

Projet CleanThis - Cahier des charges

1. Introduction

La société Nettoyage **CleanThis** souhaite développer une application pour la gestion de ses opérations de nettoyage destinées aux professionnels et particuliers. L'application doit permettre la gestion des utilisateurs avec différents rôles, la création et le suivi des opérations, ainsi que la génération de rapports en PDF à envoi par e-mail. Des détaillent de gestion des statistiques de ventes via des graphiques, offrant une vision claire des performances de ventes par type d'opération et des analyses visuelles.

L'application doit être multilingue (internationalisation EN_en ; FR_fr)

2. Maquette :

- Une maquette détaillée de l'interface utilisateur sera conçue à l'aide de l'outil de conception Figma ou autre.
- La maquette inclura les différentes pages de l'application, les éléments d'interface, et les interactions prévues.
- Les utilisateurs et l'équipe de développement pourront ainsi avoir une vision claire de la structure et de l'apparence de l'application avant le début du développement.

3. Utilisateurs et Rôles

L'application comprendra trois types d'utilisateurs :

- Expert (Administrateur)
- Senior
- Apprenti

4. Fonctionnalités par Rôle

- **Expert (Administrateur) :**
 - Ajouter du personnel
 - Prendre jusqu'à 5 opérations à la fois
 - Créer les identifiants du personnel
 - Modifier le rôle du personnel
 - Consulter le chiffre d'affaires de l'entreprise
- **Senior :**
 - Prendre jusqu'à 3 opérations à la fois
 - Connexion par un Senior :
 - Ajouter une opération
 - Terminer une opération
 - Lister les opérations en cours
- **Apprenti :**
 - Ne peut prendre qu'une opération à la fois
 - Connexion par un Apprenti :
 - Ajouter une opération
 - Terminer une opération
 - Lister les opérations en cours

5. **Opérations et Clients**

- Chaque opération est gérée par un seul utilisateur (Expert, Senior, ou Apprenti).
- Un client peut demander plusieurs opérations.

6. **Affichage Initial**

- Afficher la liste des opérations en cours (assignées à chaque rôle, triées par ordre alphabétique).
- Afficher la liste des opérations terminées (triées par ordre alphabétique).
- Le personnel peut se connecter avec un identifiant et un mot de passe.
- Une fonctionnalité de trie par colonne et un champ de recherche sont nécessaires pour une recherche rapide.

7. **Menu Selon le Rôle**

- Afficher un menu spécifique selon le rôle de l'utilisateur connecté.

8. **Types d'Opérations**

- Grosse (5,000€)
- Moyenne (2,500€)
- Petite manœuvre (1,000€)
- Custom (xxx€)

9. **Informations Demandées lors de l'Ajout d'une Opération**

- Type de l'opération
- Nom, prénom et adresse du client
- Description de l'opération

10. **Intégration API Externe**

- Faire un appel API externe vers le site du gouvernement pour la gestion des adresses client en utilisant la technologie AJAX.
- [API Gouv - Base Adresse Nationale](#)

11. **Technologies Requises**

- MCD et MLD à fournir
- Diagramme de classe
- Diagramme des appels (Draw.io ou Framindmap.org)
- Modèle MVC doit être respecté
- Symfony requis
- Authentification via Google
- L'application doit générer des factures en fichier PDF et les envoyer par e-mail au client.
- Des appels Ajax ou Fetch sont requis pour appeler le backend.
- Javascript pour les charts graphiques.

12. **Livrables Attendus**

- MCD et MLD
- Diagramme de classe
- Diagramme des appels
- Application Symfony fonctionnelle avec authentification Google
- Intégration de l'appel API externe pour la gestion des adresses
- Génération de rapports en PDF et envoi par e-mail

- Documentation complète du code
- Tests unitaires et d'intégration

13. **Gérer les statistiques de ventes via des graphiques (histogrammes, camemberts).**

- Visualiser les performances de ventes par type d'opération.
- Analyser les tendances mensuelles ou annuelles.
- Accéder à des rapports visuels pour une meilleure compréhension.

Parties 2 :

1. **Gestion des Logs et Intégration de Services Tierces**

L'application inclura des fonctionnalités avancées de gestion des logs et utilisera des services tiers pour la reconnaissance d'image et la transcription audio. Ces fonctionnalités seront intégrées à la base de données NoSQL MongoDB via une API Node.js.

- **Logs dans une BDD NoSQL (MongoDB) :**
 - L'application enregistrera les logs dans une base de données NoSQL MongoDB pour assurer la traçabilité des opérations et des événements.
 - Les logs incluront des informations telles que les actions effectuées, les utilisateurs concernés, les horodatages, etc.
 - L'API Node.js sera responsable de l'alimentation de la base de données MongoDB avec les logs générés par l'application.
- **Traçabilité et Archivage :**
 - L'objectif principal de ces intégrations est de renforcer la traçabilité des actions effectuées dans l'application.
 - Les informations loggées dans la base de données MongoDB permettront de suivre les activités des utilisateurs de manière détaillée.
 - Les résultats de la reconnaissance d'image et de la transcription audio seront archivés pour permettre aux utilisateurs de consulter l'historique de leurs actions.
- **API Node.js :**
 - Une API Node.js servira de back-end
 - Elle gèrera les requêtes et les réponses entre l'application front-end, la base de données MongoDB, et les services tiers.
 - Responsable de l'intégration des logs dans la base de données MongoDB.
- **Utilisation de l'API Lens pour la Reconnaissance d'Image :**

- L'API Lens sera utilisée pour la reconnaissance d'image dans l'application.
- Les images fournies par les utilisateurs seront traitées via l'API Lens pour extraire des informations pertinentes, telles que des objets, des textes, etc.
- **Objectif de la Fonctionnalité Demande de Nettoyage via Photo :**
 - Permettre au client de soumettre une demande de nettoyage en envoyant une photo de l'objet à nettoyer.
 - Utiliser Google Lens pour la reconnaissance d'image afin d'identifier la catégorie de l'objet.
 - Ajouter la demande en base de données avec une validation préalable par le client.
 - Faire une estimation du prix du lavage en fonction de la catégorie de l'objet identifiée par Google Lens.
- **Soumission de la Photo :**
 - Le client peut soumettre une demande en prenant une photo de l'objet à nettoyer via l'application.
 - Intégration de Google Lens pour la reconnaissance d'image afin d'identifier la catégorie de l'objet.
- **Validation Préalable par le Client :**
 - Avant d'ajouter la demande en base de données, le client peut valider ou ajuster les informations identifiées par Google Lens.
 - Le client peut également fournir des détails supplémentaires sur la demande, tels que le niveau de saleté, les instructions spécifiques, etc.
- **Ajout en Base de Données :**
 - Une fois validée par le client, la demande est ajoutée à la base de données avec les détails de la photo, de la catégorie identifiée et des informations supplémentaires.
- **Estimation du Prix :**
 - En fonction de la catégorie de l'objet identifiée par Google Lens, le système effectue une estimation automatique du prix du lavage.
 - Le client peut voir une estimation du coût avant de confirmer la demande.