

Cahier des Charges pour le Projet d'Optimisation de l'Itinéraire de Livraison

1. Introduction

1.1 Contexte

L'optimisation des itinéraires de livraison est cruciale pour les entreprises de logistique afin de minimiser les coûts et améliorer l'efficacité. Ce projet vise à développer un algorithme qui optimise les itinéraires de livraison en minimisant le temps et le coût.

1.2 Objectifs

- Développer un algorithme d'optimisation d'itinéraire.
- Implémenter une interface utilisateur pour visualiser les itinéraires.
- Stocker et gérer les données de livraison dans une base de données.
- Comparer et analyser les performances de différents algorithmes d'optimisation.

2. Cahier des Charges Fonctionnel

2.1 Fonctionnalités Principales

- Saisie des données** : Interface pour entrer les points de départ, les destinations, les distances, et les coûts.
- Optimisation d'itinéraire** : Implémentation d'algorithmes comme Dijkstra, algorithmes génétiques, et autres techniques pertinentes.
- Visualisation des itinéraires** : Affichage des itinéraires optimisés sur une carte interactive.
- Rapports de performance** : Génération de rapports détaillés sur le temps et le coût des itinéraires optimisés.
- Base de données** : Stockage des données de livraison et des résultats d'optimisation.

2.2 Fonctionnalités Optionnelles

- Prévisions basées sur l'historique** : Utilisation des données historiques pour améliorer les prédictions d'itinéraire.
- Notifications en temps réel** : Alertes en cas de déviations ou de retards dans les itinéraires planifiés.

3. Cahier des Charges Technique

3.1 Technologies Utilisées

- Langage de programmation** : Python (pandas, numpy, scikit-learn)
- Algorithmes** : Algorithme de Dijkstra, algorithmes génétiques
- Base de données** : SQL (MySQL, PostgreSQL)
- Visualisation** : Folium, Matplotlib
- Interface utilisateur** : HTML, CSS, JavaScript (Frameworks : Flask/Django pour le backend)

3.2 Architecture du Système

1. Frontend :

- Formulaire de saisie des données de livraison.
- Visualisation des itinéraires optimisés.

2. Backend :

- Traitement des données et optimisation des itinéraires.
- API pour communiquer avec le frontend.

3. Base de données :

- Stockage des points de livraison, distances, coûts et résultats d'optimisation.

4. Déroulement du Projet

4.1 Phase de Conception

- **Analyse des besoins** : Compréhension des exigences et des contraintes du projet.
- **Spécifications fonctionnelles** : Documentation des fonctionnalités principales et optionnelles.
- **Design du système** : Schémas de la base de données, architecture du système.

4.2 Phase de Développement

- **Développement du backend** : Implémentation des algorithmes d'optimisation et API.
- **Développement du frontend** : Création de l'interface utilisateur et des visualisations.
- **Intégration de la base de données** : Configuration de la base de données et intégration avec le backend.

4.3 Phase de Test

- **Tests unitaires** : Validation des composants individuels du système.
- **Tests d'intégration** : Vérification de l'interaction entre les différents composants.
- **Tests de performance** : Évaluation de l'efficacité des algorithmes et du système global.

4.4 Phase de Déploiement

- **Mise en production** : Déploiement de l'application sur un serveur.
- **Documentation** : Création de la documentation utilisateur et technique.
- **Formation** : Formation des utilisateurs finaux (si nécessaire).

4.5 Phase de Maintenance

- **Support technique** : Assistance en cas de problèmes ou de bugs.
- **Mises à jour** : Améliorations et ajout de nouvelles fonctionnalités.

5. Contraintes et Risques

5.1 Contraintes

- **Budget** : Gestion des ressources financières pour le développement et le déploiement.
- **Technologie** : Limites techniques des technologies choisies.

5.2 Risques

- **Complexité algorithmique** : Difficulté à implémenter des algorithmes avancés.
 - **Fiabilité des données** : Données incomplètes ou incorrectes pouvant affecter les résultats.
 - **Sécurité** : Protéger les données sensibles des utilisateurs.
-

Annexe

6.1 Ressources

- Documentation des bibliothèques et frameworks utilisés.
- Tutoriels et cours en ligne pour approfondir les compétences nécessaires.

6.2 Références

- Littérature sur les algorithmes d'optimisation (Dijkstra, algorithmes génétiques).
- Articles et études de cas sur l'optimisation des itinéraires de livraison.