

Cahier des charges – Application Rimgaz

1. Présentation du projet
2. Objectifs principaux
3. Périmètre fonctionnel
4. Acteurs et rôles
5. Parcours principaux
 - 5.1. Création et gestion d'un client
 - 5.2. Demande de livraison / échange de bouteille
 - 5.3. Tournée du bus
 - 5.4. Paiement avec envoi de reçu
 - 5.5. Suivi du compte client
6. Exigences fonctionnelles détaillées
 - 6.1. Gestion des clients
 - 6.2. Wallet & paiements (MRU)
 - 6.3. Gestion des bouteilles et des dépôts
 - 6.4. Tournées de bus et chauffeurs
 - 6.5. Suivi en temps réel des bus (cartographie)
 - 6.6. Historique des positions et itinéraires (type Uber)
 - 6.7. Administration & reporting
 - 6.8. Notifications & communication
7. Exigences non fonctionnelles
8. UX adaptée aux personnes âgées ou peu familières avec la technologie
9. Architecture technique proposée
10. Évolutions possibles

Cahier des charges – Application Rimgaz

1. Présentation du projet

- **Nom de l'application** : Rimgaz
- **Activité** : Vente et distribution de bouteilles de gaz.
- **Objectif général** : Digitaliser la gestion des clients, de leurs bouteilles, de leur portefeuille (wallet) en MRU et des tournées de bus de distribution, avec suivi cartographique en temps réel et historique des itinéraires (type Uber).
- **Plateformes cibles** :
 - Application web (back-office + éventuellement portail client).
 - Application mobile Flutter (chauffeurs/livreurs et, à terme, clients).
 - Backend/API en Django.

2. Objectifs principaux

- Connaître en temps réel, pour chaque client :
 - Le nombre de bouteilles de gaz détenues (pleines / vides / en dépôt).
 - Sa situation financière : solde du wallet en MRU, dettes, paiements en attente.
- Simplifier et sécuriser le travail des bus de distribution :
 - Organisation des tournées par secteurs/zones.
 - Suivi des livraisons, des reprises de bouteilles et des encaissements.
- Offrir aux clients :

- La possibilité de commander / demander une livraison.
- La possibilité de payer (mobile money, virement, dépôt) et d'envoyer une capture/photo du reçu.
- Une mise à jour automatique de leur compte après validation des paiements.
- Fournir un **suivi cartographique temps réel** des bus, avec **historique des itinéraires** (visualisation type Uber).
- Proposer une interface **bilingue français / arabe**, adaptée aux personnes âgées ou peu familières avec la technologie : boutons larges, peu d'options par écran, texte lisible, icônes parlantes.

3. Périmètre fonctionnel

Le projet couvre les fonctions suivantes :

- Gestion des comptes clients.
- Gestion du wallet/client (portefeuille en MRU) et de l'historique des paiements.
- Gestion des bouteilles, des dépôts et des stocks (dépôts, bus, clients).
- Gestion des tournées de bus et des chauffeurs.
- Suivi en temps réel des bus sur carte + historique des itinéraires.
- Gestion des paiements avec upload de reçus et validation.
- Tableaux de bord et reporting (volumes, recettes, clients en retard de paiement, performances des bus).
- Notifications au client (SMS, WhatsApp, push, e-mail selon faisabilité).
- Administration des utilisateurs et des rôles (droits d'accès).

4. Acteurs et rôles

- **Client** (particulier ou professionnel) :
 - Consulte son solde et ses bouteilles.
 - Demande des livraisons.
 - Effectue des paiements et envoie des reçus.
- **Chauffeur / Livreurs (Bus)** :
 - Utilise l'application mobile pour consulter ses tournées.
 - Enregistre les livraisons, reprises de bouteilles, paiements sur place.
 - Envoie automatiquement sa position GPS.
- **Opérateur back-office / Caissier** :
 - Crée et met à jour les fiches clients.
 - Vérifie les reçus de paiement et les valide/rejette.
 - Suit les comptes clients et corrige les anomalies.
- **Responsable des tournées / Logisticien** :
 - Planifie les tournées des bus par secteurs.
 - Suit en temps réel la position des bus et l'avancement des tournées.
 - Analyse les performances (temps de tournée, nombre de clients servis, etc.).
- **Administrateur système** :
 - Gère les utilisateurs, rôles et permissions.
 - Paramètre l'application (tarifs, secteurs, types de bouteilles, moyens de paiement, etc.).

