**🧠 Cahier d’analyse fonctionnelle**

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

**🧭 1. Contexte et objectif**

Les points Mobile Money effectuent quotidiennement de nombreuses transactions (retraits, dépôts, transferts).  
Ces opérations sont souvent consignées sur papier ou dans des carnets, ce qui rend difficile :

* le suivi des montants,
* le calcul des totaux journaliers,
* et la consultation rapide des historiques.

**Objectif de l’application :**  
Créer une plateforme web intuitive pour **enregistrer, consulter et analyser les transactions Mobile Money** effectuées par un point de service.

**👥 2. Acteurs du système**

| **Acteur** | **Description** | **Rôle principal** |
| --- | --- | --- |
| **Administrateur** | Responsable principal du point Mobile Money | Gère les agents, configure les paramètres, consulte tous les rapports |
| **Agent de caisse** | Effectue les opérations (retrait/dépôt) | Enregistre les transactions quotidiennes |
| **Utilisateur invité (optionnel)** | Peut uniquement visualiser les données (sans modification) | Consultation simple |

**⚙️ 3. Besoins fonctionnels**

**🔹 3.1. Gestion des transactions**

* Ajouter une nouvelle transaction (nom, numéro, type, opérateur, montant, date automatique).
* Modifier ou supprimer une transaction existante.
* Afficher toutes les transactions dans un tableau dynamique.
* Rechercher une transaction par nom, numéro ou opérateur.
* Filtrer les transactions par :
  + Date
  + Type (retrait, dépôt)
  + Opérateur (MTN, Moov, Celtis)

**🔹 3.2. Tableau de bord (Dashboard)**

* Afficher :
  + Le nombre total de transactions du jour.
  + Le total des retraits et dépôts.
  + Le solde du jour.
  + Des graphiques (transactions par opérateur, par jour).

**🔹 3.3. Rapports**

* Générer un **rapport PDF ou Excel** par :
  + Jour
  + Semaine
  + Mois
* Chaque rapport contient :
  + La liste des transactions
  + Les totaux par type et opérateur
  + La date de génération

**🔹 3.4. Authentification et gestion des utilisateurs**

* Connexion avec identifiant et mot de passe.
* Gestion des rôles :
  + **Admin** → accès total
  + **Agent** → accès restreint
* Historique d’activité (connexion, ajout, suppression, modification).

**🔹 3.5. Paramétrage du système**

* Ajouter / modifier les opérateurs (MTN, Moov, Celtis).
* Définir les taux de commission (optionnel).
* Modifier le profil du point Mobile Money (nom, adresse, logo).

**🧩 4. Besoins non fonctionnels**

| **Critère** | **Description** |
| --- | --- |
| **Performance** | L’application doit être fluide même avec plusieurs milliers de transactions. |
| **Sécurité** | Authentification obligatoire, validation des formulaires, HTTPS. |
| **Ergonomie** | Interface claire, responsive, adaptée à PC et mobile. |
| **Disponibilité** | Accessible en ligne 24h/24, hébergement stable. |
| **Extensibilité** | Possibilité d’ajouter de nouveaux opérateurs ou fonctionnalités. |

**🗃️ 5. Modèle de données (MCD simplifié)**

**🧱 Entités principales :**

**1️⃣ Transaction**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id | int | Identifiant unique |
| nom\_client | string | Nom complet du client |
| numero | string | Numéro de téléphone |
| type | string | Retrait / Dépôt / Transfert |
| operateur | string | MTN / Moov / Celtis |
| montant | float | Montant de la transaction |
| date\_transaction | datetime | Date et heure |
| created\_by | int | ID de l’utilisateur ayant enregistré |

**2️⃣ Utilisateur**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id | int | Identifiant unique |
| nom | string | Nom de l’utilisateur |
| email | string | Email ou identifiant |
| password | string | Mot de passe chiffré |
| role | string | Admin / Agent |

**🧩 Relations :**

* **Un utilisateur** effectue plusieurs **transactions**  
  → Relation **1-N** entre *Utilisateur* et *Transaction*.

**🧠 6. Cas d’utilisation (UML simplifié)**

| **ID** | **Cas d’utilisation** | **Acteur principal** | **Description** |
| --- | --- | --- | --- |
| UC01 | Se connecter | Utilisateur | L’utilisateur s’authentifie pour accéder à l’application. |
| UC02 | Ajouter une transaction | Agent / Admin | Enregistre une nouvelle opération. |
| UC03 | Modifier une transaction | Admin | Corrige une erreur dans une transaction. |
| UC04 | Supprimer une transaction | Admin | Retire une transaction erronée. |
| UC05 | Rechercher / filtrer une transaction | Tous | Permet de retrouver rapidement une opération. |
| UC06 | Générer un rapport | Admin | Produit un rapport PDF/Excel par période. |
| UC07 | Gérer les utilisateurs | Admin | Ajoute ou modifie un agent. |

**🎨 7. Interfaces prévues (IHM)**

| **Écran** | **Description** | **Éléments principaux** |
| --- | --- | --- |
| **Page de connexion** | Authentification utilisateur | Formulaire (email + mot de passe) |
| **Dashboard** | Vue globale des statistiques | Graphiques, totaux, résumé du jour |
| **Transactions** | Liste complète des opérations | Tableau dynamique avec filtres |
| **Formulaire d’ajout** | Enregistrement d’une transaction | Champs de saisie + bouton Enregistrer |
| **Rapports** | Historique + exports PDF/Excel | Filtres de période + boutons d’export |
| **Paramètres** | Gestion des opérateurs et utilisateurs | Formulaires et liste des agents |

**🧱 8. Architecture technique**

| **Couche** | **Technologie** |
| --- | --- |
| **Frontend** | Vue.js 3 + TailwindCSS |
| **Backend** | Laravel 11 (ou Node.js Express) |
| **Base de données** | MySQL |
| **API** | RESTful (JSON) |
| **Authentification** | Laravel Sanctum / JWT |
| **Hébergement** | Hostinger / Railway / Render |
| **Build** | Vite |

**🔐 9. Sécurité**

* Chiffrement des mots de passe (bcrypt).
* Validation côté frontend et backend.
* Middleware d’authentification.
* HTTPS pour la transmission des données.
* Journalisation des actions sensibles.

**📅 10. Planning prévisionnel**

| **Phase** | **Description** | **Durée estimée** |
| --- | --- | --- |
| 1 | Analyse et modélisation (MCD, MLD, routes API) | 2 jours |
| 2 | Création du backend (API Laravel/Node) | 4 jours |
| 3 | Développement du frontend (Vue.js) | 5 jours |
| 4 | Liaison front/back + tests | 3 jours |
| 5 | Rapports, exports, finalisation | 2 jours |
| 6 | Hébergement & déploiement | 1 jour |

**Durée totale estimée : ~17 jours**

**📈 11. Perspectives d’évolution**

* 📱 Version mobile (via Capacitor)
* 💬 Notifications automatiques (WhatsApp ou SMS)
* ☁️ Sauvegarde cloud et synchronisation multi-points
* 💰 Gestion automatique des commissions et bilans financiers