# **Cahier des Charges : Migration et Développement de l'Application de Gestion des Relevés de Prix**

## 1. ****Contexte et objectifs du projet****

### 1.1 ****Contexte****

L'entreprise S2M souhaite moderniser son application de gestion des relevés de prix concurrents. L'application existante, développée en PHP, présente certaines limitations en termes de performance, évolutivité, et sécurité. Le projet consiste à migrer cette application vers le framework Django tout en ajoutant de nouvelles fonctionnalités, optimisant ainsi la gestion des relevés de prix.

### 1.2 ****Objectifs****

L'objectif principal est de créer une application robuste, performante, et sécurisée qui permette :

* La gestion efficace des relevés de prix concurrents.
* Une interface utilisateur intuitive et moderne.
* Des fonctionnalités étendues pour la manipulation et l’analyse des données (recherches, filtres, import/export).
* La gestion avancée des articles et des zones concurrentes.

## 2. ****Périmètre fonctionnel****

### 2.1 ****Fonctionnalités principales de la gestion des relevés****

1. **Formulaire d'ajout de relevé** :
   * Interface permettant de créer un nouveau relevé avec les informations nécessaires (date, enseigne, zone, etc.).
2. **Recherche multicritères des relevés** :
   * Moteur de recherche multicritères sur les relevés, avec possibilité de filtrer par date, enseigne, zone, état (en attente, en cours, validé, chargé).
   * Affichage d’une liste paginée des relevés.
3. **Suppression d'un relevé** :
   * Fonctionnalité permettant de supprimer un relevé, avec confirmation de suppression pour éviter les suppressions accidentelles.
4. **Modification des états d'un relevé** :
   * Changement des états des relevés :
     + **En cours** : Le relevé est en cours de modification.
     + **Chargeable** : Le relevé est prêt à être chargé dans le système.
     + **En attente** : Le relevé attend des données supplémentaires ou une validation.
     + **Validé** : Le relevé a été approuvé.
     + **Chargé** : Le relevé a été intégré dans le système principal.
5. **Détail d'un relevé** :
   * Affichage des articles inclus dans un relevé.
   * Comparaison dans un tableau entre les articles de l’entreprise et ceux des concurrents pour chaque relevé.
   * Possibilité d'importer des articles dans un relevé, avec des critères de sélection (rayon, référence article, secteur, famille, sous-famille).
   * Sélection des articles à partir d’une autre base de données avec vérification des articles existants avant l’insertion.
   * Modification des articles si le relevé n'est pas encore validé, incluant la modification des articles concurrents.
   * Affichage des photos des articles pour une meilleure visualisation.
6. **Importation des articles concurrents** :
   * Importation des articles concurrents non encore rattachés aux articles détaillés dans le relevé.
   * Système de vérification pour identifier et associer les nouveaux articles concurrents.
7. **Export des relevés** :
   * Exportation des relevés et des détails en formats **Excel**, ou **PDF**.

### 2.2 ****Fonctionnalités de gestion des zones et enseignes****

1. **Gestion des zones** :
   * Liste des zones géographiques avec recherche multicritère et exportation des données en PDF, Excel.
   * Ajout, modification des zones.
2. **Gestion des enseignes concurrentes** :
   * Liste des enseignes concurrentes avec recherche multicritère et exportation des données en PDF, Excel.
   * Ajout, modification des enseignes.

### 2.3 ****Fonctionnalités de gestion des articles****

1. **Articles S2M** :

* **Liste des articles S2M :**
  + Affichage de la liste des articles de S2M avec une recherche multicritère par rayon, gencode, référence, et fournisseur.
* **Recherche multicritère :**
  + Possibilité de filtrer les articles selon différents critères (rayon, secteur, fournisseur).
* **Rattachement des articles concurrents :**
  + Gestion du rattachement entre les articles S2M et les articles des concurrents.
* **Historique des articles :**
  + Affichage de l'historique des actions (modifications, rattachements) effectuées sur chaque article.
* **Import des articles :**
  + Import des articles S2M via un fichier Excel pour rattacher des articles concurrents.
* **Exportation des articles :**
  + Exportation des données des articles en PDF, Excel avec les articles rattachés ou non.

1. **Articles concurrents** :

* **Recherche d'articles concurrents :**
  + Recherche selon des critères comme l'enseigne concurrente, l'article rattaché ou non.
* **Rattachement des articles non encore rattachés :**
  + Gestion des rattachements des articles concurrents aux articles S2M non encore rattachés.
* **Import de rattachements concurrents :**
  + Importation en masse des articles concurrents via Excel pour rattachement.
* **Export des articles concurrents :**
  + Export en Excel et PDF des articles concurrents rattachés ou non.

