

# Cahier des charges – PoC de gestion des stocks ETML

---

## 1. Contexte et objectifs

### Contexte :

L'objectif principal de ce projet est de réaliser une **Preuve de Concept (PoC)** pour un système de gestion des stocks destiné à l'ETML. Ce PoC servira de base pour identifier les éventuelles difficultés techniques qui pourraient survenir lors du développement complet du projet, notamment dans le cadre du TPI.

### Objectifs :

- Évaluer la faisabilité technique d'une application de gestion des stocks en C# avec Windows Forms.
- Identifier et documenter les défis potentiels liés à la conception, au développement et à la mise en œuvre de l'application.
- Proposer des solutions ou des pistes d'amélioration pour résoudre ces problèmes avant le TPI.

## 2. Livrables attendus

Le livrable final doit être une **application fonctionnelle en C# (Windows Forms)** qui intègre les fonctionnalités essentielles suivantes :

### 1. Gestion des articles :

- Ajout, modification, suppression et consultation des articles en stock.
- Champs obligatoires : nom, description, quantité, catégorie, prix unitaire.

### 2. Suivi des mouvements de stock :

- Enregistrement des entrées (ajouts) et sorties (retraits) de stock.
- Historique des mouvements avec date et type (entrée/sortie).

### 3. Interface utilisateur :

- Une interface simple, intuitive et ergonomique pour faciliter l'utilisation.

### 4. Intégration d'une base de données :

- Mise en place d'une base de données locale (ex. : SQL Server ou SQLite) pour stocker les informations.

### 5. Sécurité et gestion des exceptions :

- Gestion des erreurs (ex. : saisie incorrecte, connexion à la base de données échouée).
- Protection des données sensibles (ex. : mot de passe administrateur).

### 6. Optimisation :

- Temps de réponse rapide pour les opérations CRUD (Create, Read, Update, Delete).

## 3. Liste des difficultés à identifier

Voici les principales difficultés techniques qu'il convient d'identifier et de résoudre dans le cadre de ce PoC :

### 1. Conception de l'interface utilisateur :

- Comment concevoir une interface ergonomique tout en respectant les contraintes de simplicité ?
- Quelles sont les meilleures pratiques pour organiser les formulaires et rendre l'expérience utilisateur fluide ?

## 2. Intégration de la base de données :

- ✓ Quelle solution de base de données choisir (SQL Server, SQLite, etc.) et comment l'intégrer efficacement ?
- ✓ Comment structurer la base de données pour assurer une bonne organisation des données (tables, relations, index) ?
- ✓ Quelles sont les bonnes pratiques pour écrire des requêtes SQL optimisées ?

## 3. Fonctionnalités CRUD :

- ✓ Comment implémenter les opérations CRUD de manière robuste et sans erreur ?
- ✓ Quelles validations doivent être mises en place pour éviter les données invalides ou corrompues ?

## 4. Suivi des mouvements de stock :

- ✓ Comment gérer les entrées et sorties de stock tout en conservant un historique précis ?
- ✓ Faut-il intégrer des alertes pour les stocks critiques ou des seuils de réapprovisionnement ?

## 5. Gestion des exceptions et sécurité :

- ✓ Comment traiter les erreurs techniques (ex. : connexion à la base de données rompue, fichier manquant) ?
- ✓ Quelles mesures de sécurité mettre en place pour protéger les données sensibles ?
- ✓ Faut-il implémenter une authentification pour restreindre l'accès à certaines fonctionnalités ?

## 6. Optimisation des performances :

- ✓ Comment garantir que l'application reste réactive même avec un volume croissant de données ?
- ✓ Quelles stratégies d'optimisation peuvent être appliquées (indexation, pagination, cache, etc.) ?

## 7. Documentation et maintenance :

- ✓ Comment documenter le code et les choix techniques pour faciliter la maintenance future ?
- ✓ Quelles sont les bonnes pratiques pour garantir une évolutivité du projet ?

## 4. Critères de réussite

Pour valider ce PoC, les critères suivants doivent être remplis :

- L'application doit être fonctionnelle et répondre aux exigences décrites dans la section "Livrables attendus".
- Les difficultés identifiées doivent être clairement documentées, avec des propositions de solutions ou des recommandations pour les résoudre.
- Le temps de réponse de l'application doit être acceptable pour un usage quotidien.
- L'interface utilisateur doit être intuitive et accessible, même pour des utilisateurs non techniques.