

# Spécifications pour une Application de Livraison (Inspiration UberEats)

## 1. Objectif du Projet

Créer une plateforme de livraison novatrice, mettant l'accent sur la transparence, l'équité, et une expérience utilisateur optimisée, tout en comblant les lacunes identifiées dans des solutions comme UberEats.

## 2. Acteurs de l'Application

### a. Clients

- Commandent des repas ou des produits.
- Suivent leurs commandes en temps réel.
- Partagent des avis et des notes sur les livraisons.

### b. Restaurants/Commerçants

- Configurent et gèrent leurs menus.
- Traitent les commandes et gèrent leur disponibilité.
- Accèdent à des rapports d'analyse (ventes, tendances).

### c. Livreurs

- Acceptent et complètent des courses via l'application.
- Utilisent le GPS intégré pour optimiser leurs trajets.
- Consultent leurs revenus détaillés par livraison.

#### **d. Administrateurs**

- Gèrent les utilisateurs, les commandes et les performances de la plateforme.
- Supervisent les interactions entre clients, restaurants et livreurs.
- Accèdent aux statistiques globales pour ajuster les stratégies.

### **3. Besoins Fonctionnels**

#### **a. Côté Clients**

1. Recherche par catégorie (restaurant, type de cuisine, popularité).
2. Options de paiement diversifiées et sécurisées.
3. Suivi des commandes en temps réel avec estimations de livraison.
4. Notifications push pour les mises à jour de commandes.
5. Système de fidélisation pour récompenser les clients réguliers.

#### **b. Côté Restaurants/Commerçants**

1. Gestion des menus (ajout, modification, suppression).
2. Suivi des commandes en cours avec mises à jour en temps réel.
3. Accès aux rapports de ventes (meilleurs plats, périodes de pointe).

#### **c. Côté Livreurs**

1. Acceptation ou refus des commandes en fonction de leur disponibilité.
2. Suivi GPS avec itinéraires optimisés.
3. Historique des courses avec rémunération détaillée.

4. Mode « indisponible » pour suspendre les notifications.

#### **d. Côté Administrateurs**

1. Tableau de bord centralisé pour superviser l'ensemble des activités.
2. Gestion des promotions et des offres spéciales.
3. Gestion des litiges (remboursements, notes injustes).
4. Contrôle des activités suspectes ou frauduleuses.

#### **4. Besoins Non Fonctionnels**

##### **1. Performance:**

- Temps de réponse rapide pour les actions critiques (recherche, commande).
- Optimisation pour les pics d'utilisation (heures de déjeuner/dîner).

##### **2. Sécurité :**

- Chiffrement des données sensibles (paiements, données personnelles).
- Authentification forte (2FA pour les administrateurs).

##### **3. Scalabilité :**

- Architecture modulaire pour étendre les services (épiceries, pharmacies).
- Support multi-langue et multi-devise.

##### **4. Fiabilité :**

- Temps de disponibilité supérieur à 99,9%.
- Tolérance aux pannes pour éviter les interruptions majeures.

## 5. Accessibilité :

- Compatibilité avec les normes WCAG pour les personnes en situation de handicap.
- Interface intuitive et adaptée aux écrans mobiles.

## 5. Points Différenciants (Pour se démarquer d'UberEats)

### a. Transparence sur les Coûts

- Explication claire des frais (livraison, service, taxes).
- Réduction des commissions imposées aux restaurants.

### b. Équité Sociale

- Rétribution équitable pour les livreurs, avec un revenu minimum garanti.
- Mise en avant des commerces locaux et indépendants.

### c. Personnalisation Avancée

- Suggestions basées sur les préférences des utilisateurs.
- Filtrage selon les régimes alimentaires (végétarien, sans gluten, etc.).

### d. Durabilité Écologique

- Promotion de la livraison en vélo ou en véhicules électriques.
- Système de compensation carbone pour chaque commande.

### e. Interaction Sociale

- Mode commande partagée (groupes d'amis ou collègues).

- Programmes de fidélisation collaboratifs (réduction pour commandes groupées).

#### f. Résolution Rapide des Problèmes

- Service client accessible 24/7 via chat intégré.
- Suivi des litiges et historique des résolutions.

### 6. Technologies Recommandées

#### a. Frontend

- React.js ou Vue.js pour des interfaces dynamiques.
- TailwindCSS ou Bootstrap pour un design moderne et responsive.

#### b. Backend

- Laravel pour sa robustesse et sa flexibilité.
- Node.js pour une gestion asynchrone des événements (ex : notifications en temps réel).

#### c. Base de Données

- PostgreSQL ou MySQL pour la gestion des données relationnelles.
- Redis pour le caching et les sessions.

#### d. Services Externes

- Stripe ou PayPal pour les paiements sécurisés.
- Google Maps API pour le suivi GPS.
- Firebase pour les notifications push.

## **7. Design Anticipé**

### **a. Interface Client**

- Page d'accueil simple avec barre de recherche avancée.
- Suggestions basées sur l'historique des commandes.
- Affichage clair des promotions en cours.

### **b. Interface Restaurant**

- Tableau de bord centralisé avec commandes et statistiques.
- Gestion rapide des disponibilités et des stocks.

### **c. Interface Livreurs**

- Carte interactive avec itinéraire en temps réel.
- Affichage des gains par livraison et total hebdomadaire.
- Bouton d'activation/désactivation de disponibilité.

## **8. Plan d'Évolution**

1. Phase 1 : Développement de la livraison de repas.
2. Phase 2 : Intégration des commerces locaux (épiceries, pharmacies).
3. Phase 3 : Livraison collaborative entre utilisateurs (peer-to-peer).

## **9. Prochaines Étapes**

1. Rédiger un cahier des charges détaillé.
2. Concevoir des maquettes fonctionnelles pour chaque rôle.

3. Prototyper un MVP (Produit Minimum Viable).
4. Lancer une phase de tests bêta avec des utilisateurs ciblés.
5. Ajuster le produit final en fonction des retours.