5. Parcours principaux

5.1. Création et gestion d'un client

- Création par le back-office (et plus tard, auto-inscription possible côté client).
- Informations à saisir :
 - Nom, prénom / raison sociale.
 - Téléphone(s), WhatsApp éventuel.
 - Adresse détaillée (quartier, ville, points de repère).

- Localisation GPS (facultatif mais recommandé).
- Type de client (ménage, commerce, restaurant, etc.).
- Langue d’interface (FR/AR).
- Modification, archivage, changement de statut (actif, suspendu, en retard de paiement, etc.).

5.2. Demande de livraison / échange de bouteille

- Le client demande :
 - Livraison d’une nouvelle bouteille, ou
 - Échange d’une bouteille vide contre une pleine.
- La demande est liée à :
 - Sa fiche client (identité, coordonnées).
 - Son adresse/localisation.
- Affectation de la demande à une tournée de bus (manuelle au départ, optimisation possible plus tard).

5.3. Tournée du bus

- Le chauffeur se connecte sur l’application mobile.
- Il consulte ses tournées du jour :
 - Pour chaque tournée : liste des clients à visiter, avec ordre proposé.
- Chez chaque client :
 - Enregistrement des bouteilles livrées (pleines) et récupérées (vides).
 - Saisie des paiements reçus (cash / mobile money sur place, si applicable).
 - Mise à jour instantanée de l’état du client et du stock du bus.

5.4. Paiement avec envoi de reçu

- Le client paye via un canal externe (mobile money, banque, dépôt en agence, etc.).
- Sur l’application (web/mobile) :
 - Saisie du montant payé (en MRU).
 - Sélection du mode de paiement.
 - Upload de la capture/photo du reçu.
- Le back-office :
 - Vérifie la validité du reçu (montant, référence, date…).
 - Valide ou rejette la demande :
 - Si validé : crédit du wallet du client, mise à jour de sa dette éventuelle.
 - Si rejeté : indication du motif (erreur montant, reçu illisible, etc.) notifiée au client.

5.5. Suivi du compte client

- Pour chaque client, l’interface affiche :
 - Solde wallet en MRU (positif, nul ou négatif si dette).
 - Historique des mouvements (facturations, paiements, ajustements).
 - Nombre et type de bouteilles actuellement en sa possession (par capacité : 6 kg, 12 kg, etc.).
 - Synthèse des dernières livraisons et paiements.

6. Exigences fonctionnelles détaillées

6.1. Gestion des clients

- Création, modification, archivage des fiches clients.
- Champs principaux :
 - Identité : nom, prénom / raison sociale.
 - Coordonnées : téléphone(s), WhatsApp si utilisé.
 - Adresse postale et description du lieu (quartier, repère visuel).

- Localisation GPS (si disponible).
 - Type de client (ménage, commerce, restaurant, etc.).
 - Langue préférée (FR/AR).
 - Statut (actif, suspendu, en retard de paiement...).
- Recherche multi-critères : nom, téléphone, secteur, type, statut.

6.2. Wallet & paiements (MRU)

- Un wallet par client, avec :
 - Solde courant (en MRU).
 - Historique des opérations (crédit, débit, ajustement).
- Types d'opérations :
 - Facturation (livraison, échange, caution, etc.).
 - Paiement reçu et validé.
 - Ajustement manuel (avec justification obligatoire).
- Gestion des paiements par reçus :
 - Formulaire de dépôt : montant, mode de paiement, date, upload de pièce jointe (jpg, png, pdf).
 - Workflow de validation : en attente → validé ou rejeté.
 - Historique des reçus et statut associé.

