

AVRIL 2024

# Documentation techniques Fovéa

LOGICIEL DE GESTION DES RÉFÉRENCES  
D'ARTICLES VIA LA RECONNAISSANCE  
D'IMAGES PAR IA

PROMOTION :  
2026

EQUIPE :  
Quentin ROBERT  
Maxime LECOMTE  
Nicolas SAINTY  
Florent CANALES



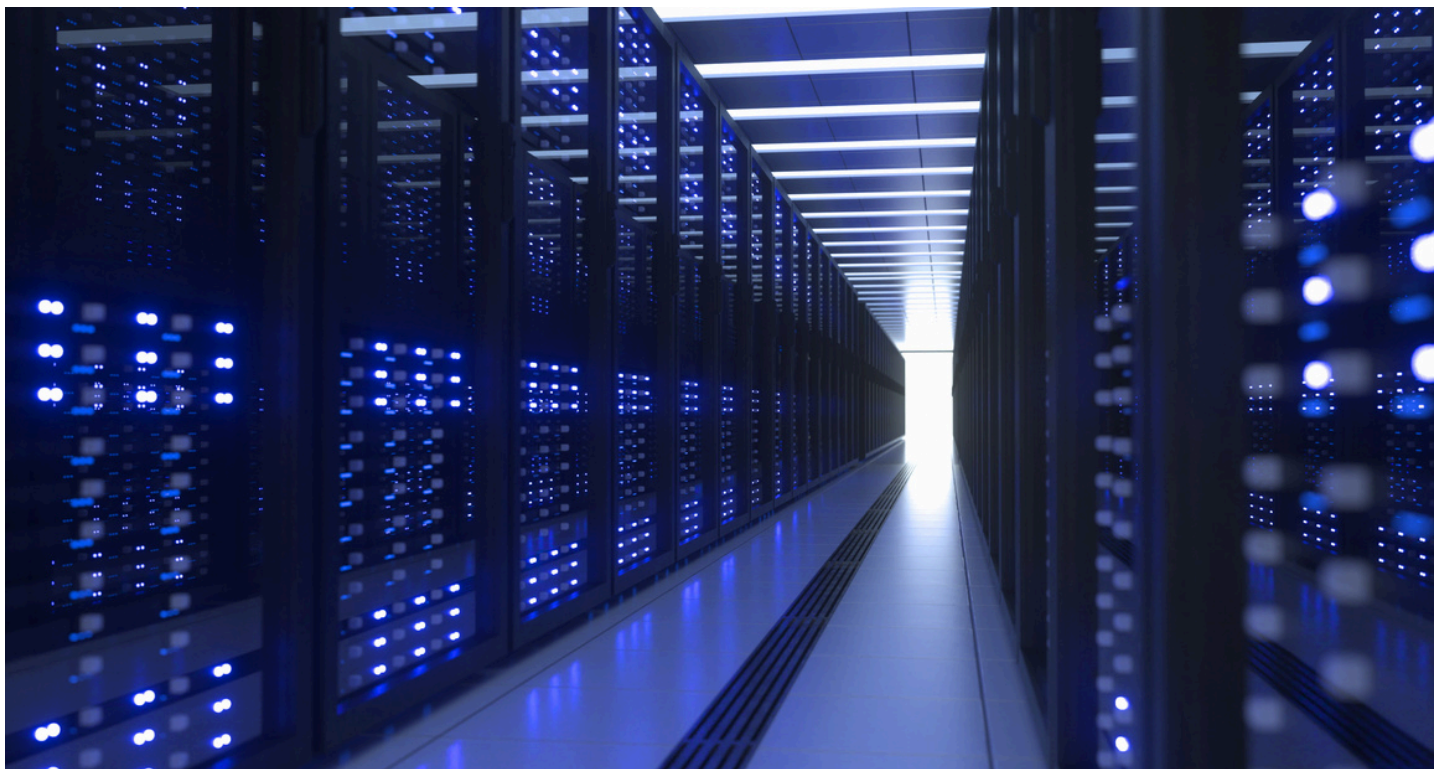
# Sommaire

<b><u>OBJECTIF DE LA DOCUMENTATION</u></b>	<b>03</b>
<b><u>VUE D'ENSEMBLE DU SYSTÈME DE SCAN D'ARTICLES</u></b>	<b>04</b>
<b><u>SCHÉMA TECHNIQUE</u></b>	<b>05</b>
<b><u>EXPLICATION TECHNIQUE ET JUSTIFICATION DES CHOIX DE TECHNOLOGIES</u></b>	<b>06</b>
<i>Reconnaissance des Produits</i>	06
<i>Base de Données</i>	07
<i>Synchronisation des Données entre les Tablettes</i>	08
<b><u>GUIDE D'INSTALLATION ET CONFIGURATION</u></b>	<b>09</b>
<i>Prérequis Matériels et Logiciels</i>	09
<i>Installation de l'API Express JS sur le Serveur Local</i>	10
<i>Configuration de la Base de Données MySQL</i>	10
<i>Configuration de l'Interface Front-end React sur les Tablettes</i>	10
<b><u>GUIDE DE DÉVELOPPEMENT</u></b>	<b>12</b>
<i>Structure du Code Source</i>	12
<i>Architecture Logicielle</i>	13
<i>Bonnes Pratiques de Développement</i>	13
<i>Gestion des Dépendances</i>	13
<b><u>INTÉGRATION ET DÉPLOIEMENT</u></b>	<b>14</b>
