

Projet Livraison de Repas

Introduction

Le système de livraison de repas est une plateforme en ligne permettant aux clients de commander des repas, de suivre l'état de leurs commandes et de recevoir leurs repas à domicile. Ce système est conçu pour trois types d'utilisateurs : l'Admin, qui gère les restaurants et les utilisateurs ; le Client, qui passe des commandes et suit leur livraison ; et le Livreur, qui accepte et livre les commandes aux clients.

Ce projet vise à simplifier l'ensemble du processus de commande et de livraison de repas en offrant une interface intuitive pour tous les utilisateurs, tout en garantissant une gestion efficace des commandes et une traçabilité des livraisons en temps réel.

Rôles des utilisateurs

1. Admin

L'Admin a un contrôle total sur le système. Il peut :

- Ajouter, modifier et supprimer des restaurants.
- Gérer les menus des restaurants.
- Visualiser toutes les commandes et leur statut.
- Ajouter, modifier ou supprimer des utilisateurs.

2. Client

Le Client peut :

- Parcourir les menus des restaurants disponibles.
- Ajouter des produits au panier et passer une commande.
- Suivre l'état de la commande et de la livraison.
- Effectuer un paiement en ligne pour les commandes.

3. Livreur

Le Livreur peut :

- Voir les commandes qui lui sont attribuées.
- Accepter ou refuser des commandes.
- Marquer une commande comme livrée ou en cours de livraison.
- Modifier son statut de disponibilité.

Fonctionnalités

- **Gestion des restaurants** : Les administrateurs peuvent ajouter, supprimer ou modifier les restaurants et leurs menus.

- **Commandes** : Les clients peuvent choisir des repas, passer des commandes et effectuer des paiements en ligne.
- **Suivi des livraisons** : Les livreurs peuvent accepter des commandes et marquer leur statut en temps réel.
- **Authentification et sécurité** : Le système inclut une connexion sécurisée avec un mot de passe crypté pour chaque utilisateur.
- **Notifications en temps réel** : Les clients et les livreurs reçoivent des notifications lorsqu'une commande est acceptée, expédiée ou livrée.

Architecture du système

Le système repose sur une architecture client-serveur, où une base de données MySQL est utilisée pour stocker les informations des utilisateurs, des restaurants, des commandes et des transactions. L'interface utilisateur est construite avec HTML, CSS et JavaScript, tandis que le back-end est géré avec Node.js et Express.

Schéma de l'architecture

 Architecture Diagram

 Sequence Diagram

 Use case Diagram

Priorités des Cas d'Utilisation

Pour le premier sprint, les cas d'utilisation suivants sont considérés comme prioritaires :

1. **Le client passe une commande**
2. **L'admin gère les restaurants**

Spécifications

1. Le client passe une commande

- **Précondition:** Le client doit être connecté et avoir des articles dans le panier.
- **Postcondition:** La commande est enregistrée dans la base de données, et le client reçoit une confirmation de commande.
- **Flux d'événements:**
 1. Le client ajoute des articles au panier.
 2. Le client procède à la caisse.
 3. Le système valide le paiement.
 4. Le système confirme la commande et met à jour la base de données.

2. L'admin gère les restaurants

- **Précondition:** L'admin est connecté à l'interface admin.
- **Postcondition:** La liste des restaurants est mise à jour.
- **Flux d'événements:**

1. L'admin se connecte à l'interface.
2. L'admin sélectionne l'option de gestion des restaurants.
3. L'admin ajoute, modifie ou supprime des informations sur les restaurants.

Cas de Test pour "Le client passe une commande"

1. Test valide:

- **Préconditions:** Client connecté, articles dans le panier.
- **Action:** Client passe la commande.
- **Résultat attendu:** La commande est enregistrée, et le client reçoit une confirmation.

2. Test paiement invalide:

- **Préconditions:** Client connecté, articles dans le panier.
- **Action:** Client tente de passer une commande avec un paiement échoué.
- **Résultat attendu:** La commande n'est pas enregistrée, et le client est informé de l'échec du paiement.