

Cahier des Charges - Projet Informatique Agile

NoWaste

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 :

NoWaste est un projet 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, mettre à jour la liste de courses pour optimiser les achats futurs. 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.
- **Réduire le gaspillage alimentaire** en incitant les utilisateurs à consommer les produits avant expiration.
- **Assurer la synchronisation des données** entre plusieurs appareils via le cloud .
- **Garantir la sécurité et la confidentialité** des données utilisateur (authentification sécurisée).
- **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 rapide 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.**
- **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, Discord).
- **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 large accessibilité.
- **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 **NoWaste** vise à résoudre ce problème en proposant un site web une **application mobile intelligente** qui aide les utilisateurs à mieux gérer leurs denrées, à éviter le gaspillage, et à optimiser leur consommation.

NB :

- **En France** : on jette en moyenne 63 kg par habitant de denrées encore comestibles chaque année (notre-environnement.gouv.fr)

- **En USA** : \$728 par personne/an de nourriture achetée mais non consommée (tous modes confondus). Pour un foyer de 4 personnes : environ \$2 913/année.

(<https://www.biocycle.net/epa-updates-data-on-cost-of-household-food-waste>)

3.2 Parties Prenantes

Partie prenante	Responsible (Responsable)	Responsible (Responsable)	Accountable (Autorité)	Consulted (Consulté)	Informed (Informé)
Marwane Abouhamou	Product Owner	Définit les priorités, gère le backlog, valide les fonctionnalités clés.	Pablo directeur	Client	Les Developpeurs / UI UX designers
Youssef Najimi	Développeur Back-end	Développe la logique serveur, gère la base de données et les API.	Directeur Technique	Product Owner	Directeur Technique
Abdellah Belgnaou	Développeur Front-end / UI-UX	Conçoit l'interface, intègre les fonctionnalités côté client et assure la fluidité d'utilisation.	Directeur Technique	UI / UX Designer	Directeur Technique
Utilisateur final	Utiliser application et donner un feedback	Valide les livrables, oriente les choix techniques et méthodologiques.	✓	✓	Equipe de support
Utilisateurs finaux	Consommateurs / Familles	Fournissent des retours d'expérience sur l'application et ses fonctionnalités.	✓	✓	✓
Services externes (APIs et outils)	Fournissent des services ou des données	Fournissent le stockage cloud, les données de recettes et la messagerie de notifications.	Développeur Back-end - Front-end	✓	✓

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.
- **Mise à jour automatique du stock** après consommation ou ajout d'un nouvel achat
- **Création automatique d'une liste de courses basée sur** Les produits manquants

- **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 (frais, bientôt expiré, expiré, unité, poids, catégorie).
- **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.
- **Historique des modifications** (qui a ajouté ou consommé un produit)
- **Gestion des rôles au sein du compte familial** (propriétaire, éditeur, lecteur).
- **Sauvegarde et synchronisation des données** dans le cloud .
- **Interface multilingue (français/anglais)** pour élargir l'accessibilité.
- **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 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 non encore étudiées :**
 - tauri pour l'application mobile (Android et iOS).
 - Next.js, Express.js, node.js pour le site web.
 - MongoDB pour la base de données
- **Dépendance à des services tiers** (APIs pour les recettes et la reconnaissance de codes-barres).
- **Disponibilité du temps de travail restreinte** en raison des autres projets universitaires des membres.
- **Exigence de qualité et de stabilité** du produit avant la soutenance finale.
- **Langues prises en charge** : français et anglais