<i>Intégration des Nouveaux Magasins</i>	14
<i>Déploiement de l'Application</i>	15
<i>Gestion des Mises à Jour et des Versions</i>	15
<b><u>MAINTENANCE ET GESTION DES ERREURS</u></b>	<b>16</b>
<i>Maintenance du Système</i>	16
<i>Gestion des Erreurs et des Exceptions</i>	17
<i>Dépannage des Problèmes Techniques</i>	18
<b><u>SÉCURITÉ ET CONFIDENTIALITÉ</u></b>	<b>19</b>
<i>Gestion des Accès et des Autorisations</i>	19
<i>Protection des Données Sensibles</i>	19
<i>Surveillance de la Sécurité</i>	20
<i>Formation et Sensibilisation à la Sécurité</i>	20

# Objectif de la Documentation

Cette documentation a pour objectif de :

- Présenter les différentes fonctionnalités techniques de l'application de scan d'articles, telles que la reconnaissance des produits, la gestion de la base de données et la synchronisation des données entre les tablettes.
- Détailler l'architecture du système, en mettant en évidence les interactions entre ses différents composants.
- Expliquer les choix de technologies effectués pour chaque aspect du système, en justifiant leur pertinence par rapport aux besoins du projet.
- Servir de guide pratique pour l'installation, le développement, le déploiement, la maintenance et la sécurité du système.



