



# CAHIER DES CHARGES

## Literie Processor

### Application de Traitement Automatisé des Devis Matelas

Développé par SCINNOVA pour SAS Literie Westelynck

Projet	Literie Processor (anciennement MatelasApp)
Client	SAS Literie Westelynck
Développeur	SCINNOVA
Version	3.0.1
Date	17/07/2025
Contact	sebastien.confrere@scinnova.fr

# Table des Matières

- 1. Présentation du projet
  - 1.1 Contexte et objectifs
  - 1.2 Problématique
  - 1.3 Solution proposée
- 2. Spécifications fonctionnelles
  - 2.1 Fonctionnalités principales
  - 2.2 Types de matelas supportés
  - 2.3 Fonctionnalités avancées
- 3. Spécifications techniques
  - 3.1 Architecture générale
  - 3.2 Technologies utilisées
  - 3.3 Structure des données
  - 3.4 Performance et scalabilité
- 4. Contraintes et exigences
  - 4.1 Contraintes techniques
  - 4.2 Contraintes fonctionnelles
  - 4.3 Contraintes organisationnelles
- 5. Livrables
- 6. Planning et phases
- 7. Tests et validation
- 8. Maintenance et évolution
- 9. Risques et mitigation
- 10. Conclusion

# 1. Présentation du projet

## 1.1 Contexte et objectifs

**Projet** : Literie Processor (anciennement MatelasApp)

**Client** : SAS Literie Westelynck

**Développeur** : SCINNOVA

**Date de création** : 2025

**Version actuelle** : 3.0.1

## 1.2 Problématique

La SAS Literie Westelynck reçoit quotidiennement de nombreux devis de matelas au format PDF. Le traitement manuel de ces devis présente plusieurs défis majeurs :

- Chronophage : Plusieurs heures par jour de traitement manuel
- Source d'erreurs humaines : 15-20% de corrections nécessaires
- Difficile à standardiser : Processus non uniforme
- Coûteux en ressources humaines : 2000-3000€/mois
- Délais de production : Retards dus au traitement manuel

## 1.3 Solution proposée

Développement d'une application automatisée capable de :

- Analyser automatiquement les devis PDF
- Extraire les informations structurées
- Générer des fichiers Excel prêts pour l'inscription
- Standardiser le processus de traitement
- Réduire drastiquement les erreurs et le temps de traitement

## 2. Spécifications fonctionnelles

### 2.1 Fonctionnalités principales

#### 2.1.1 Traitement des devis PDF

- Extraction de texte : Lecture et extraction du contenu textuel des fichiers PDF
- Analyse LLM : Utilisation d'intelligence artificielle pour comprendre et structurer les informations
- Validation : Vérification de la cohérence des données extraites
- Gestion d'erreurs : Traitement robuste des cas particuliers et des erreurs

#### 2.1.2 Génération de fichiers Excel

- Templates spécialisés : Modèles Excel adaptés aux différents types de matelas
- Remplissage automatique : Insertion des données dans les cellules appropriées
- Coloration conditionnelle : Mise en forme visuelle selon les caractéristiques
- Validation des données : Vérification de la cohérence avant export

#### 2.1.3 Interface utilisateur

- Interface graphique moderne : Application PyQt6 avec design professionnel
- Sélection de fichiers : Interface drag & drop et sélection multiple
- Suivi en temps réel : Barre de progression et logs détaillés
- Gestion des résultats : Affichage et export des fichiers générés

### 2.2 Types de matelas supportés

#### 2.2.1 Latex Naturel

- Caractéristiques : Densité, épaisseur, dimensions
- Calculs spécifiques : Prix selon référentiel JSON
- Options : Différentes fermetés et finitions

#### 2.2.2 Latex Mixte 7 Zones

- Zones de confort : 7 zones différenciées
- Matériaux : Latex naturel + synthétique
- Calculs : Prix par zone et global

#### 2.2.3 Latex Renforcé

- Renforcement : Structure renforcée
- Applications : Usage intensif
- Prix : Majoration selon référentiel

#### 2.2.4 Mousse Viscoélastique

- Propriétés : Mémoire de forme
- Densités : Différentes options disponibles
- Calculs : Prix selon densité et épaisseur

### **2.2.5 Mousse Rainurée 7 Zones**

- Rainures : Structure aérée
- Zones : 7 zones de confort
- Ventilation : Amélioration de l'aération

### **2.2.6 Select 43**

- Matériau spécial : Mousse haute densité
- Applications : Usage professionnel
- Calculs : Prix spécifique

## 3. Spécifications techniques

### 3.1 Architecture générale

#### 3.1.1 Frontend (Interface utilisateur)

- Framework : PyQt6
- Design : Interface moderne et intuitive
- Responsive : Adaptation à différentes résolutions
- Thème : Intégration du logo Westelynck

#### 3.1.2 Backend (Traitement)

- Langage : Python 3.8+
- Modules spécialisés : Un module par type de matelas
- API LLM : Intégration multi-providers
- Gestion des données : JSON et Excel

#### 3.1.3 Communication

- Interface : backend\_interface.py
- Sérialisation : JSON pour les échanges
- Gestion d'erreurs : Exceptions structurées

### 3.2 Technologies utilisées

#### 3.2.1 Core Python

- PyPDF2 : Lecture des fichiers PDF
- openpyxl : Manipulation des fichiers Excel
- requests : Appels API HTTP
- json : Sérialisation des données