### 2.4 ****Fonctionnalités de récapitulatif et d’analyse****

1. **Récapitulatif des actions** :
   * Affichage d'un tableau récapitulatif des actions passées sur les articles et relevés.
   * Comparaison entre les prix de S2M et ceux des concurrents.
2. **Filtrage des récapitulatifs** :
   * Filtrage par date, référence ou gencode pour analyser les écarts de prix entre S2M et les concurrents.
3. **Exportation des recapitulatifs** :
   * Export des récapitulatifs en Excel ou PDF.

## 3. ****Spécifications fonctionnelles détaillées****

### 3.1 ****Interface utilisateur****

* **Tableaux dynamiques** : Affichage des relevés et des articles dans des tableaux interactifs, avec pagination, filtrage, et tri dynamique.
* **Formulaires intuitifs** : Simplification des formulaires pour l’ajout/modification des relevés et articles, avec validation en temps réel.
* **Import/Export** : Support des formats CSV, Excel, et PDF pour l’importation et l’exportation des données.
* **Gestion des états** : Interface de gestion des états des relevés (sélection par liste déroulante).
* **Affichage des images** : Possibilité d'afficher les images des articles directement dans l’interface.

### 3.2 ****Back-end et base de données****

* **Django ORM** : Utilisation de Django ORM pour la gestion des modèles et des relations entre les tables.
* **Vérification des doublons** : Système de vérification des doublons lors de l’importation de relevés ou d’articles concurrents.
* **Optimisation des requêtes SQL** : Mise en place de requêtes optimisées pour gérer efficacement les grands volumes de données.
* **Sécurisation des imports/exports** : Gestion sécurisée des imports et exports pour éviter les erreurs ou corruptions de données.

## 4. ****Contraintes et exigences techniques****

### 4.1 ****Sécurité****

 **Gestion des utilisateurs :**

* Authentification via login et mot de passe crypté.
* Gestion des rôles avec différents niveaux d'accès pour les administrateurs et les utilisateurs.

 **Protection des données :**

* Protection contre les attaques CSRF et XSS.
* Gestion des sessions sécurisées avec tokens.

### 4.2 ****Performance****

* **Optimisation du temps de réponse** : Réduction des temps de chargement par l’optimisation des requêtes et l’utilisation du cache.
* **Scalabilité** : Prévoir la capacité de l’application à supporter une augmentation du nombre d’utilisateurs et de données.

### 4.3 ****Compatibilité****

* **Navigateurs supportés** : Assurer la compatibilité avec les navigateurs récents (Chrome, Firefox, Edge).
* **Responsive design** : L’application doit être accessible et fonctionnelle sur différents types d’appareils (ordinateurs, tablettes, smartphones).

## 5. ****Technologies et outils****

* **Backend** : Python, Django
* **Base de données** : PostgreSQL et MySQL.
* **Frontend** : HTML5, CSS3, JavaScript (avec Bootstrap pour une interface responsive).
* **Outils d'import/export** : Gestion des fichiers CSV et Excel pour l’import/export des relevés.
* **Gestion des images** : Utilisation de Django pour la gestion des images (relevés d'articles).

## 6. ****Étapes et planification du projet****

1. **Analyse de l'application existante** :
   * Analyse des fonctionnalités existantes en PHP.
   * Revue de la structure actuelle de la base de données.
2. **Conception de la base de données Django** :
   * Conversion de la base de données MySQL existante en Django ORM.
   * Modélisation des entités et relations sous Django.
3. **Développement des fonctionnalités clés** :
   * Implémentation des fonctionnalités de gestion des relevés, articles, et utilisateurs sous Django.
4. **Migration des données** :
   * Migration des données existantes de PHP/MySQL vers Django/PostgreSQL (ou MySQL).
5. **Développement de l'interface utilisateur** :
   * Refonte de l'interface en utilisant un design moderne et responsive.
6. **Tests et validation**:

* Vérification de l'absence d'erreurs majeures dans l'application.
* Test d'enregistrement correct des données dans la base.
* Tests sur différents navigateurs (Chrome, Firefox, Edge, Safari).
* Mesure des temps de chargement.
* Test complet du système et ajustements.

## 7. ****Conclusion****

Ce cahier des charges définit les objectifs et les exigences pour la migration de l'application de gestion des relevés de prix vers Django. Le projet vise à améliorer les performances et la sécurité tout en ajoutant des fonctionnalités clés pour faciliter la gestion des articles et des relevés dans un environnement plus moderne et flexible.