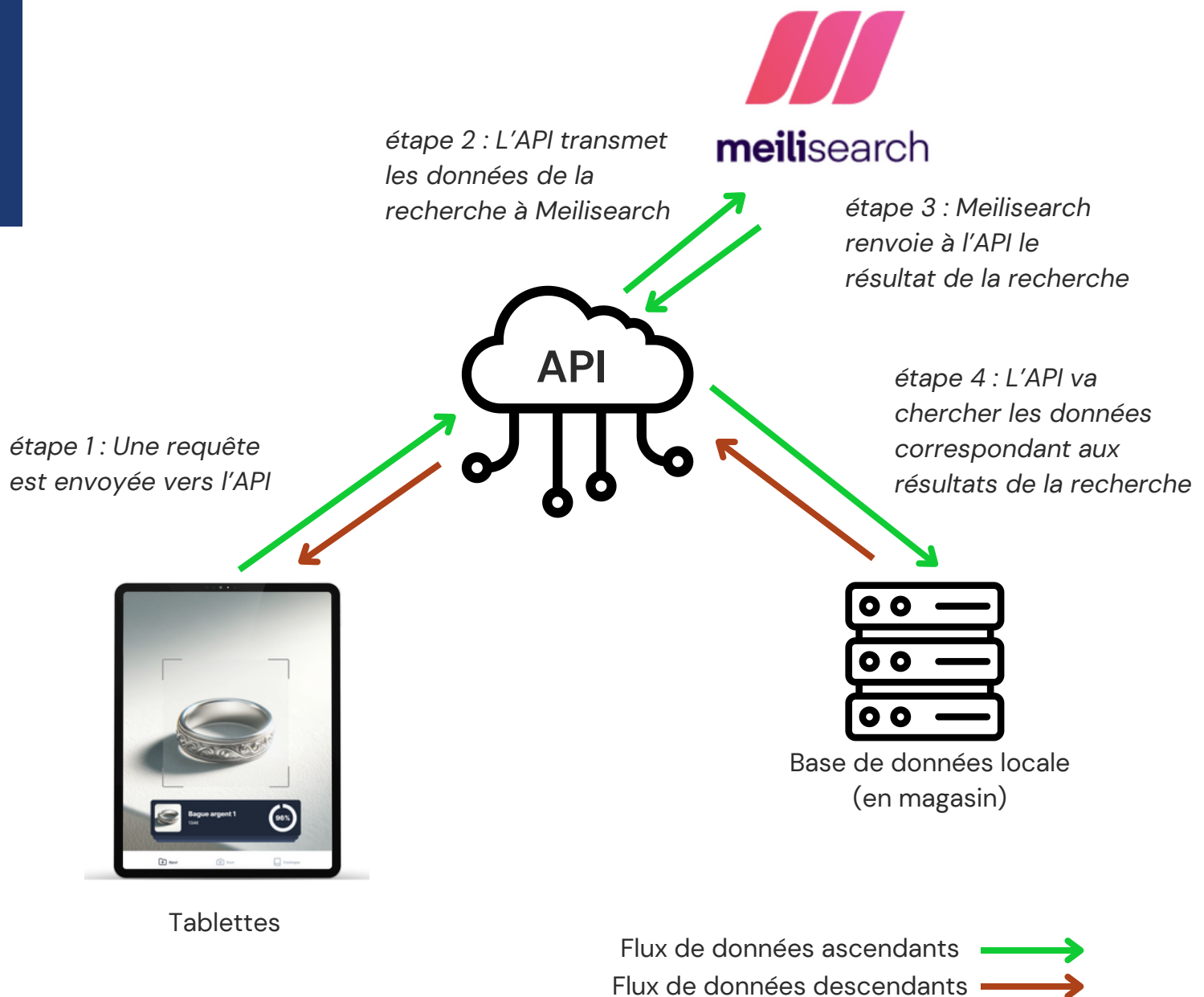
# Vue d'Ensemble du Système de Scan d'Articles

Le système de scan d'articles se compose de plusieurs composants interconnectés, offrant des fonctionnalités avancées pour les utilisateurs finaux. Parmi les principales fonctionnalités du système, on peut citer :

- Reconnaissance des produits : Capacité à reconnaître et à identifier automatiquement les produits à partir des scans effectués par les utilisateurs.
- Base de données : Stockage sécurisé et fiable des articles scannés et des données associées, permettant un accès rapide et efficace.
- Synchronisation des données : Synchronisation en temps réel des données entre les différentes tablettes utilisées dans les magasins, garantissant la cohérence et l'actualité des informations.

# Schéma Technique

Diagramme architectural du système de scan d'article



# Explication Technique et Justification des Choix de Technologies

## Reconnaissance des Produits

Meilisearch a été choisi comme moteur de recherche pour faciliter la reconnaissance des produits dans la base de données. Sa capacité à fournir des résultats de recherche rapides et précis, ainsi que sa simplicité d'intégration avec notre architecture, en ont fait un choix idéal pour cette fonctionnalité.

Meilisearch offre des performances de recherche exceptionnelles, même avec de grandes quantités de données, ce qui est essentiel pour assurer une reconnaissance rapide et efficace des produits. Sa simplicité d'intégration via des API RESTful permet une mise en œuvre facile et rapide dans notre architecture existante, réduisant ainsi le temps de développement et de déploiement.

Autres Technologies Évaluées :

- Elasticsearch : Bien qu'Elasticsearch soit largement utilisé pour la recherche en texte intégral, sa complexité et ses exigences en matière de ressources peuvent rendre sa mise en œuvre plus coûteuse et moins adaptée à notre cas d'utilisation spécifique.
- Algolia : Algolia offre des fonctionnalités de recherche similaires à Meilisearch, mais ses coûts associés peuvent être prohibitifs pour les projets de taille moyenne.

# Base de Données

MySQL a été choisi comme système de gestion de base de données relationnelles pour stocker les articles scannés et les données associées. Sa robustesse, sa fiabilité et sa compatibilité avec nos besoins en matière de stockage de données ont été des facteurs déterminants dans ce choix.

MySQL est largement reconnu pour sa stabilité et sa performance dans des environnements de production, offrant ainsi une solution fiable pour la gestion des données critiques de l'application.