6.3. Gestion des bouteilles et des dépôts

- Catalogue des produits :
 - Liste des types de bouteilles (ex : 6 kg, 12 kg, 39 kg...).
 - Prix de vente / d'échange (en MRU).
 - Montant de caution/dépôt si applicable.
- Suivi par client :
 - Nombre de bouteilles par type : en dépôt, rendues, manquantes.
- Suivi par dépôt central et par bus :
 - Stock initial, mouvements (sorties vers bus, retours de vides, pertes éventuelles), stock théorique courant.

6.4. Tournées de bus et chauffeurs

- Gestion des bus :
 - Identification du bus (nom/numéro, plaque, capacité).
 - Chauffeur associé, contact.
- Planification des tournées :
 - Par jour, par secteur/zone.
 - Assignment des clients à une tournée.
- Application chauffeur (mobile Flutter) :
 - Authentification (identifiant + mot de passe ou téléphone + code).
 - Vue liste des tournées du jour.
 - Détails d'une tournée : liste des clients avec adresse, distance approximative, état (à visiter / visité / non joint).
 - Actions : marquer une visite, enregistrer livraisons/reprises, saisir un paiement.

6.5. Suivi en temps réel des bus (cartographie)

Objectif : suivre la position des bus en temps réel sur une carte, de manière similaire à Uber.

- Chaque appareil chauffeur envoie régulièrement sa position GPS au serveur, avec :
 - Latitude, longitude.
 - Date/heure.
 - Statut (en tournée, en pause, en retour dépôt, hors ligne...).
- Interface web (back-office / logistique) :

- Carte affichant tous les bus :
 - Chaque bus représenté par une icône/marker avec son identifiant.
 - Actualisation automatique des positions (toutes les 15 à 60 secondes selon paramétrage).
- Au clic sur un bus :
 - Nom/numéro du bus.
 - Nom du chauffeur.
 - Tournée en cours (secteur, date).
 - Nombre de clients restants à servir.
- Filtres : par secteur, par tournée, par statut de bus.
- Option vue client (facultative) :
 - Afficher la position approximative du bus qui doit le desservir.
 - Donner une estimation simple du temps d'arrivée (par exemple “Le bus est proche”, “10–15 minutes”) sans précision excessive.

6.6. Historique des positions et itinéraires (type Uber)

Objectif : pouvoir rejouer ou analyser l'itinéraire d'un bus sur une période.

- Pour chaque bus et chaque journée de tournée :
 - Enregistrement des points GPS successifs avec timestamp.
 - Construction d'un itinéraire (trace) sur la carte.
- Interface d'historique :
 - Sélection du bus + date (ou plage de dates).
 - Affichage du trajet complet sur la carte sous forme de polyline.
 - Option “replay” :
 - Animation du déplacement du bus le long de la trajectoire, avec curseur de temps.
- Statistiques associées :
 - Distance totale approximative parcourue.
 - Temps total en tournée.
 - Nombre de clients servis / non servis.
 - Points/segments où le bus est resté longtemps (embouteillages, arrêts prolongés).
- Contraintes techniques spécifiques :
 - Fréquence d'envoi des positions ajustable (ex. 20–30 s en mouvement, moins à l'arrêt).
 - Filtrage des points redondants (bus à l'arrêt) pour réduire la taille des données.
 - Mode hors ligne : stockage temporaire local des points et envoi différé dès que la connexion revient.

6.7. Administration & reporting

- Tableau de bord global :
 - Nombre de livraisons du jour/semaine/mois.
 - Volume de bouteilles distribuées par type.
 - Montant total des paiements validés (MRU).
 - Nombre de clients en retard de paiement (solde négatif, dettes dépassant un seuil).
- Rapports détaillés :
 - Par client (historique complet des livraisons, paiements, incidents).
 - Par bus/tournée (clients visités, temps de tournée, recettes encaissées, kilomètres parcourus).
 - Par type de bouteille.
- Gestion des utilisateurs :
 - Création/suppression de comptes utilisateurs.
 - Attribution de rôles (admin, opérateur, chauffeur, etc.).
 - Gestion des mots de passe.

