'une base commune (Firebase).
<b>Dépendance aux API externes (Spoonacular, Firebase)</b>	Moyenne	Moyenne	Prévoir une base locale de secours et tester les API en amont.
<b>Manque de temps pour les tests et corrections</b>	Moyenne	Haute	Réserver une période dédiée aux tests avant la soutenance (Sprint 6).
<b>Perte de données ou erreurs de sauvegarde</b>	Basse	Haute	Utiliser les sauvegardes automatiques et la sécurité intégrée de Firebase.

## 9. Livrables et Critères d'Acceptation

— **Livrable 1 : Application mobile SmartPantry (Android et iOS)**

- **Critères d'acceptation :**
  - L'application est installable et fonctionnelle sur Android et iOS.
  - Les principales fonctionnalités (scan, notifications, recettes, comptes familiaux) sont opérationnelles.
  - Les données sont synchronisées avec la base Firebase.

#### — Livrable 2 : Site web SmartPantry

- **Critères d'acceptation :**
  - Le site web est accessible en ligne (Firebase Hosting ou Netlify).
  - Les utilisateurs peuvent se connecter et accéder à leurs produits via la même base de données.
  - L'interface est responsive et compatible avec les navigateurs récents.

#### — Livrable 3 : Base de données et services cloud

- **Critères d'acceptation :**
  - Les données sont stockées de manière sécurisée sur Firebase.
  - Les sauvegardes automatiques fonctionnent correctement.
  - Les performances et la disponibilité sont stables.

#### — Livrable 4 : Documentation technique et fonctionnelle

- **Critères d'acceptation :**
  - Le document décrit l'architecture, les technologies, les APIs et la structure du code.
  - Les étapes d'installation et de déploiement sont claires.
  - La documentation est validée par le professeur encadrant.

#### — Livrable 5 : Manuel utilisateur

- **Critères d'acceptation :**
  - Le manuel explique de façon simple l'utilisation des fonctionnalités principales.
  - Les captures d'écran et instructions sont à jour.
  - Le guide est compréhensible pour tout type d'utilisateur.

#### — Livrable 6 : Présentation et soutenance finale

- **Critères d'acceptation :**
  - Présentation PowerPoint structurée et claire.
  - Démonstration fonctionnelle de l'application mobile et du site web.
  - Réponses aux questions techniques et fonctionnelles du jury.

## 10. Suivi

### — KPIs (Indicateurs de performance)

- **Vélocité de l'équipe** : mesure du nombre de user stories complétées à chaque sprint.
- **Taux de bugs** : suivi du nombre d'erreurs détectées et corrigées par sprint.
- **Respect du planning** : comparaison entre les tâches prévues et réalisées.
- **Taux de satisfaction des utilisateurs tests** : évaluation de la facilité d'utilisation et de la stabilité du système.
- **Taux de couverture des fonctionnalités** : pourcentage de fonctionnalités principales développées et validées.
- **Taux de réussite des tests unitaires** : suivi de la qualité du code et de la fiabilité des fonctionnalités.

### — Rapports et Outils de suivi

- **Rapport hebdomadaire** : rédigé par le Scrum Master pour résumer les avancements, les points bloquants et les objectifs du sprint suivant.
- **Burndown Chart** : graphique de progression permettant de visualiser le travail restant et la vitesse de réalisation.
- **Revue de Sprint (Sprint Review)** : présentation au professeur encadrant à la fin de chaque sprint pour démontrer les nouvelles fonctionnalités.
- **Rérospective** : réunion d'équipe en fin de sprint pour identifier les points forts et les axes d'amélioration.
- **Outils utilisés :**
  - **Trello / Jira** pour la gestion des tâches et du backlog.
  - **GitHub** pour le suivi du code source et des versions.
  - **Firebase Analytics** pour le suivi des performances et de l'utilisation réelle de l'application.