**📘 Cahier des charges**

**Application Web de gestion des retraits et dépôts Mobile Money (MTN, Moov, Celtis)**

**Technologie principale : Vue.js (frontend) + Laravel ou Node.js (backend)**

**🧭 1. Objectif du projet**

Créer une **application web de gestion** permettant d’enregistrer, suivre et consulter les opérations de **retraits, dépôts et transferts** effectuées via les opérateurs **MTN, Moov** et **Celtis**.

L’application vise à :

* Simplifier la gestion quotidienne des transactions.
* Éviter les erreurs de saisie et de calcul manuel.
* Centraliser toutes les opérations dans une base de données unique.
* Générer des rapports journaliers et mensuels.

**👥 2. Public cible**

* Gérants de points mobile money.
* Agents de dépôt et retrait.
* Petites agences ou boutiques proposant des services MTN, Moov, Celtis.

**⚙️ 3. Fonctionnalités principales**

**🔹 A. Gestion des transactions**

* Enregistrer une transaction avec les champs suivants :
  + Nom du client
  + Numéro de téléphone
  + Type d’opération : *Retrait / Dépôt / Transfert*
  + Opérateur : *MTN / Moov / Celtis*
  + Montant
  + Date et heure automatiques
  + Observations (facultatif)
* Modifier ou supprimer une transaction existante.
* Rechercher une transaction par nom, numéro, date, type ou opérateur.

**🔹 B. Tableau de bord**

* Afficher :
  + Le nombre total de transactions du jour.
  + Le montant total des dépôts, retraits, et le solde du jour.
  + Des graphiques (barres ou camembert) pour visualiser les statistiques.
* Filtrer les transactions par jour, semaine, mois ou opérateur.

**🔹 C. Rapports et exports**

* Génération automatique de rapports :
  + Journalier, hebdomadaire, mensuel.
* Exportation au format :
  + **PDF** (pour impression)
  + **Excel (.xlsx)** (pour archivage)
* Affichage des totaux dans chaque rapport.

**🔹 D. Gestion des utilisateurs (optionnel)**

* Connexion / déconnexion avec mot de passe.
* Rôles :
  + **Admin** → accès total
  + **Agent** → accès limité aux transactions
* Journalisation des activités (historique des actions).

**🔹 E. Paramètres**

* Gérer la liste des opérateurs (MTN, Moov, Celtis).
* Gérer les taux de commission ou frais (facultatif).
* Modifier les informations de l’entreprise ou du point mobile money.

**🧱 4. Architecture technique**

| **Couche** | **Technologie recommandée** |
| --- | --- |
| Frontend (interface utilisateur) | **Vue.js 3** (avec Vite, TailwindCSS pour le style) |
| Backend (API REST) | **Laravel 11** ou **Node.js (Express)** |
| Base de données | **MySQL** ou **SQLite** |
| Authentification | JWT ou Sanctum |
| Graphiques | Chart.js ou ApexCharts |
| Export PDF | jsPDF |
| Export Excel | SheetJS (xlsx) |
| Hébergement | Hostinger / Render / Railway / Fly.io |

**🧩 5. Structure du projet**

**🌐 Frontend (Vue.js)**

**Pages prévues :**

1. **Connexion / Inscription**
2. **Dashboard**
3. **Liste des transactions**
4. **Formulaire d’ajout / modification**
5. **Rapports**
6. **Paramètres**

**⚙️ Backend (API RESTful)**

**Routes principales :**

| **Méthode** | **Route** | **Description** |
| --- | --- | --- |
| POST | /api/login | Connexion utilisateur |
| GET | /api/transactions | Lister toutes les transactions |
| POST | /api/transactions | Ajouter une transaction |
| PUT | /api/transactions/:id | Modifier une transaction |
| DELETE | /api/transactions/:id | Supprimer une transaction |
| GET | /api/reports | Générer un rapport PDF/Excel |

**🧠 6. Fonctionnalités à venir (évolutives)**

* Version mobile (via **Capacitor**)
* Notifications automatiques (ex : total du jour envoyé par mail/SMS)
* Synchronisation cloud multi-appareils
* Mode hors-ligne (PWA)

**🎨 7. Charte graphique**

* **Couleurs principales :**
  + Jaune MTN (#ffcb05)
  + Bleu Moov (#0098d9)
  + Vert Celtis (#00a859)
* **Design :** simple, professionnel et responsive.
* **Police :** Poppins ou Inter.

**🧾 8. Livrables**

* Code source complet (frontend + backend)
* Base de données prête à l’emploi
* Documentation technique d’installation
* Guide d’utilisation pour l’utilisateur final

**📅 9. Étapes de réalisation (plan de travail)**

| **Étape** | **Tâche** | **Durée estimée** |
| --- | --- | --- |
| 1 | Rédaction du cahier des charges | 1 jour |
| 2 | Création du design (UI/UX maquette) | 2 jours |
| 3 | Développement du backend (API) | 4 jours |
| 4 | Développement du frontend (Vue.js) | 5 jours |
| 5 | Liaison frontend-backend | 2 jours |
| 6 | Tests et corrections | 2 jours |
| 7 | Déploiement et documentation | 1 jour |
| **Total estimé** | — | **~17 jours** |

**🔐 10. Contraintes et sécurité**

* Protection des données (utilisateurs et transactions).
* Sauvegarde automatique quotidienne.
* Accès sécurisé via HTTPS.
* Validation des champs côté front et back.

**📈 11. Objectif final**

Avoir une application :

* Simple à utiliser, même sans grande connaissance informatique.
* Sécurisée et rapide.
* Accessible sur tous les appareils (ordinateur, téléphone, tablette).
* Facilement extensible vers une version mobile Android plus tard.