5. Exigences

5.1 Fonctionnelles (User Stories)

User Story	Priorité	Estimation (points)	Critères d'acceptation
En tant qu'utilisateur, je veux scanner un code-barres ou un QR code pour enregistrer automatiquement un produit.	Haute	5	Le scan détecte le produit, récupère le nom, la catégorie et la date d'expiration si disponibles. L'utilisateur peut confirmer ou modifier les données avant l'ajout.
En tant qu'utilisateur, je veux ajouter manuellement un produit quand le code-barres est absent ou non reconnu.	Haute	3	L'utilisateur peut saisir le nom, la catégorie, la quantité et la date d'expiration. Le produit s'ajoute au stock après validation.
En tant qu'utilisateur, je veux que le stock soit mis à jour automatiquement après ajout ou consommation.	Haute	4	Le stock se met à jour en temps réel lors d'un ajout ou d'une suppression. Les données sont sauvegardées et reflètent l'état actuel du garde-manger.
En tant qu'utilisateur, je veux qu'une liste de courses soit générée automatiquement à partir des produits manquants.	Moyenne	5	L'application détecte les produits à zéro stock et les ajoute à la liste de courses. L'utilisateur peut la modifier manuellement.
En tant qu'utilisateur, je veux visualiser les dates d'expiration avec le nombre de jours restants.	Haute	3	Chaque produit affiche la date d'expiration et le nombre de jours restants avant péremption.
En tant qu'utilisateur, je veux recevoir une notification avant la date de péremption d'un produit.	Haute	5	Des alertes push ou e-mail sont envoyées X jours avant expiration (paramétrable). Les notifications s'affichent même en arrière-plan.
En tant qu'utilisateur, je veux un tableau de bord qui regroupe mes produits selon leur statut et catégorie.	Moyenne	8	Le dashboard affiche les produits classés (frais, bientôt expiré, expiré, par catégorie, unité ou poids). Données mises à jour dynamiquement.
En tant qu'utilisateur, je veux recevoir des recettes personnalisées basées sur mes produits proches de la date limite.	Moyenne	8	L'application propose des recettes adaptées aux produits à consommer rapidement. L'utilisateur peut filtrer par type de plat ou allergènes.

En tant qu'utilisateur, je veux pouvoir créer un compte personnel et éventuellement un compte familial.	Haute	5	L'inscription, connexion et gestion de compte sont disponibles. Option de mise à niveau vers un compte familial.
En tant qu'utilisateur familial, je veux partager mon garde-manger avec d'autres membres.	Haute	5	Plusieurs membres peuvent accéder au même stock en temps réel. Les modifications sont synchronisées entre profils.
En tant qu'utilisateur, je veux consulter l'historique des modifications pour savoir qui a ajouté ou consommé un produit.	Moyenne	4	Chaque action est enregistrée (ajout, suppression, modification) avec le nom du membre et la date.
En tant qu'administrateur de compte familial, je veux attribuer des rôles (propriétaire, éditeur, lecteur).	Moyenne	4	Le propriétaire peut gérer les droits d'accès des membres (ajout, suppression, lecture seule).
En tant qu'utilisateur, je veux que mes données soient sauvegardées et synchronisées dans le cloud.	Haute	6	Les données sont automatiquement synchronisées entre appareils connectés au même compte. Sauvegarde en cas de déconnexion.
En tant qu'utilisateur, je veux pouvoir utiliser l'application en plusieurs langues (FR/EN).	Basse	3	L'utilisateur peut changer la langue depuis les paramètres. Les textes sont traduits intégralement.
En tant qu'utilisateur, je veux une authentification sécurisée et une bonne protection de mes données.	Haute	8	Authentification chiffrée (OAuth2, JWT), stockage sécurisé des mots de passe et conformité RGPD.
En tant qu'utilisateur, je veux utiliser l'application sur Android et iOS.	Haute	8	L'application est compatible et testée sur les deux plateformes avec une interface identique et responsive.

5.2 Non-Fonctionnelles

- **Performance** : le temps de réponse de l'application et du site web doit être rapide.
- **Sécurité** : authentification sécurisée , 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 :**
- **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.).
- **Maintenance :** facilité de mise à jour du code et documentation claire pour futurs contributeurs.

