

# RAPPORT DE PROJET

## Gestion des Stocks et des Commandes - Supermarché



Encadrant : M. BADDI Youssef

Réaliser par :

CHENNOUFI Imane

MEFTAH Nouhaila

LAFKIH Hiba

2025-2026

# TABLE DES MATIÈRES

## Table des matières

<b>TABLE DES MATIÈRES</b> .....	1
<b>1. INTRODUCTION</b> .....	2
<b>2. OBJECTIFS DU PROJET</b> .....	2
<b>3. ARCHITECTURE GÉNÉRALE</b> .....	2
<b>4. DESCRIPTION FONCTIONNELLE</b> .....	5
<b>4.1 Fonctionnalités Principales</b> .....	5
<b>4.2 Flux d'Utilisation</b> .....	5
<b>5. COMPOSANTS TECHNIQUES</b> .....	6
<b>5.1 Technologies Utilisées</b> .....	6
<b>5.2 Structure des Packages</b> .....	6
<b>6. FORMAT DES DONNÉES</b> .....	8
<b>6.1 Structure des Fichiers CSV</b> .....	8
<b>6.2 Conventions de Format</b> .....	9
<b>7. GUIDE D'UTILISATION</b> .....	10
<b>7.1 Prérequis</b> .....	10
<b>7.2 Lancement de l'Application</b> .....	10
<b>7.3 Menu Principal</b> .....	10
<b>7.4 Exemples d'Utilisation</b> .....	11
<b>8. CONCLUSION</b> .....	12
<b>Points Forts</b> .....	13
<b>Possibilités d'Amélioration Futures</b> .....	13

# 1. INTRODUCTION

Le projet **Supermarché CSV** est une application Java développée dans le cadre d'une évaluation académique. Il s'agit d'un système complet de gestion des stocks et des commandes pour un supermarché, permettant aux utilisateurs de gérer les produits, les clients et les commandes via une interface en ligne de commande (CLI).

L'application utilise des fichiers CSV pour la persistance des données, assurant ainsi une gestion simplifiée et portable des informations sans nécessiter de base de données complexe.

## 2. OBJECTIFS DU PROJET

Les objectifs principaux de ce projet sont :

- **Gestion des produits** : Ajouter, modifier et consulter les produits du supermarché avec gestion du stock
- **Gestion des clients** : Maintenir une base de clients avec leurs informations de contact
- **Gestion des commandes** : Créer et consulter les commandes passées par les clients
- **Alerte de stock** : Signaler automatiquement les produits dont le stock est faible
- **Persistance des données** : Sauvegarder toutes les modifications dans des fichiers CSV

Ces objectifs contribuent à créer un système fonctionnel et pratique pour la gestion opérationnelle d'un petit supermarché.

## 3. ARCHITECTURE GÉNÉRALE

L'application suit une architecture en couches respectant les bonnes pratiques de développement orienté objet :

## SUPERMARCHE-CSV/

- |— 📁 data/ # Fichiers de données sources (CSV)
  - | |— customers.csv
  - | |— order\_items.csv
  - | |— orders.csv
  - | |— products.csv
- |— 📁 lib/ # Bibliothèques externes (JARs)
  - | |— commons-lang3-3.2.jar
  - | |— commons-text-1.14.jar
  - | |— junit-platform-console.jar
  - | |— opencsv-5.12.0.jar
- |— 📁 src/ # Code source Java
  - | |— 📁 app/
    - | | |— Main.java # Point d'entrée de l'application
  - | |— 📁 csv/ # Data Access Objects (DAO) pour le CSV
    - | | |— CustomerCsvDao.java
    - | | |— OrderCsvDao.java
    - | | |— ProductCsvDao.java
  - | |— 📁 model/ # Classes de données (POJOs)
    - | | |— Customer.java
    - | | |— Order.java

```
| | └─ OrderItem.java
| | └─ Product.java
| └─ 📁 service/          # Logique métier
|   └─ InventoryService.java
|   └─ OrderService.java
└─ 📄 RAPPORT_PROJET.pdf    # Documentation du projet
└─ 📄 README.md             # Instructions et présentation
```

Cette architecture en couches assure une séparation claire des responsabilités et facilite la maintenance et l'évolution du système.

