

Système de Traçabilité des Produits Périssables

Formation : Génie Logistique

Porteur de projet :

Date : [Date]

Technologies : PHP, MySQL, HTML/CSS/JavaScript

1. Contexte et Objectifs

Contexte

Les produits périssables (aliments, médicaments) nécessitent un suivi rigoureux pour garantir leur qualité et sécurité. Ce projet vise à développer un système d'information (SI) web pour tracer ces produits tout au long de la chaîne logistique, en exploitant **PHP** pour le back-end et **MySQL** pour la base de données.

Objectifs

- **Principal** : Concevoir un SI web permettant de suivre en temps réel les produits périssables.
- **Secondaires** :
 - Identifier les étapes critiques (production, transport, stockage).
 - Générer des alertes en cas de non-respect des conditions de conservation.
 - Fournir un historique complet accessible aux acteurs (fournisseurs, distributeurs).

2. Périmètre du Projet

Fonctionnalités Incluses

- Gestion des produits (ajout, suppression, mise à jour).
- Suivi des températures et localisations via saisie manuelle ou intégration capteurs (simulée).
- Génération de codes QR pour identification unique.
- Alertes automatisées (email/SMS) en cas de dépassement de seuils.
- Tableau de bord analytique (graphiques, rapports).

Limites

- Pas d'intégration matérielle réelle de capteurs IoT (simulation via formulaires).
- Focus sur un segment local (pas de gestion multi-entrepôts internationaux).

3. Exigences Fonctionnelles

3.1 Modules du Système

1. **Module Admin :**
 - Gestion des utilisateurs (rôles : admin, fournisseur, transporteur).
 - Configuration des seuils critiques (température, durée de stockage).
 2. **Module Produits :**
 - Ajout de produits (nom, type, date de péremption, fournisseur).
 - Génération automatique de codes QR (liés à l'ID MySQL).
 3. **Module Traçabilité :**
 - Saisie des conditions de conservation (température, humidité, géolocalisation).
 - Historique des mouvements (entrepôt A → entrepôt B → magasin).
 4. **Module Alertes :**
 - Envoi d'emails aux responsables en cas de dépassement de seuils.
 5. **Module Reporting :**
 - Visualisation des données via graphiques (Chart.js ou équivalent).
-

4. Exigences Techniques

4.1 Front-End

- **Technologies :** HTML5, CSS3, JavaScript (pour l'interactivité).
- **Bibliothèques :**
 - **Bootstrap** pour le design responsive.
 - **QRCode.js** pour générer des codes QR côté client.

4.2 Back-End

- **Langage :** PHP (version 8.x recommandée).
- **Framework :** Optionnel (ex : Laravel pour une structure MVC, ou PHP natif).
- **Fonctionnalités clés :**
 - Sessions utilisateurs avec **PHP Session**.
 - Génération de PDF pour les rapports (avec **TCPDF** ou **DomPDF**).
 - Envoi d'emails via **PHPMailer**.

4.3 Base de Données

- **SGBD :** MySQL.
- **Structure des Tables :**
 - **Utilisateurs** : id, nom, email, mot_de_passe_hash, role.
 - **Produits** : id, nom, type, date_production, date_peremption, fournisseur_id.
 - **Traçabilité** : id, produit_id, temperature, localisation, date_mesure, utilisateur_id.
 - **Alertes** : id, produit_id, message, date_alerte, statut (non résolu/résolu).

4.4 Sécurité

- Hashage des mots de passe avec `password_hash()` (PHP).
 - Protection contre les injections SQL avec PDO ou MySQLi.
 - Validation des formulaires côté serveur (PHP) et client (JavaScript).
-

5. Contraintes

- **Budget** : Utilisation de technologies gratuites (XAMPP/WAMP pour le serveur local).
 - **Délai** : 4 à 6 semaines.
 - **Ressources** :
 - Serveur local (XAMPP) ou hébergement web mutualisé.
 - Éditeur de code (VS Code, PHPStorm).
-

6. Livrables

6.1 Code Source

- Application web fonctionnelle (dossier PHP/MySQL).
- Scripts SQL pour la création de la base de données.

6.2 Documentation

- **Manuel d'installation** : Configuration de XAMPP, import de la BDD.
- **Manuel utilisateur** : Guide des fonctionnalités par rôle.
- **Rapport technique** : Explication des choix technologiques, schéma de la BDD.

6.3 Démonstration

- Présentation du système avec cas d'utilisation réels (ex : suivi d'un lot de lait).
-

7. Planification

Phase	Durée	Tâches
Conception	1 semaine	Diagramme de cas d'utilisation, modèle de BDD.
Développement Back-End	2 semaines	Programmation PHP, création des tables MySQL.
Développement Front-End	1 semaine	Intégration HTML/CSS, connexion à la BDD.
Tests & Optimisation	1 semaine	Tests unitaires, résolution des bugs.

8. Exemple de Cas d'Utilisation

Scénario : Un transporteur saisit la température d'un camion transportant des vaccins.

1. Le transporteur se connecte via son compte.
 2. Il scanne le QR code du produit pour accéder à sa fiche.
-

3. Il saisit la température actuelle dans un formulaire PHP.
 4. Si la température dépasse 8°C, le système déclenche une alerte email à l'admin.
 5. L'admin consulte le tableau de bord pour voir l'historique du produit.
-

9. Critères de Validation

- Le système permet de suivre un produit de la production à la vente.
 - Les alertes sont générées correctement lors des dépassements de seuil.
 - L'interface est intuitive pour les utilisateurs non techniques.
-

10. Conclusion

Ce système répond aux enjeux logistiques de traçabilité tout en utilisant des technologies accessibles (PHP/MySQL). Il pourra être étendu à l'avenir avec des capteurs IoT réels et une application mobile.

Signature : [Responsable]

Validation : [Encadrant]

Ce cahier des charges guide le développement pas à pas, en alignant les besoins métier (logistique) avec les solutions techniques (PHP/MySQL).