

Cahier des charges – Application WPF GestionOceanBijoux

Cahier des charges – Application WPF GestionOceanBijoux.....	1
1. Présentation du projet.....	2
2. Objectifs.....	2
3. Fonctionnalités principales.....	2
3.1. Interface utilisateur (Front-End local).....	2
3.2. Gestion du catalogue (Back-End via API).....	3
4. Contenu.....	3
5. Contraintes techniques.....	3
6. Maintenance et évolutivité.....	4
7. Validation du projet.....	4

1. Présentation du projet

★ **Nom du projet** : GestionOceanBijoux

★ **Objectif principal** : Développer une application de bureau en WPF (C#, .NET) permettant la gestion complète des produits affichés sur le site e-commerce *Océan de Bijoux*, via une interaction sécurisée avec la base de données distante à l'aide d'une API Laravel.

Cette application s'inscrit dans une logique de continuité du projet web, avec une interface locale optimisée pour la gestion en back-office.

2. Objectifs

★ **Gestion optimisée** : Proposer une solution complète pour visualiser, modifier, supprimer et bientôt ajouter les produits affichés sur le site.

★ **Connexion à distance** : Communiquer avec l'API Laravel pour lire et écrire dans la base de données MySQL hébergée sur o2switch.

★ **Architecture professionnelle** : Utiliser une architecture MVVM, modulaire, réutilisable et maintenable.

★ **Interopérabilité** : Maintenir la compatibilité entre l'application de gestion et la plateforme e-commerce en ligne.

3. Fonctionnalités principales

3.1. Interface utilisateur (Front-End local)

★ **Design** :

- Interface claire, moderne et fonctionnelle, développée en XAML.
- Organisation logique des vues : tableau de bord produits, formulaire d'édition, etc.

★ **Affichage des produits** :

- Détails visibles : nom, prix, description, stock, image (optionnel).

★ **Édition des produits** :

- Formulaire de modification sécurisé et pré-rempli.
- Suppression des produits avec confirmation.

★ **Fonctionnalités à venir :**

- Ajout de produits.
- Recherche textuelle et avancée.

3.2. Gestion du catalogue (Back-End via API)

★ **Connexion API Laravel :**

- Appels GET/POST/PUT/DELETE sécurisés.
- Traitement des réponses JSON.

★ **Synchronisation distante :**

- Affichage en temps réel des données disponibles en base.
- Répercussion automatique des modifications effectuées dans l'application sur le site.

4. Contenu

★ **Données gérées :**

- Produits : nom, description, prix, stock, image, identifiants API, catégorie.
- Prévision d'intégration des commandes, utilisateurs, et catégories dans de futures versions.

★ **Documentation :**

- Cahier des charges et documentation technique publiés sur :
<https://oceane.ifcsioslam.fr#xpPro>

★ **Dépôt Git :**

- Code source complet disponible sur GitHub :
<https://github.com/oce75ndb/GestionOceanBijoux>

5. Contraintes techniques

★ **Technologies :**

- Framework : WPF (.NET 6+)
- Langage : C#
- Pattern : MVVM
- API REST Laravel (hébergée sur o2switch)
- Base de données MySQL

★ **Outils utilisés :**

- Visual Studio 2022
- Postman (tests API)
- GitHub (versionning)

★ **Compatibilité :**

- Windows 10 et versions supérieures

6. Maintenance et évolutivité

★ **Maintenance corrective :**

- Prise en charge des erreurs, tests manuels, logs d'exception.

★ **Maintenance évolutive :**

- Ajout de fonctionnalités : création de produits, gestion des catégories, commandes, statistiques visuelles.
- Application conçue pour une scalabilité du code et des vues.

7. Validation du projet

★ **Épreuve BTS SIO SLAM – E6**

- Présentation prévue : **26 mai 2025 – IFC Avignon**
- Réalisation professionnelle n°2