

# Cahier des charges

## IVHM

Développement d'une application de gestion de stocks

Développé et conçu par :  
Muttalip et Max

## Sommaire

Introduction :

contexte

Objectifs du projet :

Usages

Fonctionnalités :

CRUD

Choix Technologiques

back et front

Conception de la BDD :

les tables et relations

Interface utilisateur :

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 :

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

## Choix Technologiques

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

- Frontend (Angular) : Pour une interface utilisateur réactive, dynamique et facile à maintenir.
- Backend (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

- BDD : SQL
- BDD TECHNO : MySQL
- Framework : ExpressJs
- ORM : Prisma
- Outil de visualisation bdd : DBeaver

Frontend

- mobile : ReactNative
- desktop : Angular

DevOps

- Conteneurisation : Docker
- Déploiement : Kubernetes

## Conception de la BDD

Roles

Admin	
Gestionnaires	
Utilisateur	

## Conception de la BDD

Entités

Produits	
Stock	
Emplacements	
Catégories	
Fournisseurs	
Mouvements_Stock	
Commandes_Fournisseurs	
Détails_Commande_Fournisseur	
Utilisateurs	
Inventaires	
Détails_Inventaire	id_detail (clé primaire) id_inventaire (clé étrangère) id_produit (clé étrangère) quantite_systeme quantite_comptee ecart