

# Cahier des Charges – Plateforme de Mise en Relation Prestataires/Clients GR 12

15 décembre 2024

## 1. Présentation du Projet

On souhaite de développer une plateforme Web permettant de mettre en relation des prestataires de services indépendants (en ligne et hors ligne) avec des clients potentiels. Les services concernés incluent, par exemple :

- Prestataires « online » : Développeurs, Designers, Rédacteurs, Traducteurs, Community managers, etc.
- Prestataires « offline » : Plombiers, Électriciens, Menuisiers, Techniciens de climatisation, etc.

L'objectif principal est d'offrir un espace unique où le client peut :

1. Rechercher un prestataire selon sa spécialité, de façon guidée ou par filtrage.
2. Consulter les offres de services (tarifs, délais, compétences).
3. Communiquer avec les prestataires et évaluer leurs services.
4. Poster une demande de service spécifique et recevoir des candidatures.
5. Gérer et Communiquer avec les candidats.
6. Gestion des réclamations (Clients, Prestataires ).

## 2. Objectifs et Périmètre du Projet

### Objectifs fonctionnels

- Permettre aux prestataires de créer et gérer leur profil et leurs offres de services.
- Permettre aux clients de consulter les services, contacter les prestataires, poster une offre personnalisée, et gérer leurs commandes.
- Intégrer un LLM (GroqCloud) pour analyser automatiquement la demande initiale du client (problématique décrite en texte libre) et en extraire des paramètres structurés (dans un schéma JSON défini).
- Utiliser ce schéma JSON pour filtrer automatiquement les prestataires dans la base de données (catégorie, ville, expérience, disponibilité, etc.), réduisant ainsi le temps de recherche.
- Mettre en place un système d'évaluation et de commentaires.
- Assurer une messagerie interne (notifications, échanges directs).
- Offrir un tableau de bord d'administration pour gérer les comptes utilisateurs, modérer les contenus, et valider ou supprimer des services.

## Objectifs techniques

- Développer une architecture évolutive et sécurisée.
- Intégrer des technologies back-end (Spring Boot, PostgreSQL) et front-end (React).
- Déployer la plateforme dans le cloud (AWS EC2) avec un pipeline d'intégration continue (Jenkins, Docker, SonarQube).
- Utiliser GroqCloud (LLMs) pour le prétraitement de la demande du client.
- Prévoir une évolutivité ultérieure pour intégrer un module de paiement.

## Hors Périmètre (Phase initiale)

- Gestion des paiements en ligne.
- Stratégies marketing approfondies.
- Localisation multilingue (optionnel, peut être envisagée ultérieurement).

## 3. Fonctionnalités Détails

### Espace Prestataire

- Créer un compte (profil, coordonnées, champs de spécialisation).
- Ajouter, modifier, supprimer des services (description, tarifs, délais, images).
- Consulter et répondre aux offres publiées par les clients.
- Gérer les commandes (statut, messagerie, historiques).
- Consulter les évaluations et commentaires, y répondre.

### Espace Client

- Créer un compte (informations personnelles, préférences).
- Rédiger une demande libre de service sous forme de problématique (ex : "J'ai besoin d'un électricien à Casablanca pour réparer une prise dans ma salle de bain").
- Le LLM (GroqCloud) analyse la demande et produit un schéma JSON standardisé (par ex. { "serviceType": "électricité", "ville": "Casablanca", "délai": "urgent", ...}).
- La plateforme utilise ce schéma pour filtrer automatiquement les prestataires correspondants.
- Commander un service ou publier une demande personnalisée.
- Gérer les commandes, communiquer avec les prestataires, suivre l'avancement.
- Évaluer les prestataires et laisser des commentaires.

### Espace Administrateur

- Gérer les comptes (clients, prestataires), valider ou suspendre les profils.
- Gérer le catalogue de services (valider, supprimer, modérer les contenus).
- Traiter les réclamations et signalements.
- Accéder à des statistiques globales (nombre d'utilisateurs, services, activités).

### Messagerie Interne & Notifications

- Système de messagerie pour échanger entre client et prestataire.
- Notifications pour les nouvelles commandes, offres, messages, évaluations.

## 4. Exigences Techniques et de Qualité

### Back-end

Spring Boot (Java), base de données PostgreSQL.

### Front-end

React, design responsive, support multi-navigateurs.

### Intégration & Déploiement

Docker (containers), Jenkins (CI/CD), SonarQube (qualimétrie), AWS EC2 (hébergement).

### Intégration du LLM (GroqCloud)

- Appeler l'API du LLM avec la description du besoin du client.
- Récupérer la réponse structurée (JSON) définie par un schéma prédéfini.
- Utiliser ce JSON comme source de filtres pour interroger la base de données.

### Sécurité

- Authentification par email et mot de passe, récupération de mot de passe.
- Protection contre les injections SQL et XSS.
- Gestion des rôles (admin, prestataire, client) et droits d'accès.

## 5. Gestion de Projet et Suivi

- Méthodologie Agile (Scrum), sprints courts.
- Outils de gestion de projet : Jira, Trello ou GitLab/GitHub Projects.
- Tests unitaires, intégration, end-to-end (JUnit, Playwright, Apache JMeter).
- SonarQube pour indicateurs de qualité.

## 6. Évolutions Futures

- Intégration d'une passerelle de paiement (Stripe, PayPal).
- Version multilingue (Français, Anglais, Arabe).
- Fonctionnalités marketing (newsletter, offres promotionnelles).
- Application mobile dédiée (React Native).