6. Méthodologie Agile

6.1 Framework

Le projet **NoWaste** 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 :**
Responsable de la définition des besoins et des priorités du produit. Il gère le backlog, communique avec le client (professeur encadrant) qui valide les livrables.
- **Scrum Master :**
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.
 - **Marouane 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.
- **Backlog Grooming :** Mettre à jour et prioriser le backlog produit pour les prochains sprints.

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 (Next.js, Express.js, node.js, mongodb, tauri).
- Configuration de la base de données et du stockage cloud.
- Crédit des premières interfaces : accueil, inscription, authentification sécurisée.
- Mise en place de l'architecture multi-plateforme (Android et iOS).
- Préparation de la structure pour sauvegarde et synchronisation des données.

Livrable :

- Application et site web initialisés avec système d'inscription sécurisé et architecture cloud prête.

— Sprint 2 : Gestion des Produits et Stock (Semaines 9 à 12)

Objectifs :

- Développement de la fonctionnalité de scan (code-barres / 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.

- Mise à jour automatique du stock après consommation ou ajout d'un nouvel achat.
- Sauvegarde et affichage des produits dans la base de données.

Livrable :

- Module “gestion des produits” fonctionnel avec scan, ajout manuel et mise à jour automatique du stock.

— Sprint 3 : Dates d’expiration, Notifications et Dashboard (Semaines 13 à 16)

Objectifs :

- Gestion automatique des dates d’expiration et affichage des jours restants.
- Envoi de notifications avant la date limite de consommation.
- Création d’un tableau de bord regroupant tous les produits classés (frais, bientôt expiré, expiré, unité, poids, catégorie).

Livrable :

- Module “alertes et dashboard” pleinement opérationnel.

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

Objectifs :

- Intégration d’un moteur de recettes pour proposer des suggestions personnalisées à partir des produits proches de la date d’expiration.
- Filtrage des recettes par type de plat, allergènes, etc.
- Test de la cohérence entre les produits du stock et les suggestions de recettes.

Livrable :

- Module “recettes intelligentes” fonctionnel et connecté au stock de produits.

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

Objectifs :

- 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).
- Historique des modifications (qui a ajouté ou consommé un produit).

Livrable :

- Module “compte familial” complet et opérationnel.

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

Objectifs :

- Interface multilingue (français/anglais).
- Vérification de la sécurité des données et conformité RGPD.
- Tests unitaires et d’intégration.
- Optimisation des performances et correction des bugs.
- Validation finale de la compatibilité multi-plateforme (Android et iOS).

Livrable :

- Version finale stable, sécurisée, rapide, multilingue et 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)
Retard technique dans le développement (Sprint 1 et 2 : initialisation et gestion des produits)	Moyenne	Haute	Suivi régulier du planning, découpage précis des tâches, priorisation des dépendances critiques et communication constante entre les membres.
Problèmes d'intégration entre la version mobile et web (Sprint 1 et 2)	Moyenne	Moyenne	Définir une architecture commune dès le Sprint 1, tests fréquents de synchronisation, réutilisation de composants partagés et mise en place d'une base de données unique.
Dépendance aux API externes pour recettes ou notifications (Sprint 4 et 3)	Moyenne	Moyenne	Prévoir une base locale de secours, tester les API en amont et prévoir un plan B si l'API est indisponible.
Manque de temps pour les tests et corrections (Sprint 3 à 6)	Moyenne	Haute	Réserver des périodes dédiées aux tests et à la correction des bugs dans Sprint 6, automatiser les tests unitaires et d'intégration dès le Sprint 2.
Perte de données ou erreurs de sauvegarde (Sprint 2 et 5 : gestion produits et comptes familiaux)	Basse	Haute	Utiliser les sauvegardes automatiques, mise en place de versioning des données et test régulier de la synchronisation cloud.
Problèmes de sécurité (Sprint 1 et 6)	Moyenne	Haute	Mise en place dès Sprint 1 d'une authentification sécurisée, chiffrement des données, et audits de sécurité avant déploiement.
Difficulté de compatibilité multi-plateforme (Android/iOS) (Sprint 1 et 6)	Moyenne	Moyenne	Tests réguliers sur les deux plateformes dès Sprint 1, utilisation de composants multiplate-formes et correction rapide des incohérences UI/UX.
Difficulté d'adoption par les utilisateurs (ergonomie / UX) (Sprint 2 à 4)	Moyenne	Moyenne	Prototypage et tests utilisateurs dès Sprint 2, interface intuitive, feedback régulier et amélioration progressive du design et Dashboard.

