**Cahier des charges**

**IVHM**

Développement d’une application de gestion de stocks

Conçu par:

**Cadu** et **Rasko**

**Sommaire**

**Introduction :**

contexte

**Objectifs du projet :**

Usages  
  
**Fonctionnalités :**

CRUD  
  
**Choix Technologiques:**

back et front  
  
**Conception de la BDD :**

les tables et relations

**Interface Front-end:**

Affichage en fonction des rôles

**Introduction**

Dans le cadre de notre projet de développement d'une application de gestion de stocks, nous avons conçu une solution fullstack moderne et performante, visant à répondre aux besoins des entreprises en matière de gestion et de suivi des inventaires.

Cette application est développée avec Angular pour le frontend, Node.js avec Express pour le backend, et utilise Prisma comme ORM (Object-Relational Mapping) pour une gestion optimisée des données.

L'objectif principal de ce projet est de démontrer nos compétences techniques et notre capacité à concevoir des solutions robustes et évolutives.

**Objectifs du projet**

La gestion des stocks est un enjeu critique pour de nombreuses entreprises, quelle que soit leur taille.

Une gestion efficace permet de réduire les coûts, d'optimiser les processus logistiques et d'améliorer la satisfaction client.

Notre application vise à fournir une solution intuitive et complète pour:

1. Centraliser la gestion des stocks : Permettre aux utilisateurs de suivre en temps réel les entrées, sorties et niveaux de stock.
2. Automatiser les tâches répétitives : Simplifier les opérations de mise à jour et de suivi des produits
3. Améliorer la prise de décision : Fournir des rapports et des analyses pour aider les gestionnaires à anticiper les besoins
4. Faciliter la collaboration : Offrir une interface conviviale et des fonctionnalités de partage pour les équipes.

**Choix Technologiques**

Nous avons opté pour une architecture full stack moderne et modulaire, en utilisant des technologies largement adoptées dans l'industrie:

* Front End (Angular) : Pour une interface utilisateur réactive, dynamique et facile à maintenir
* Back End (Node.js + Express) : Pour une API RESTful performante et scalable
* ORM (Prisma) : Pour simplifier l'accès et la gestion des données, tout en garantissant une base de code propre et type-safe
* Base de Données : Une base de données relationnelle (ex : PostgreSQL) pour stocker les données de manière structurée et sécurisée
* Déploiement : Utilisation de Docker pour une mise en production facile et reproductible.

**Technologies**

**BackEnd:**

1. BDD : SQL
2. BDD TECHNO: Mysql
3. Framework: expressJs
4. ORM: Prisma
5. Outil de visualisation BDD: Dbeaver

**FrontEnd:**

1. Mobile: React-native
2. Desktop: Angular

**DevOps:**

1. Conteneurisation: Docker
2. Déploiement: Kubernetes

**Conception de la BDD**

#### **Rôles et permissions**

#### **Introduction**

Dans ce système de gestion de stocks, les accès et actions disponibles sont définis en fonction du rôle attribué à chaque utilisateur. Cette approche permet d’assurer une meilleure sécurité, d'éviter les erreurs humaines et d’optimiser la gestion des stocks.

##### **3.2 Définition des rôles**

📌 **Administrateur**  
 L'administrateur dispose des droits les plus étendus dans le système. Il est responsable de la gestion des utilisateurs et des paramètres généraux de l’application. Ses principales actions incluent :

* Ajouter, modifier ou supprimer des utilisateurs
* Gérer les permissions et rôles
* Accéder à l’ensemble des fonctionnalités sans restriction
* Modifier les paramètres globaux du système

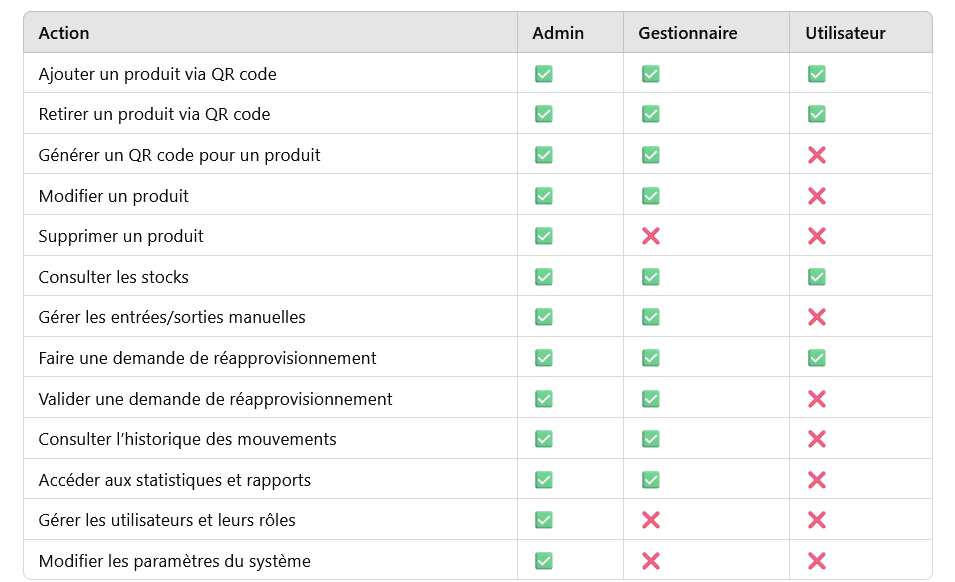
📌 **Gestionnaire**  
 Le gestionnaire est responsable des opérations quotidiennes liées aux stocks. Son rôle principal est d’assurer un suivi des produits et d’éviter les ruptures de stock. Ses principales actions incluent :

