

# Cahier des Charges Fonctionnel et Technique - ERP de Gestion de Laboratoire Hospitalier

## 1. Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles pour le développement d'un ERP (Enterprise Resource Planning) destiné à la gestion d'un laboratoire hospitalier. L'objectif est de fournir une solution complète et intégrée pour optimiser les opérations du laboratoire, de la gestion des patients et des examens à la gestion des stocks et des ressources humaines.

## 2. Contexte et Objectifs

L'ERP sera un système multi-tenant, permettant à plusieurs entités (laboratoires ou départements) de partager la même infrastructure logicielle tout en garantissant l'isolation et la sécurité de leurs données respectives. Les principaux objectifs sont :

- Optimiser la gestion des patients, des médecins et des examens.
- Assurer une traçabilité complète des demandes d'examens et des résultats.
- Gérer efficacement les stocks de réactifs et consommables.
- Simplifier la facturation et le suivi des paiements.
- Faciliter la gestion des horaires et des présences des employés.
- Fournir un système sécurisé et performant.

## 3. Exigences Fonctionnelles

### 3.1. Module de Gestion des Patients

Ce module permettra de gérer l'ensemble des informations relatives aux patients du laboratoire.

- **Création, modification, consultation et suppression** d'une fiche patient.
- **Recherche** de patients par différents critères (nom, prénom, date de naissance, code, etc.).
- **Gestion des types de patients** (résident, ambulant) : Pour les patients résidents, le médecin de l'hôpital sera enregistré. Pour les patients ambulants, le nom et prénom du

médecin prescripteur externe seront saisis.

- **Historique** des demandes d'examens pour un patient.
- **Gestion des informations de facturation** (assurances, mutuelles, paiement direct).

### 3.2. Module de Gestion des Médecins

Ce module gèrera les informations des médecins internes et l'association des médecins externes aux demandes.

- **Création, modification, consultation et suppression** d'une fiche médecin (pour les médecins internes).
- **Recherche** de médecins par différents critères (nom, prénom, spécialité, numéro d'identification).
- **Gestion des informations de contact** et de spécialité.
- **Saisie simple du nom et prénom** pour les médecins prescripteurs externes lors de la demande d'examen.

### 3.3. Module de Gestion des Examens

Ce module permettra de définir et de gérer les différents types d'examens proposés par le laboratoire.

- **Création, modification, consultation et suppression** des types d'examens.
- **Recherche** d'examens par nom, code, etc.
- **Gestion des détails de chaque examen** : nom, code, description, prix, délai de rendu estimé, unités de mesure, valeurs de référence (éventuellement par âge ou sexe), type d'échantillon requis, conditions pré-analytiques, équipement ou réactifs nécessaires.
- **Association des examens aux articles de stock consommés** : Spécification de la quantité d'article consommée par examen.

### 3.4. Module de Gestion des Demandes d'examens

Ce module couvrira le processus de gestion des demandes d'examens, de la prescription à la validation des résultats.

- **Création** d'une nouvelle demande d'examen pour un patient (avec lien vers le patient et le médecin prescripteur).
- **Ajout de plusieurs examens** à une même demande.
- **Modification et consultation** des détails d'une demande (date, statut, notes facultatives).

- **Suivi du statut de la demande** : en attente, en cours, terminée, annulée.
- **Enregistrement des résultats** pour chaque examen associé à une demande.
- **Génération automatique d'une facture** après validation des examens réalisés.
- **Gestion des dates de prélèvement** et de réception de l'échantillon au laboratoire.

### 3.5. Module de Gestion des Stocks

Ce module permettra de gérer les réactifs, consommables et autres articles du laboratoire.

- **Gestion des articles en stock** : création, modification, consultation, suppression.
- **Catégorisation des articles** : Produits de nettoyage et de désinfection, réactifs, consommables, médicaments, équipements de laboratoire, substances contrôlées, etc.
- **Enregistrement des mouvements de stock** : entrées (réception de nouvelles livraisons) et sorties (utilisation pour les examens). Les informations importantes pour ces mouvements incluent la date, la quantité, l'utilisateur, le motif et la demande d'examen associée pour les sorties.
- **Suivi des quantités actuelles** et des seuils critiques.
- **Génération d'alertes automatiques** lorsque le stock atteint un seuil critique, avec un message prédéfini.
- **Recherche** d'articles par nom, catégorie, fournisseur.
- **Gestion des numéros de lot** et dates de péremption pour les réactifs.

