

---

# Cahier des Charges - Phase de Développement du MVP "Djobea AI"

## 1. Introduction

Ce cahier des charges décrit les fonctionnalités et les exigences techniques du **Produit Minimum Viable (MVP)** de "Djobea AI", un agent conversationnel basé sur l'IA, destiné à faciliter la commande et la gestion de services à la demande au Cameroun, via WhatsApp. L'objectif est de valider le concept, la viabilité technique et l'acceptation par le marché dans une niche et une zone géographique spécifiques avant une expansion future.

## 2. Objectifs du MVP

- **Valider la proposition de valeur** : Démontrer que l'agent IA peut efficacement comprendre les demandes des utilisateurs et connecter aux prestataires.
- **Tester l'acceptation utilisateur** : Évaluer la facilité d'utilisation de l'interface conversationnelle et la satisfaction des clients.
- **Établir un processus de matching fonctionnel** : Assurer que le système peut identifier et notifier les prestataires adéquats.
- **Obtenir les premiers retours d'expérience** : Collecter des données pour l'amélioration continue et l'itération future.
- **Maîtriser les défis techniques initiaux** : Notamment la compréhension du langage local et l'intégration WhatsApp.
- **Démontrer la faisabilité de la monétisation** : Confirmer la viabilité du modèle économique de commission.

## 3. Périmètre du MVP

### 3.1. Services Ciblés (Niche Initiale)

- **Plomberie** : Réparations de fuites, débouchage, installation simple.
- **Électricité** : Réparation de pannes, installation d'ampoules/prises, réparation de ventilateurs/appareils simples.
- **Réparation d'Électroménager (Petit)** : Ex. Réparation de fer à repasser, bouilloire, mixeur.

### 3.2. Zone Géographique Ciblée

- **Douala** : Quartier de Bonamoussadi.

### 3.3. Utilisateurs Ciblés

- Particuliers résidant dans la zone cible, ayant un smartphone et utilisant WhatsApp.

## 4. Fonctionnalités Détaillées du MVP

#### 4.1. Module d'Interaction Utilisateur (via WhatsApp Business API)

- **Envoi de requêtes** : L'utilisateur peut envoyer un message texte décrivant son besoin.
- **Compréhension de la demande** : L'agent doit :
  - Identifier le **service requis** (ex: plomberie, électricité, réparation ventilateur).
  - Extraire la **localisation exacte** (adresse dans la zone cible).
  - Extraire la **description du problème** et les détails pertinents.
  - Extraire le **délai souhaité** (ex: "cet après-midi", "demain matin", "urgent").
  - Demander des informations manquantes de manière conversationnelle et polie (ex: "Quel service avez-vous besoin ?", "Pouvez-vous me donner votre adresse exacte ?").
- **Confirmation de la demande** : Après avoir collecté toutes les informations, l'agent doit récapituler la demande à l'utilisateur pour validation.
- **Statut de la demande** : L'utilisateur peut demander le statut de sa demande et recevoir une mise à jour.
- **Annulation de la demande** : L'utilisateur peut annuler sa demande tant qu'elle n'est pas encore acceptée par un prestataire.

#### 4.2. Module de Gestion des Prestataires

- **Enregistrement des prestataires (manuel pour le MVP)** :
  - Les prestataires seront ajoutés manuellement à la base de données par l'équipe administrative après un processus de vérification rigoureux (hors périmètre MVP).
  - **Informations à stocker** : Nom, contact WhatsApp, services offerts, zones couvertes, disponibilité initiale (simple oui/non ou plage horaire), note moyenne (initialement vide).
- **Notification des prestataires** :
  - Lorsqu'une demande d'utilisateur est reçue et validée, l'agent identifie le(s) prestataire(s) qualifié(s) et disponible(s).
  - Envoi d'un message WhatsApp au(x) prestataire(s) concerné(s) avec les détails de la mission (service, localisation, description, délai).
  - Le prestataire doit pouvoir **accepter ou refuser** la mission via une réponse simple par WhatsApp (ex: "OUI" pour accepter, "NON" pour refuser).
- **Gestion de la réponse du prestataire** :
  - Si accepté : L'agent confirme à l'utilisateur que le prestataire est en route/disponible.
  - Si refusé ou pas de réponse après un délai X : L'agent tente de contacter le prestataire suivant ou informe l'utilisateur d'un délai.
- **Communication directe (après acceptation)** : Une fois la mission acceptée, l'agent fournit les contacts directs (numéro de téléphone) de l'utilisateur au prestataire et vice-versa pour faciliter la communication sur place.

#### 4.3. Module de Base de Données

- **Stockage des Utilisateurs** : ID WhatsApp, nom (optionnel), historique des demandes.

- **Stockage des Prestataires** : ID, nom, contact WhatsApp, services, zones, disponibilité, statut, évaluations.
- **Stockage des Demandes/Missions** : ID unique, ID utilisateur, ID prestataire (si assigné), service demandé, description, localisation, délai souhaité, statut de la mission (en attente, assignée, en cours, terminée, annulée), date de création, date de fin.
- **Stockage des Conversations** : Pour le débogage et l'amélioration de la compréhension du LLM.

#### 4.4. Module d'Administration (Accès Restreint)

- **Tableau de bord simple** : Pour visualiser les demandes en cours, les prestataires disponibles, et les problèmes.
- **Gestion manuelle des prestataires** : Ajouter, modifier, désactiver des prestataires.
- **Suivi des conversations** : Accès aux logs des interactions pour analyse et amélioration.