* Ajouter, modifier ou supprimer des produits
* Gérer les entrées et sorties de stock
* Consulter et exporter les rapports d’inventaire
* Valider ou refuser les demandes de réapprovisionnement

📌 **Utilisateur**  
 L’utilisateur a un accès limité au système. Son rôle est principalement consultatif, avec la possibilité de demander des produits si nécessaire. Ses principales actions incluent :

* Consulter les stocks disponibles
* Effectuer des demandes de réapprovisionnement
* Suivre le statut de ses demandes

**3.3 Tableau des permissions**

****

**Interface utilisateur**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Page : Inscription**

**Fonctionnalités :**

* **Ajouter un utilisateur : Formulaire pour créer un utilisateur.**
* **Lien vers page connexion**

**( Nom | Email | Mot de passe | Actions )**

**Visibilité : Administrateur, Gestionnaire, Utilisateur**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Page : Connexion**

**Fonctionnalités :**

* **Connecter un utilisateur : Formulaire pour connecter un utilisateur.**
* **Lien vers page inscription**

**( Email | Mot de passe | Actions )**

**Visibilité : Administrateur, Gestionnaire, Utilisateur**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Page : Gestion des Utilisateurs**

**Fonctionnalités :**

* **Ajouter un utilisateur : Formulaire pour créer un utilisateur (Nom, Email, Rôle).**
* **Modifier un utilisateur : Permet de changer le rôle ou l'email d'un utilisateur.**
* **Supprimer un utilisateur : Supprimer un utilisateur du système.**
* **Vue Liste des utilisateurs : Nom, Email, Rôle, Actions.**

**( Nom | Email | Mot de passe | Rôle | Actions )**

**Visibilité : Administrateur \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### **Page : Gestion des produit**

**Fonctionnalités :**

* **Ajouter un produit : Formulaire pour ajouter un produit (Nom, Quantité, Prix, etc.).**
* **Modifier un produit : Modifier les détails d'un produit existant.**
* **Supprimer un produit : Retirer un produit du système.**
* **Vue liste de stocks : Nom de l’article, quantité, date d’arrivée, date d’expédition**

**(Nom du produit |Quantité en stock |Quantité de produit sélectionné |Actions )**

**Visibilité : Administrateur, Gestionnaire**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### **Page : Rapports d'Inventaire**

**Fonctionnalités :**

* **Vue des rapports d'inventaire : Voir les rapports détaillés des produits en stock, les entrées et sorties.**
* **Exporter les rapports : Permet de télécharger les rapports sous forme de fichier (PDF, Excel).**

**( Nom de l’Inventaire | date de création | format de document | Actions )**

**Visibilité : Administrateur, Gestionnaire**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### **Page : Demandes de Réapprovisionnement**

**Fonctionnalités :**

**Valider ou refuser les demandes de réapprovisionnement : Permet au gestionnaire d'accepter ou refuser une demande.**

**( Produit | Quantité demandée | Demande par | Actions )**

**Visibilité : Administrateur, Gestionnaire**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### **Page : Consultation des Stocks**

**Fonctionnalités :**

* **Vue des produits : nom des produits et leur quantité en stock.**
* **Demander un réapprovisionnement**
* **Suivre le statut : Voir si la demande a été validée ou refusée.**

**( Nom du produit | Quantité en stock | Quantité de réapprovisionnement demandé | Statut | Actions )**

**Visibilité : Administrateur, Gestionnaire, Utilisateur**

**Read.me**

**Variable d’environnement**

**node -v : v20.15.0**

**npm -v : 10.9.2**

**commande d’installation**

**Angular CLI installation : npm install -g @angular/cli**

**nouveau projet Angular création : ng new nom-du-projet**

**✔ Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS ]**

**✔ Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? Yes**

**✔ Would you like to use the Server Routing and App Engine APIs (Developer Preview) for this server application? Yes**

**se déplacer sur le dossier du projet avant !**

**démarrer l'application et la voir dans votre navigateur : ng server**

#### **Créer un composant : ng generate component nom-du-composant**

#### 

#### 