9. Livrables et Critères d'Acceptation

— Livrable 1 : Cahier des charges complet et maquettes validées

Critères d'acceptation :

- Analyse des besoins fonctionnels et techniques documentée.
- Maquettes graphiques (Figma) validées par l'équipe .
- Architecture globale définie pour mobile, web et base de données.

— Livrable 2 : Application et site web initialisés avec système d'inscription

Critères d'acceptation :

- Application installable et fonctionnelle sur Android et iOS.
- Système d'inscription et authentification sécurisée opérationnel.
- Architecture cloud et base de données configurées pour la sauvegarde et synchronisation.

— Livrable 3 : Module “gestion des produits”

Critères d'acceptation :

- Scan des produits via code-barres ou QR code fonctionnel.
- Ajout manuel des produits possible lorsque le code-barres n'est pas reconnu.
- Mise à jour automatique du stock après consommation ou ajout d'un nouvel achat.
- Données stockées et synchronisées correctement .

— Livrable 4 : Module “alertes et Dashboard”

Critères d'acceptation :

- Gestion automatique des dates d'expiration avec affichage des jours restants.
- Envoi de notifications avant péremption fonctionnel.
- Tableau de bord regroupant tous les produits par statut, catégorie, unité et poids.

— Livrable 5 : Module “recettes intelligentes”

Critères d’acceptation :

- Suggestions de recettes basées sur les produits proches de la péréemption.
- Moteur de recherche et filtres opérationnels (type de plat, allergènes).
- Cohérence entre produits disponibles et suggestions vérifiée.

— Livrable 6 : Module “compte familial”

Critères d’acceptation :

- Création et mise à niveau vers un compte familial fonctionnel.
- Partage du garde-manger entre plusieurs membres synchronisé en temps réel.
- Gestion des rôles (propriétaire, éditeur, lecteur) opérationnelle.
- Historique des modifications consultable pour tous les membres.

— Livrable 7 : Version finale stable et sécurisée

Critères d’acceptation :

- Interface multilingue (français/anglais) fonctionnelle.
- Sécurité des données et conformité RGPD validées.
- Tests unitaires et d’intégration réalisés avec succès.
- Optimisation des performances et correction des bugs effectuées.
- Compatibilité Android/iOS confirmée.

— Livrable 8 : Application mobile et site web en production + Documentation +

Présentation

Critères d’acceptation :

- Déploiement de l’application mobile (APK) et du site web en ligne fonctionnels.
- Documentation technique et fonctionnelle complète et validée.
- Manuel utilisateur clair et à jour.
- Présentation et démonstration finale prêtes et validées pour la soutenance.

10. Suivi

Méthode de suivi

- **Burndown Chart** : visualise l'avancement des tâches par rapport au planning du sprint.
- **Réunion hebdomadaire (Stand-up / suivi d'équipe)** : point rapide sur l'avancement et les obstacles.
- **Outils utilisés :**
 - **Trello / Jira** pour la gestion des tâches et du backlog.
 - **GitHub** pour le suivi du code source et des versions.

KPIs obligatoires

- **Vélocité de l'équipe** : nombre de story points terminés par sprint.
- **Taux de bugs** : pourcentage de bugs détectés par rapport aux fonctionnalités livrées.

Rapports obligatoires

- **Rapport de sprint** : fin de chaque sprint, incluant les livrables atteints et les problèmes rencontrés.
- **Rapport final de projet** : résumé complet du projet avec livrables finaux et bilan des risques.