Sa compatibilité avec un large éventail de langages de programmation et de frameworks a facilité son intégration avec notre architecture existante, permettant une transition en douceur vers notre système de scan d'articles.

Autres Technologies Évaluées :

- MongoDB : Bien que MongoDB offre une flexibilité et une évolutivité intéressantes, sa modélisation de données orientée document ne correspondait pas parfaitement aux besoins de notre application, qui nécessitait une structure de données relationnelle plus traditionnelle.
- PostgreSQL : PostgreSQL est également une option solide pour les bases de données relationnelles, mais sa courbe d'apprentissage plus raide et ses exigences en matière de maintenance ont été des considérations importantes dans notre décision de choisir MySQL.



# Synchronisation des Données entre les Tablettes

Express JS a été choisi comme framework pour développer l'API permettant la synchronisation des données entre les tablettes et les serveurs locaux. Sa légèreté, sa flexibilité et sa popularité dans l'écosystème Node.js ont été des facteurs déterminants dans ce choix.

Express JS offre une approche minimaliste et modulaire pour le développement d'API RESTful, ce qui permet une mise en œuvre efficace des fonctionnalités de synchronisation des données tout en maintenant un code propre et maintenable.

Sa compatibilité avec d'autres modules Node.js et sa large communauté de développeurs ont facilité l'intégration avec notre architecture existante et ont permis un développement rapide du système de scan d'articles.

Autres Technologies Évaluées :

- Hapi.js : Hapi.js est un autre framework populaire pour le développement d'API en Node.js, mais sa courbe d'apprentissage plus raide et son approche plus structurée ont été des obstacles à son adoption dans notre projet, qui nécessitait une mise en œuvre rapide et flexible.
- LoopBack : LoopBack est un framework robuste pour la création d'API RESTful, mais sa complexité et sa lourdeur le rendaient moins adapté à notre cas d'utilisation spécifique, où la simplicité et la légèreté étaient des priorités.



# Guide d'Installation et Configuration

## Prérequis Matériels et Logiciels

Avant de commencer l'installation, il faut s'assurer que les systèmes répondent aux exigences matérielles et logicielles suivantes

Serveur Local :

- Matériel :
  - Un serveur ou un ordinateur disposant d'une capacité de stockage adéquate pour héberger la base de données MySQL et l'API Express JS.
  - Une connexion réseau stable pour permettre la communication avec les tablettes des magasins.
- Logiciel :
  - Un système d'exploitation compatible avec les logiciels requis (par exemple, Linux, Windows).
  - Node.js et npm installés pour exécuter l'API Express JS.
  - MySQL Server installé et configuré pour gérer la base de données.

Tablettes des Magasins :

- Matériel :
  - Tablettes compatibles avec le logiciel front-end React et les exigences de l'application.
- Logiciel :
  - Un navigateur web compatible avec React et les technologies web modernes.

# Installation de l'API Express JS sur le Serveur Local

Pour installer et configurer l'API Express JS sur le serveur local, les étapes à suivre sont les suivantes :

1. Clonez le dépôt Git contenant le code source de l'API Express JS sur le serveur local.
2. Ouvrez une fenêtre de terminal et naviguez jusqu'au répertoire du projet.
3. Installez les dépendances du projet en exécutant la commande `npm install`.
4. Configurez les paramètres de connexion à la base de données dans le fichier de configuration approprié.
5. Démarrez l'API en exécutant la commande `npm start`.
6. Vérifiez que l'API est accessible via l'URL spécifiée dans la configuration, en utilisant un navigateur web ou un outil de test d'API tel que Postman.

## Configuration de la Base de Données MySQL

Procédure pour configurer la base de données MySQL sur le serveur local :

1. Installez MySQL Server sur le serveur local en suivant les instructions fournies par le fournisseur.
2. Créez une nouvelle base de données pour le système de scan d'articles en utilisant une interface utilisateur graphique ou une commande SQL.
3. Créez les tables nécessaires dans la base de données en exécutant les scripts de création fournis avec le code source de l'API Express JS.
4. Configurez les autorisations d'accès à la base de données pour permettre à l'API Express JS d'interagir avec les tables et les données.
5. Vérifiez que la base de données est accessible et fonctionnelle en exécutant des requêtes de test à partir de l'API Express JS ou de tout autre outil de gestion de bases de données.