#### 3.2.2 Interface graphique

- PyQt6 : Framework GUI principal
- Qt Designer : Conception des interfaces
- QThread : Traitement asynchrone
- QProgressBar : Indicateurs de progression

#### 3.2.3 Intelligence artificielle

- OpenAI API : GPT-4 pour l'analyse
- OpenRouter : Alternative multi-modèles
- Prompts spécialisés : Instructions détaillées pour l'extraction

### **3.2.4 Sécurité et stockage**

- Cryptographie : Chiffrement des clés API
- Configuration : Fichiers JSON sécurisés
- Logs : Traçabilité complète

## **4. Contraintes et exigences**

### **4.1 Contraintes techniques**

#### **4.1.1 Compatibilité**

- Systèmes d'exploitation : Windows 10+, macOS 10.15+, Linux
- Python : Version 3.8 ou supérieure
- Mémoire : Minimum 4 GB RAM
- Espace disque : 500 MB pour l'installation

#### **4.1.2 Sécurité**

- Chiffrement : AES-256 pour les clés API
- Validation : Vérification des entrées utilisateur
- Isolation : Pas d'accès réseau non autorisé

#### **4.1.3 Performance**

- Temps de réponse : < 2 secondes pour l'interface
- Traitement : < 30 secondes par devis
- Mémoire : Gestion optimisée des ressources

### **4.2 Contraintes fonctionnelles**

#### **4.2.1 Précision**

- Taux de réussite : > 95% d'extraction correcte
- Validation : Vérification automatique des données
- Correction : Possibilité de correction manuelle

#### **4.2.2 Fiabilité**

- Robustesse : Gestion des cas d'erreur
- Récupération : Sauvegarde automatique
- Logs : Traçabilité complète



## 5. Livrables

### 5.1 Application principale

- Exécutable standalone : Installation simple
- Interface graphique : PyQt6 moderne
- Documentation : Guide utilisateur complet

### 5.2 Documentation technique

- Cahier des charges : Ce document
- Documentation développeur : Guide technique
- API documentation : Référence des modules

### 5.3 Fichiers de configuration

- Templates Excel : Modèles pour chaque type
- Référentiels JSON : Données de prix et calculs
- Mappings : Correspondances personnalisables

### 5.4 Outils de maintenance

- Scripts de diagnostic : Outils de dépannage
- Système de logs : Traçabilité complète
- Utilitaires : Outils d'administration

## 6. Planning et phases

### 6.1 Phase 1 : Développement initial (Terminée)

- Architecture : Structure modulaire
- Core modules : Modules de base
- Interface : GUI PyQt6
- Tests : Validation fonctionnelle

### 6.2 Phase 2 : Optimisation (En cours)

- Performance : Optimisation des traitements
- Robustesse : Gestion d'erreurs avancée
- Documentation : Guides complets
- Packaging : Distribution standalone

### 6.3 Phase 3 : Évolutions futures

- Nouveaux types : Support de nouveaux matelas
- API REST : Interface web
- Cloud : Version SaaS
- Mobile : Application mobile

## **7. Tests et validation**

### **7.1 Tests unitaires**

- Modules backend : Tests de chaque fonction
- Calculs : Validation des formules
- API : Tests des appels externes

### **7.2 Tests d'intégration**

- Workflow complet : PDF → Excel
- Interface : Tests utilisateur
- Performance : Tests de charge

### **7.3 Tests de validation**

- Cas réels : Tests avec vrais devis
- Précision : Vérification des extractions
- Robustesse : Tests d'erreurs

## **8. Maintenance et évolution**

### **8.1 Maintenance préventive**

- Monitoring : Surveillance des performances
- Logs : Analyse des erreurs
- Mises à jour : Corrections et améliorations

### **8.2 Maintenance corrective**

- Bugs : Correction des erreurs
- Compatibilité : Adaptation aux évolutions
- Sécurité : Corrections de vulnérabilités

### **8.3 Évolutions**

- Nouvelles fonctionnalités : Ajouts demandés
- Optimisations : Amélioration des performances
- Intégrations : Nouvelles APIs

## **9. Risques et mitigation**

### **9.1 Risques techniques**

- Évolution des APIs → Monitoring et adaptation
- Compatibilité → Tests réguliers
- Performance → Optimisation continue

### **9.2 Risques fonctionnels**

- Précision LLM → Amélioration des prompts
- Évolution des devis → Adaptation des modèles
- Formats → Support de nouveaux formats

### **9.3 Risques organisationnels**

- Formation → Documentation complète
- Support → Contact technique dédié
- Évolution → Roadmap claire

## 10. Conclusion

Le projet Literie Processor répond parfaitement aux besoins de la SAS Literie Westelynck en automatisant le traitement des devis PDF. L'application combine technologies modernes (IA, interface graphique) et robustesse technique pour offrir une solution complète et évolutive.

### Points forts :

- Automatisation complète du processus
- Précision élevée grâce à l'IA
- Interface utilisateur intuitive
- Architecture modulaire et évolutive
- Documentation complète

### Perspectives :

- Extension à d'autres types de documents
- Interface web pour accès distant
- Intégration avec les systèmes existants
- Développement d'une version SaaS

# Cahier des Charges

**Literie Processor**

**Développé par SCINNOVA**

**Pour SAS Literie Westelynck**

Merci de votre confiance

Contact technique	sebastien.confrere@scinnova.fr
Téléphone	06.66.05.72.47
Éditeur	SCINNOVA
Client	SAS Literie Westelynck