

Cahier des charges fonctionnel

1. Présentation du Projet

- **Nom du projet** : NewChef
- **Description** : Le projet consiste en la création d'une application utilisant Firebase pour gérer les recettes favorites des utilisateurs. L'application permettra aux utilisateurs de s'inscrire, de se connecter, et de gérer leurs recettes favorites (ajout, mise à jour, suppression et consultation).
- **Contexte et Justification** : Cette application s'adresse aux personnes qui souhaitent découvrir de nouvelles recettes de cuisine de manière rapide grâce au scan de codes-barres pour trouver des recettes. Notre application vise à satisfaire les cuisiniers occasionnels en leur permettant d'élargir leur panel de recettes de manière simple et intuitive, en utilisant des ingrédients qu'ils ont sous la main.

2. Objectifs du Projet

- Permettre aux utilisateurs de s'inscrire et de se connecter via un système d'authentification sécurisé utilisant Firebase Authentication.
- Offrir aux utilisateurs la possibilité d'ajouter, de mettre à jour, de supprimer et de consulter leurs recettes favorites.
- Assurer une communication fluide entre l'application et la base de données Firestore pour la gestion des recettes.
- Faciliter la découverte de nouvelles recettes grâce au scan de codes-barres des produits.

3. Fonctionnalités Principales

- **Authentification et Gestion des Comptes Utilisateurs**
 - **Inscription** : Les utilisateurs doivent pouvoir créer un compte en fournissant un email et un mot de passe.
 - **Connexion** : Les utilisateurs doivent pouvoir se connecter à leur compte en utilisant leur email et leur mot de passe.
 - **Déconnexion** : Les utilisateurs doivent pouvoir se déconnecter de leur compte.
- **Gestion des Recettes Favorites**
 - **Ajout de Recettes** : Les utilisateurs doivent pouvoir ajouter des recettes à leurs favoris. Chaque recette doit inclure les informations suivantes :
 - Identifiant de la recette (recipeld)
 - Titre de la recette
 - Ingrédients
 - Instructions
 - **Consultation des Recettes** : Les utilisateurs doivent pouvoir consulter leurs recettes favorites.

- **Mise à Jour des Recettes** : Les utilisateurs doivent pouvoir mettre à jour les informations d'une recette favorite existante.
- **Suppression de Recettes** : Les utilisateurs doivent pouvoir supprimer une recette de leurs favoris.
- **Recherche de Recettes par Scan de Code-Barres** : Les utilisateurs doivent pouvoir scanner un code-barres de produit pour obtenir des suggestions de recettes correspondantes.

4. Spécifications Techniques

- **Backend**
 - **Firebase Authentication** : Pour gérer l'inscription et la connexion des utilisateurs.
 - **Firestore** : Pour stocker et gérer les données des recettes favorites.
 - **Fonctions Firebase** : Pour implémenter la logique de gestion des recettes (ajout, mise à jour, suppression, consultation).
 - **Utilisation de l'API Spoonacular** : Pour rechercher des recettes.
 - **Utilisation de l'API Open Food Facts** : Pour rechercher des informations sur un produit.
 - **API RESTful**
 - **Endpoints d'authentification** :
 - POST /signup : Inscription des utilisateurs
 - POST /login : Connexion des utilisateurs
 - **Endpoints de gestion des recettes** :
 - GET /getFavoriteRecipes : Consultation des recettes favorites
 - POST /addFavoriteRecipe : Ajout d'une recette favorite
 - PUT /updateFavoriteRecipe : Mise à jour d'une recette favorite
 - DELETE /deleteFavoriteRecipe : Suppression d'une recette favorite
 - **Endpoints de recherche par scan de code-barres** :
 - GET /scanBarcode : Recherche de recettes par code-barres
- **Sécurité**
 - Utilisation de tokens JWT pour sécuriser les endpoints d'API.
 - Validation des données reçues pour éviter les injections et autres attaques.
- **Documentation API**
 - Utilisation de Swagger pour documenter l'API et permettre une interaction facile avec les endpoints.

5. Étapes du Projet

- **Conception**
 - Définition des modèles de données pour les utilisateurs et les recettes.
 - Conception des endpoints d'API et de la structure de la base de données Firestore.
- **Développement**

- Implémentation des endpoints d'authentification et de gestion des recettes.
- Intégration de Firebase Authentication et Firestore.
- Développement des fonctions Firebase pour la gestion des recettes.
- Implémentation de la fonctionnalité de scan de codes-barres.
- **Tests**
 - Tests unitaires et d'intégration pour chaque endpoint.
 - Tests de performance pour assurer la scalabilité de l'application.
- **Déploiement**
 - Déploiement des fonctions Firebase et de la base de données Firestore.
 - Configuration des règles de sécurité Firebase.
- **Documentation**
 - Rédaction de la documentation utilisateur.
 - Finalisation de la documentation Swagger pour l'API.

6. Livrables

- Code source de l'application
- Documentation de l'API Swagger
- Documentation utilisateur

7. Contraintes

- Respect des bonnes pratiques de développement et de sécurité.
- Respect des délais fixés pour chaque étape du projet.
- Tests rigoureux pour assurer la qualité et la fiabilité de l'application.
- Conformité aux réglementations de protection des données.

8. Risques

- Problèmes de sécurité liés à l'authentification et à la gestion des données sensibles.
- Délais de réponse de la base de données Firestore en cas de forte charge.
- Bugs et erreurs pouvant survenir lors de l'intégration des différentes composantes du système.
- Dépendance aux API externes pour les recettes et les informations sur les produits.

9. Annexes

- Diagramme de classe pour illustrer le fonctionnement de l'application.
- Exemples de requêtes et de réponses API pour chaque endpoint.
- Wireframes et prototypes pour illustrer l'interface utilisateur.