# Configuration de l'Interface Front-end React sur les Tablettes

Pour configurer l'interface front-end React sur les tablettes des magasins, la démarche à suivre est la suivante :

1. Compilez le code source de l'interface front-end React en utilisant l'outil de compilation approprié (par exemple, Webpack, Parcel).
2. Transférez les fichiers compilés sur les tablettes des magasins via une connexion réseau ou un système de distribution de fichiers approprié.
3. Assurez-vous que les tablettes sont configurées pour exécuter l'interface front-end React en tant qu'application autonome ou dans un navigateur web.
4. Testez l'interface front-end React pour vous assurer qu'elle fonctionne correctement et qu'elle est capable de communiquer avec l'API Express JS sur le serveur local.

# Guide de Développement

Cette section fournit des orientations détaillées pour le développement du système de scan d'articles, en mettant l'accent sur la structure du code source, l'architecture logicielle, les bonnes pratiques de développement et la gestion des dépendances.

## Structure du Code Source

La structure du code source du système de scan d'articles doit être organisée de manière logique et cohérente, en suivant les bonnes pratiques de développement. Voici comment est constitué l'ensemble du code source :

bash

```
scan-darticles/
|
├─ api/                # Code source de l'API Express JS
|   ├─ controllers/    # Contrôleurs de l'API
|   ├─ models/         # Modèles de données de l'API
|   ├─ routes/         # Routes de l'API
|   └─ config/         # Configuration de l'API
|
├─ frontend/          # Code source de l'interface front-end React
|   ├─ components/     # Composants réutilisables
|   ├─ pages/          # Pages de l'application
|   ├─ services/       # Services pour la communication avec l'API
|   └─ styles/         # Feuilles de style CSS
|
└─ database/          # Scripts de création de la base de données MySQL
```

# Architecture Logicielle

L'architecture logicielle du système de scan d'articles suit une approche client-serveur distribuée, où les tablettes des magasins agissent comme des clients et interagissent avec le serveur local via une API. Voici une vue d'ensemble de l'architecture logicielle du système reprenant celle du schéma vu précédemment :

- Front-end (Tablettes) : L'interface utilisateur est développée en utilisant le framework React, offrant une expérience utilisateur interactive et réactive.
- Back-end (Serveur Local) : L'API est développée en utilisant Express JS, un framework JavaScript pour Node.js, pour gérer les requêtes HTTP et interagir avec la base de données MySQL.
- Base de Données : MySQL est utilisé comme système de gestion de base de données relationnelles pour stocker les articles scannés et les données associées.

## Bonnes Pratiques de Développement

Pour garantir la qualité et la maintenabilité du code source, il est recommandé de suivre les bonnes pratiques de développement suivantes :

- Utiliser des conventions de nommage cohérentes pour les variables, les fonctions et les fichiers.
- Commenter le code de manière claire et concise pour faciliter la compréhension et la maintenance.
- Écrire des tests unitaires et d'intégration pour valider le comportement des différentes fonctionnalités du système.
- Utiliser des outils de gestion de version comme Git pour suivre les modifications du code source et faciliter la collaboration entre les développeurs.
- Effectuer une revue de code régulière pour identifier et corriger les éventuels problèmes de qualité ou de performance.

## Gestion des Dépendances

Pour gérer les dépendances du projet, il est recommandé d'utiliser un gestionnaire de paquets tel que npm (Node Package Manager) pour Node.js. Les principales dépendances du projet et leur utilisation sont : React, utilisé pour développer l'interface utilisateur réactive des tablettes ; Express, Framework utilisé pour développer l'API back-end sur le serveur local ; MySQL, système de gestion de base de données relationnelles pour stocker les données du système.

# Intégration et Déploiement

## Intégration des Nouveaux Magasins

L'intégration des nouveaux magasins dans le système de scan d'articles implique les étapes suivantes :

### 1. Configuration du Serveur Local :

- Installez et configurez un nouveau serveur local dans le magasin, en suivant les instructions fournies dans la section 4.
- Assurez-vous que le serveur local est connecté au réseau du magasin et qu'il dispose d'une connexion Internet stable.

### 2. Installation de l'Interface Front-end :

- Déployez l'interface front-end React sur les tablettes du magasin en suivant les instructions fournies dans la section 4.
- Assurez-vous que les tablettes sont configurées pour se connecter au serveur local nouvellement configuré.