6.8. Notifications & communication

- Types de notifications :
 - Confirmation de création de compte.
 - Confirmation de demande de livraison.
 - Validation/rejet d'un paiement.
 - Rappel en cas de dette ou de solde insuffisant.
 - Canaux possibles (à préciser selon les intégrations) :
 - SMS.
 - WhatsApp (via API Business si disponible).
 - Notifications push sur l'application mobile.
 - E-mails (optionnel).
-

7. Exigences non fonctionnelles

- **Performance** :
 - Temps de réponse acceptable même avec plusieurs centaines de clients, bus et transactions.
 - Optimisation des requêtes pour les cartes et historiques GPS.
 - **Disponibilité** :
 - Service accessible au minimum pendant les heures d'activité commerciales.
 - Possibilité d'extension vers haute disponibilité à terme.
 - **Sécurité** :
 - Authentification sécurisée.
 - Gestion des rôles et des permissions.
 - Chiffrement des mots de passe.
 - Accès contrôlé aux données sensibles (comptes clients, photos de reçus, positions GPS).
 - **Sauvegardes et reprise** :
 - Sauvegardes régulières de la base de données.
 - Procédure de restauration en cas de problème.
 - **Multilingue** :
 - Interface disponible en français et en arabe.
 - Choix de la langue par utilisateur et conservation du choix.
 - **Monnaie** :
 - Toutes les transactions sont gérées en MRU.
 - Affichage cohérent (format monétaire adapté).
-

8. UX adaptée aux personnes âgées ou peu familières avec la technologie

- **Interfaces simplifiées** :
 - Peu d'options par écran.
 - Parcours guidés en étapes (par ex. : 1) Choisir le service, 2) Confirmer l'adresse, 3) Valider la demande).
- **Lisibilité** :
 - Taille de police suffisamment grande.
 - Contraste élevé entre texte et fond.
- **Ergonomie mobile** :
 - Boutons larges, bien espacés.
 - Icônes explicites (bouteille de gaz, camion, portefeuille...).
 - Action claire sur l'écran principal : "Commander", "Payer", "Voir mon compte".
- **Accessibilité** :
 - Possibilité (à terme) d'augmenter la taille du texte.

- Messages simples, sans jargon, en français facile et en arabe.
 - **Support :**
 - Bouton pour appeler directement le service client depuis l'application.
-

9. Architecture technique proposée

- **Backend :**
 - Django avec Django REST Framework pour exposer des APIs sécurisées.
 - Django Admin pour un back-office rapide (paramétrage, gestion basique des données).
 - **Base de données :**
 - PostgreSQL (ou équivalent robuste) pour gérer les données clients, transactions, positions GPS, etc.
 - **Frontend web :**
 - Utilisation de l'admin Django + éventuellement un front dédié pour les opérateurs.
 - **Application mobile :**
 - Flutter pour Android (prioritaire en Mauritanie) et possibilité d'iOS.
 - **Cartographie :**
 - Intégration d'un service de carte (Leaflet + tuiles OpenStreetMap, Mapbox, Google Maps selon budget et contraintes de licence).
 - **Stockage des fichiers :**
 - Stockage des reçus (images/pdf) sur le serveur ou un stockage objet (type S3-compatible), avec accès sécurisé.
 - **Intégrations futures :**
 - APIs des opérateurs de mobile money locaux.
 - WhatsApp Business API pour notifications.
-

10. Évolutions possibles

- Application client plus complète :
 - Commande de gaz en quelques clics.
 - Suivi en temps réel du bus sur carte.
 - Gestion des réclamations / support client.
- Optimisation automatique des tournées :
 - Algorithmes de planification d'itinéraires pour réduire les distances et le temps.
- Programme de fidélité :
 - Points, remises, offres promotionnelles.
- Facturation avancée :
 - Génération de factures PDF et envoi par e-mail / WhatsApp.