## 5. Exigences Techniques

### 5.1. Technologies et Langages

- **Backend & Logique Agent** : Python 3.9+
- **Framework Web** : FastAPI (recommandé pour sa performance et sa simplicité avec les APIs)
- **Orchestration LLM** : LangChain
- **Modèle de Langage** : API Google Gemini ou OpenAI GPT (dépendra de la disponibilité et du coût initial).
- **Base de Données** : PostgreSQL
- **Intégration WhatsApp** : WhatsApp Business API via un fournisseur agréé (ex: Twilio, MessageBird, ou un partenaire camerounais si plus simple).
- **Déploiement** : Conteneurisation Docker (pour le backend et la DB), déploiement sur un VPS ou un service cloud IaaS (ex: AWS EC2, Google Cloud Compute Engine) au Cameroun ou en Europe (pour des raisons de latence et de coût).

### 5.2. Sécurité

- Protection des données utilisateurs et prestataires (cryptage des données sensibles au repos et en transit).
- Sécurisation des clés API (LLM, WhatsApp).
- Gestion des accès à l'administration.

### 5.3. Performance et Scalabilité (MVP)

- La solution doit être capable de gérer au moins 50 requêtes simultanées sans dégradation majeure.
- Temps de réponse de l'agent raisonnable (moins de 5 secondes pour une réponse initiale).

### 5.4. Robustesse et Gestion des Erreurs

- Gestion des erreurs de connexion (API, DB).
- Log des erreurs pour le débogage.
- Mécanisme de "fallback" si un prestataire ne répond pas.

## 6. Stratégie de Monétisation du MVP

Le modèle économique initial de "Djobea AI" sera basé sur la **commission sur les transactions**, car il est le plus simple à implémenter pour un MVP et aligne les intérêts de toutes les parties.

- **Commission** : Un pourcentage (à définir, par exemple **10% à 20%**) sera prélevé sur le montant total de chaque service complété via la plateforme.
- **Processus de Paiement (Post-MVP)** : Bien que les paiements initiaux puissent être gérés directement entre l'utilisateur et le prestataire (avec commission payée manuellement par le prestataire), l'objectif à terme sera d'intégrer les solutions de Mobile Money (Orange Money, MTN Mobile Money) pour automatiser la collecte du paiement et le versement net au prestataire.
- **Avantages pour le MVP** :
  - **Faible barrière à l'entrée** pour les prestataires (pas de frais initiaux).
  - **Alignement des intérêts** : "Djobea AI" gagne quand les prestataires réalisent des missions.
  - **Simplicité de compréhension** pour les utilisateurs et prestataires.

## 7. Phases de Développement (Itérations du MVP)

1. **Sprint 1 : Base Technique et Connexion WhatsApp**
  - Mise en place de l'environnement de développement.
  - Configuration de la base de données PostgreSQL (tables utilisateurs, prestataires, demandes).
  - Configuration et test de l'intégration WhatsApp Business API (envoi/réception de messages simples).
  - Création d'un serveur FastAPI/Flask basique pour recevoir les webhooks WhatsApp.
2. **Sprint 2 : Compréhension et Extraction (Cerveau de l'Agent)**
  - Intégration du LLM (Gemini/GPT) via LangChain.
  - Développement de la logique de "prompt engineering" pour extraire service, localisation, description et délai.
  - Test unitaire de la compréhension des demandes types via WhatsApp.
3. **Sprint 3 : Matching et Notification Prestataire**
  - Développement de la logique de recherche et de sélection de prestataire dans la DB.
  - Implémentation de l'envoi de notifications WhatsApp aux prestataires avec les détails de la mission.
  - Mise en place de la réception de la réponse "OUI/NON" du prestataire.
4. **Sprint 4 : Retour Utilisateur et Cycle de Vie de la Demande**
  - Implémentation de la confirmation utilisateur après acceptation prestataire.
  - Développement du suivi du statut de la mission.
  - Implémentation de la fonction d'annulation.

- Création d'une interface d'administration simple pour visualiser les demandes.

## 8. Critères de Succès du MVP

- **Taux de compréhension** : L'agent comprend correctement au moins 80% des demandes initiales des utilisateurs (pour les services ciblés).
- **Taux de matching** : Au moins 70% des demandes sont connectées avec un prestataire dans le délai imparti.
- **Satisfaction utilisateur** : Un score minimum de 4/5 pour les utilisateurs ayant interagi avec l'agent (via un sondage simple post-service, si implémenté, ou via la qualité du feedback).
- **Engagement prestataire** : Au moins 50% des prestataires contactés répondent aux demandes.
- **Volume** : Capacité à gérer 5-10 demandes actives simultanément.
- **Viabilité du modèle économique** : Validation de la volonté des prestataires de payer la commission après service.

## 9. Phase de Tests

- **Tests Unitaires** : Pour chaque module (compréhension, matching, notification).
- **Tests d'Intégration** : Vérifier que les modules interagissent correctement.
- **Tests de bout en bout (End-to-End)** : Simuler un parcours utilisateur complet depuis la requête initiale jusqu'à la notification du prestataire et la confirmation.
- **Tests utilisateurs** : Impliquer un petit groupe d'utilisateurs bêta réels pour tester le service dans des conditions réelles.