### 3. Configuration Initiale :

- Effectuez la configuration initiale de l'application sur les tablettes, en saisissant les informations nécessaires telles que l'identifiant du magasin et les paramètres de connexion au serveur local.

### 4. Vérification :

- Effectuez des tests pour vous assurer que les tablettes peuvent communiquer correctement avec le serveur local et que l'application fonctionne comme prévu.

# Déploiement de l'Application

Le déploiement de l'application sur les tablettes et les serveurs locaux implique les étapes suivante :

## 1.Préparation des Fichiers :

- Compilez le code source de l'interface front-end React pour générer les fichiers statiques à déployer sur les tablettes.
- Assurez-vous que les fichiers de l'API Express JS sont correctement configurés et prêts à être déployés sur le serveur local.

## 2.Transfert des Fichiers :

- Transférez les fichiers compilés de l'interface front-end sur les tablettes du magasin, en utilisant une connexion réseau ou un système de distribution de fichiers approprié.
- Copiez les fichiers de l'API Express JS sur le serveur local du magasin, en suivant les procédures de transfert de fichiers habituelles.

## 3.Configuration Finale :

- Assurez-vous que les fichiers déployés sont configurés correctement pour se connecter au serveur local et fonctionner avec les autres composants du système.

## 4.Tests :

- Effectuez des tests pour vous assurer que l'application est déployée avec succès sur les tablettes et que les tablettes peuvent communiquer correctement avec le serveur local.

# Gestion des Mises à Jour et des Versions

La gestion des mises à jour et des versions du logiciel suivent ce processus :

## 1.Développement des Mises à Jour :

- Identifiez les nouvelles fonctionnalités et les corrections de bugs à inclure dans la prochaine version du logiciel, en fonction des retours des utilisateurs et des besoins commerciaux.

## 2.Tests des Mises à Jour :

- Effectuez des tests approfondis pour valider les nouvelles fonctionnalités et garantir qu'elles ne perturbent pas le fonctionnement existant du système.

## 3.Déploiement des Mises à Jour :

- Déployez les mises à jour du logiciel sur les tablettes et les serveurs locaux en suivant les mêmes procédures que pour le déploiement initial de l'application.

## 4.Communication avec les Utilisateurs :

- Informez les utilisateurs des nouvelles fonctionnalités et des améliorations apportées par la mise à jour, en fournissant des instructions détaillées au besoin.



# Maintenance et Gestion des Erreurs

## Maintenance du Système

La maintenance du système de scan d'articles est essentielle pour assurer son bon fonctionnement à long terme. Voici les principales activités de maintenance à effectuer :

### 1. Sauvegardes Régulières :

- Effectuez des sauvegardes régulières de la base de données MySQL pour protéger les données critiques en cas de panne du système ou de perte de données.

### 2. Surveillance du Serveur Local :

- Surveillez les performances du serveur local en utilisant des outils de surveillance système pour détecter les problèmes potentiels tels que les goulots d'étranglement de performance ou les pannes matérielles.

### 3. Gestion des Mises à Jour :

- Appliquez les mises à jour logicielles et les correctifs de sécurité sur le serveur local et les tablettes pour garantir la sécurité et la stabilité du système.

### 4. Maintenance Préventive :

- Effectuez des opérations de maintenance préventive sur le matériel du serveur local, telles que le nettoyage des ventilateurs et la vérification de l'état des disques durs, pour éviter les pannes matérielles imprévues.

# Gestion des Erreurs et des Exceptions

La gestion des erreurs et des exceptions est cruciale pour assurer une expérience utilisateur fluide et sans problème. Voici les meilleures pratiques pour gérer les erreurs dans le système de scan d'articles :

## 1. Journalisation des Erreurs :

- Mettez en place un système de journalisation robuste pour enregistrer les erreurs et les exceptions rencontrées pendant l'utilisation du système, en incluant des informations détaillées sur la cause de l'erreur et le contexte dans lequel elle s'est produite.

## 2. Notification des Erreurs :

- Configurez des alertes de notification pour les erreurs critiques qui nécessitent une intervention immédiate, telles que les pannes du serveur local ou les échecs de synchronisation des données entre les tablettes.

## 3. Gestion des Exceptions :

- Implémentez des mécanismes de gestion des exceptions dans le code source de l'API Express JS et de l'interface front-end React pour capturer et traiter les erreurs de manière appropriée, en affichant des messages d'erreur informatifs aux utilisateurs et en prenant des mesures correctives si nécessaire.

