Liv'In Paris - Application de partage de repas entre voisins

Description du projet

Liv'In Paris est une application qui permet aux habitants de Paris de partager des repas entre voisins. Les utilisateurs peuvent être clients ou cuisiniers (ou les deux). L'application permet de gérer les utilisateurs, les plats, les commandes et optimise les trajets de livraison via le métro parisien.

Architecture du projet

Classes principales

- **Application** : Point d'entrée du programme, gère le menu principal
- **GestionAdmin** : Gère les fonctionnalités administratives (clients, cuisiniers, commandes)
- **GrapheMetro** : Implémente les algorithmes de graphes pour le métro parisien
- **ConnexionBDD** : Gère la connexion à la base de données MySQL
- **GestionImportExport** : Gère l'import/export des données en JSON

Base de données

La base de données contient les tables suivantes :

- UTILISATEUR : informations des utilisateurs
- PLAT : liste des plats disponibles
- COMMANDE : historique des commandes
- STATION : stations du métro parisien
- LIGNE : lignes du métro parisien

Fonctionnalités implémentées

1. Gestion des utilisateurs

- Ajout/modification/suppression de clients et cuisiniers
- Affichage des utilisateurs par différents critères

- Gestion des plats proposés par les cuisiniers

2. Gestion des commandes

- Création et modification de commandes
- Calcul des prix
- Simulation des étapes de commande
- Optimisation des trajets de livraison

3. Algorithmes de graphes

- Dijkstra : plus court chemin
- Bellman-Ford : plus court chemin avec poids négatifs
- Floyd-Warshall : plus courts chemins entre toutes les paires
- Chargement du graphe du métro parisien

4. Import/Export

- Export des données en JSON
- Import des données depuis JSON
- Sauvegarde et restauration de l'état de l'application

5. Statistiques

- Nombre de livraisons par cuisinier
- Commandes par période
- Moyennes des prix
- Statistiques par type de plat

6. Fonctionnalités créatives

Pour les administrateurs

- **Système de notation des cuisiniers** : Les administrateurs peuvent noter les cuisiniers sur différents critères (qualité, ponctualité, etc.)
- **Générateur de rapports hebdomadaires** : Création automatique de rapports sur l'activité de la semaine
- **Système d'alertes** : Notifications pour les commandes en retard ou les problèmes de livraison

Pour les cuisiniers

- **Menu du jour intelligent** : Suggestion de plats en fonction des ingrédients disponibles et de la saison
- **Carnet de recettes** : Stockage et gestion des recettes personnelles
- **Statistiques personnelles** : Visualisation des plats les plus populaires et des retours clients

Pour les clients

- **Système de favoris** : Sauvegarde des plats et cuisiniers préférés
- **Historique des commandes** : Visualisation des commandes passées avec possibilité de les recommander
- **Système de recommandations** : Suggestions de plats basées sur les commandes précédentes

Avancement du projet

Étape 1 (Terminée)

- Implémentation du graphe simple
- Algorithmes BFS, DFS
- Création de la base de données
- Visualisation basique du graphe

Étape 2 (Terminée)

- Graphe générique
- Application au métro parisien
- Implémentation de Dijkstra, Bellman-Ford, Floyd-Warshall
- Intégration SQL
- Visualisation des chemins

Étape 3 (En cours)

- Coloration de graphe (Welsh-Powell)
- Export XML
- Fonctionnalités créatives
- Interface WPF

Tests unitaires

Les tests unitaires couvrent :

- Connexion à la base de données
- Algorithmes de graphes
- Gestion administrative
- Import/Export des données

Prochaines étapes

1. Finaliser l'interface WPF
2. Implémenter la coloration de graphe
3. Ajouter des fonctionnalités créatives
4. Optimiser les performances
5. Améliorer la gestion des erreurs

Notes techniques

- Utilisation de MySQL pour la base de données
- Format JSON pour l'import/export
- Algorithmes de graphes optimisés
- Architecture modulaire et extensible