**📄 Cahier des charges – *"SmartTontine"***

**🧾 1. Présentation du projet**

**Nom : SmartTontine****Type : Application mobile (Android/iOS) + API back-end  
Objectif : Permettre à des utilisateurs d'organiser, suivre et automatiser des tontines et épargnes collectives depuis leur téléphone.**

**🧑‍🤝‍🧑 2. Utilisateurs ciblés**

* **Membres d'une tontine**
* **Organisateurs de groupes d’épargne**
* **Petits commerçants, artisans, employés**

**🔐 3. Fonctionnalités principales**

**👤** **Authentification**

* **Inscription / Connexion**
* **Validation par email ou SMS (optionnelle)**
* **Récupération de mot de passe**

**👥 Gestion des tontines**

* **Créer un groupe de tontine**
* **Inviter des membres**
* **Définir les règles : montant, fréquence, nombre de membres**
* **Attribuer un tour de collecte automatiquement**

**📆 Échéancier & suivi**

* **Calendrier de cotisations**
* **Notifications des tours à venir**
* **Statut des membres (à jour, en retard…)**

**💸** **Opérations financières**

* **Simulation ou intégration Wave/Orange Money**
* **Saisie manuelle des cotisations (si hors ligne)**
* **Journal des transactions (audit log)**

**📊 Tableau de bord**

* **Solde total**
* **Historique des tontines**
* **Graphique des contributions**
* **Alertes / Statut des groupes**

**⚙️ Autres**

* **Profil utilisateur (photo, nom, infos bancaires)**
* **Sécurité des données (chiffrement + RGPD)**
* **Export PDF / Excel du journal**

**🧱 4. Architecture technique**

**Mobile**

* **Flutter (iOS/Android)**
* **UI responsive, moderne**
* **Stockage local en cache (Hive, SharedPrefs)**

**Back-end**

* **Node.js + Express ou Django REST**
* **Base de données : PostgreSQL ou MongoDB**
* **Authentification JWT**
* **API REST sécurisée**

**Déploiement**

* **Firebase pour push notifications**
* **Render/Heroku pour API**
* **GitHub pour versioning**

**🧪 5. Contraintes**

* **Utilisation possible sans connexion (mode hors-ligne)**
* **Interface simple, intuitive**
* **Aucune transaction réelle, mais système prêt pour l’intégration API mobile money**

**📦 Livrables attendus**

* **Maquettes Figma**
* **Documentation technique + API**
* **Code source versionné**
* **APK de test + vidéo démo**
* **Cahier de test (QA)**

**✅ Backlog initial – *Sprint Planning Agile (comme en entreprise)***

**🎯 Sprint 0 – Préparation**

* **Rédiger le cahier des charges**
* **Créer les maquettes sur Figma**
* **Initialiser le repo GitHub**
* **Choisir les technos finales**
* **Préparer l’architecture du projet (API + Flutter)**

**🚀 Sprint 1 – Auth & Création de groupe**

* **Authentification (Inscription / Connexion)**
* **Page d’accueil (Dashboard vide)**
* **Création de tontine (formulaire)**
* **Affichage des groupes créés**
* **Test back-end Auth + POST groupe**

**📅 Sprint 2 – Gestion des membres & cotisations**

* **Ajouter/inviter des membres**
* **Gestion du tour (ordre automatique)**
* **Enregistrement d’un paiement**
* **Statut des membres (à jour/en retard)**
* **Notifications à envoyer (push test)**

**📊 Sprint 3 – Journal & tableau de bord**

* **Journal des opérations**
* **Statistiques (barre/ligne/pie avec Chart)**
* **Solde épargné**
* **Export PDF ou Excel**

**🧪 Sprint 4 – Finition & tests**

* **Tests UI + API**
* **Améliorations visuelles (Dark Mode, responsive)**
* **Corrections des bugs**
* **Génération de l’APK**

**🧩 2.** **Structuration d’une équipe projet – *SmartTontine***

Même si tu es seul(e), **tu peux simuler chaque rôle** comme si tu faisais partie d’une vraie équipe. Cela renforce ta vision projet et ton CV 💪

**🎯** **Objectif d’une bonne structuration :**

* Organiser les responsabilités
* Mieux gérer les tâches
* Séparer la technique du fonctionnel
* Réfléchir comme une vraie entreprise

**👥 Équipe projet type *(simulation en solo ou en binôme)***

| **Rôle** | **Responsabilités** |
| --- | --- |
| **Product Owner (PO)** | - Écrit les besoins métier- Priorise les fonctionnalités- Rédige les user stories |
| **Scrum Master** | - Organise les sprints- Gère le planning & le backlog- S’assure que l’équipe avance |
| **UI/UX Designer** | - Crée les maquettes sur Figma- Réfléchit à l’expérience utilisateur- Assure la cohérence visuelle |
| **Développeur Front** | - Développe l’interface mobile (Flutter)- Gère les écrans, navigation, design |
| **Développeur Back** | - Crée l’API REST (Node.js ou Django)- Gère la base de données & la logique métier |
| **Testeur QA** | - Rédige des cas de test- Vérifie les bugs- S’assure que tout fonctionne avant livraison |
| **DevOps / Release Manager** | - Prépare le déploiement API / app mobile- Génère l’APK / déploie sur Firebase ou Vercel |

**📂 Exemple de structure de dossier (comme une vraie équipe)**

/smart-tontine/

│

├── frontend/ # App Flutter

│ ├── lib/

│ └── assets/

│

├── backend/ # API Express / Django

│ ├── controllers/

│ ├── models/

│ └── routes/

│

├── docs/ # Documentation projet

│ ├── cahier-des-charges.md

│ ├── user-stories.md

│ └── wireframes/

│

├── test/ # Cas de test, scripts Postman

│

└── .git/ # Git versioning