### 3.6. Module de Facturation et Paiements

Ce module gèrera le processus de facturation et de suivi des paiements.

- **Génération automatique de factures** suite à la validation des examens réalisés (une demande génère une facture).
- **Gestion des statuts de facture** : en attente de paiement, payée, annulée, partiellement payée.
- **Enregistrement des paiements** : montant, date, méthode de paiement (carte bancaire, caisse, assurance).
- **Suivi des paiements partiels**.
- **Lien entre les factures** et les demandes d'examens.
- **Recherche** de factures par patient, date, statut.

### 3.7. Module de Gestion des Employés (Horaires et Présence)

Ce module permettra de gérer les informations des employés, leurs horaires et leur présence.

- **Création, modification, consultation et suppression** d'une fiche employé.
- **Gestion des informations personnelles** (nom, postnom, prénom, date de naissance, sexe, adresse, contact) et professionnelles (matricule, poste, service, date d'embauche).
- **Planification des horaires de travail** : horaire du mois (journalier ou par heure).
- **Enregistrement des présences** : pointages d'entrée/sortie selon l'horaire du mois et le jour.
- **Suivi des heures travaillées**.

### 3.8. Module de Gestion Multi-tenancy

Ce module gèrera les différentes entités (compagnies/laboratoires) au sein du système.

- **Création, modification, consultation** des informations de chaque compagnie (nom, adresse, contact, logo, secteur d'activité, type d'établissement, description).
- **Association des données** (utilisateurs, patients, examens, stocks, etc.) à une compagnie spécifique via le `company_id` dans chaque table (sauf `superadmins` et `permissions`).
- **Le superadmin crée une compagnie** avec un rôle admin et un utilisateur admin automatiquement avec un mot de passe par défaut.
- **Les permissions sont les mêmes** pour toutes les compagnies.

### 3.9. Module de Gestion des Utilisateurs, Rôles et Permissions

Ce module gèrera l'accès au système et les niveaux d'autorisation.

- **Gestion des utilisateurs** (création, modification, suppression) avec des champs tels que `telephone`, `sexe`, `photo_de_profil`, `last_login`.
- **Gestion des rôles** et association des rôles aux utilisateurs.
- **Gestion des permissions** définies par `action` et `module` (ex: (READ, PATIENT), (CREATE, PATIENT)).
- **Table `superadmins`** distincte pour les administrateurs système sans `company_id`.

### 3.10. Module d'Audit (Logs)

Ce module enregistrera toutes les actions significatives effectuées dans le système.

- **Enregistrement de toutes les actions** dans le système (type d'action, description, horodatage, utilisateur, adresse IP).

## **4. Exigences Non Fonctionnelles**

### **4.1. Performance**

- Les attentes spécifiques en termes de vitesse de réponse de l'application, de temps de chargement des pages, ou de traitement des requêtes seront définies ultérieurement.

### **4.2. Sécurité**

- Protection des données contre les accès non autorisés.
- Exigences de chiffrement et de conformité (à définir selon les normes locales et internationales, ex: RGPD).

### **4.3. Fiabilité et Disponibilité**

- Le niveau de disponibilité attendu sera défini ultérieurement.
- Mise en place d'une stratégie de sauvegardes et de récupération des données en cas de sinistre.

### **4.4. Évolutivité**

- L'application doit être conçue pour gérer une augmentation significative du nombre d'utilisateurs, de patients et de données à l'avenir.

### **4.5. Maintenabilité**

- Le code doit être bien documenté et facile à maintenir.

### **4.6. Utilisabilité**

- L'interface utilisateur doit être intuitive, facile à utiliser et accessible.

### **4.7. Intégration**

- Pas d'intégration avec des systèmes existants pour le moment.

### **4.8. Localisation**

- Le système doit supporter plusieurs langues.

### **4.9. Conformité**

- Les réglementations spécifiques au secteur de la santé seront prises en compte (à définir).

## 5. Architecture Technique

- **Base de données** : MySQL
- **Backend** : Spring (Java)
- **Frontend** : React JS

## 6. Modèle de Données

Le modèle de données a été défini et est disponible dans le fichier `schema.sql` et le diagramme `database_diagram.png`.

## 7. Glossaire

- **ERP** : Enterprise Resource Planning
- **Multi-tenancy** : Architecture logicielle où une seule instance de l'application sert plusieurs clients (tenants).
- **Superadmin** : Administrateur système ayant des privilèges sur toutes les compagnies.