# Dépannage des Problèmes Techniques

En cas de problèmes techniques rencontrés par les utilisateurs, il est important de pouvoir fournir un support efficace pour résoudre les problèmes rapidement. Nous recommandons le protocole suivant pour le dépannage des problèmes techniques :

## 1. Identification du Problème :

- Écoutez attentivement les utilisateurs pour comprendre la nature du problème qu'ils rencontrent, en posant des questions supplémentaires si nécessaire pour clarifier les symptômes.

## 2. Diagnostic du Problème :

- Utilisez des outils de diagnostic appropriés pour identifier la cause sous-jacente du problème, en examinant les journaux d'erreurs, en effectuant des tests de connectivité réseau et en vérifiant l'état du matériel et des logiciels impliqués.

## 3. Résolution du Problème :

- Prenez les mesures nécessaires pour résoudre le problème, en appliquant des correctifs logiciels, en redémarrant les services affectés ou en remplaçant le matériel défectueux si nécessaire.

## 4. Suivi et Rétroaction :

- Suivez l'évolution du problème pour vous assurer qu'il a été résolu avec succès, en fournissant des mises à jour régulières aux utilisateurs concernés et en recueillant des commentaires sur la qualité de la résolution fournie.

# Sécurité et Confidentialité

## Gestion des Accès et des Autorisations

La gestion des accès et des autorisations est un élément clé de la sécurité du système de scan d'articles. Pour garantir que seules les personnes autorisées ont accès au système et aux données sensibles, nous mettons en place les mesures suivantes :

- **Authentification** : Tous les utilisateurs du système doivent être authentifiés avant d'accéder aux fonctionnalités critiques. Cela peut être réalisé en utilisant des méthodes telles que l'authentification par mot de passe ou l'authentification à deux facteurs pour renforcer la sécurité des comptes utilisateur.
- **Autorisations basées sur les rôles** : Nous utilisons un modèle d'autorisations basé sur les rôles pour définir les privilèges d'accès des utilisateurs en fonction de leurs responsabilités et de leurs fonctions au sein de l'organisation. Cela garantit que chaque utilisateur a uniquement accès aux fonctionnalités et aux données nécessaires à l'exécution de ses tâches.

o

## Protection des Données Sensibles

La protection des données sensibles est une priorité absolue pour assurer la confidentialité et l'intégrité des informations stockées dans le système de scan d'articles. Voici les mesures que nous prenons pour protéger les données sensibles :

- **Chiffrement des données** : Toutes les données sensibles sont chiffrées lorsqu'elles sont stockées dans la base de données MySQL et lorsqu'elles sont transférées entre les tablettes et le serveur local. Cela garantit que même en cas de violation de la sécurité, les données restent inaccessibles aux personnes non autorisées.
- **Contrôles d'accès** : Nous utilisons des contrôles d'accès granulaires pour limiter l'accès aux données sensibles uniquement aux utilisateurs autorisés. Cela peut inclure des restrictions d'accès basées sur les rôles, les attributs de sécurité ou les règles de confidentialité définies par l'organisation.

# Surveillance de la Sécurité

La surveillance de la sécurité est essentielle pour détecter et répondre rapidement aux menaces potentielles contre le système de scan d'articles. Voici comment nous assurons une surveillance efficace de la sécurité :

- **Journalisation des événements** : Nous mettons en place un système de journalisation des événements pour enregistrer les activités du système, y compris les tentatives d'accès non autorisées, les erreurs système et les changements de configuration. Cela nous permet de suivre l'activité du système et d'identifier les comportements suspects.
- **Analyse des journaux** : Nous utilisons des outils d'analyse des journaux pour examiner régulièrement les journaux d'événements à la recherche de comportements anormaux ou de signes de compromission de la sécurité. En détectant rapidement les incidents de sécurité potentiels, nous pouvons prendre des mesures correctives pour minimiser les dommages et renforcer la sécurité du système.

## Formation et Sensibilisation à la Sécurité

Nous accordons une grande importance à la formation et à la sensibilisation à la sécurité pour tous les utilisateurs du système de scan d'articles. Nous proposons des programmes de formation réguliers pour sensibiliser les utilisateurs aux meilleures pratiques de sécurité, aux risques potentiels et aux mesures qu'ils peuvent prendre pour protéger les données et prévenir les incidents de sécurité.