## 4. DESCRIPTION FONCTIONNELLE

### 4.1 Fonctionnalités Principales

#### *Gestion des Produits*

- Afficher la liste complète des produits disponibles
- Ajouter un nouveau produit au catalogue
- Modifier le stock d'un produit existant
- Consulter les détails d'un produit (prix, catégorie, stock)
- Alertes automatiques pour les produits en rupture de stock

#### *Gestion des Clients*

- Afficher la liste de tous les clients
- Ajouter un nouveau client au système
- Consulter les informations d'un client (nom, email, téléphone)

#### *Gestion des Commandes*

- Créer une nouvelle commande pour un client
- Ajouter des articles à une commande
- Afficher l'historique des commandes
- Consulter les détails d'une commande (articles, quantités, total)

#### *Persistance des Données*

- Sauvegarde automatique de tous les changements dans les fichiers CSV
- Chargement des données au démarrage de l'application
- Restauration du dernier état sauvegardé

### 4.2 Flux d'Utilisation

1. **Démarrage** : L'application charge les données des fichiers CSV
2. **Menu Principal** : L'utilisateur accède à différentes options via un menu textuel

3. **Opérations** : L'utilisateur peut effectuer des opérations sur les produits, clients et commandes
4. **Sauvegarde** : Les modifications sont persistées dans les fichiers CSV
5. **Quitter** : L'application se ferme en conservant toutes les données

## 5. COMPOSANTS TECHNIQUES

### 5.1 Technologies Utilisées

Élément	Détail
Langage	Java
Bibliothèque CSV	OpenCSV
Type d'Application	Interface en Ligne de Commande (CLI)
Système de Fichiers	Fichiers CSV pour la persistance
JDK Minimum	Java 8+

### 5.2 Structure des Packages

#### **Package app**

- **Main.java** : Point d'entrée de l'application
  - Initialise les DAOs et services
  - Gère le menu principal en boucle
  - Traite les entrées utilisateur
  - Coordonne les opérations entre services

#### **Package model**

Contient les classes métier représentant les entités du domaine :

- **Product.java** : Représente un produit avec :
  - ID, Nom, Catégorie, Prix
  - Stock courant et stock minimum
- **Customer.java** : Représente un client avec :
  - ID, Nom, Email, Téléphone
- **Order.java** : Représente une commande avec :
  - ID, Client, Date de commande, Montant total
- **OrderItem.java** : Représente une ligne de commande avec :
  - ID de commande, ID de produit, Quantité, Prix unitaire

### **Package csv**

Contient les DAOs (Data Access Objects) pour la persistance en CSV :

- **ProductCsvDao.java** : Gère la lecture/écriture des produits
- **CustomerCsvDao.java** : Gère la lecture/écriture des clients
- **OrderCsvDao.java** : Gère la lecture/écriture des commandes et articles

Utilise OpenCSV pour la sérialisation/désérialisation automatique.

### **Package service**

Contient la logique métier de l'application :

- **InventoryService.java** : Gère l'inventaire des produits
  - Ajouter/modifier des produits
  - Vérifier la disponibilité des stocks
  - Identifier les produits en alerte de stock faible
- **OrderService.java** : Gère les commandes
  - Créer des commandes
  - Ajouter des articles aux commandes
  - Calculer les montants totaux
  - Vérifier la disponibilité des stocks avant achat



## 6. FORMAT DES DONNÉES

### 6.1 Structure des Fichiers CSV

Les données sont stockées dans quatre fichiers CSV avec les structures suivantes :

#### **`data/products.csv`**

id,name,category,price,stock,minStock

1,Lait,Laiterie,2.50,15,5

2,Pain,Boulangerie,1.20,8,3

3,Pommes,Fruits,0.80,50,10

- **id** : Identifiant unique du produit
- **name** : Nom du produit
- **category** : Catégorie (Laiterie, Fruits, Légumes, etc.)
- **price** : Prix unitaire en euros (avec point décimal)
- **stock** : Quantité actuelle en stock
- **minStock** : Seuil minimum pour alerte de stock faible

#### **`data/customers.csv`**

id,name,email,phone

C001,Jean [Dupont,jean.dupont@mail.com](mailto:jean.dupont@mail.com),06 12 34 56 78

C002,Marie [Martin,marie.martin@mail.com](mailto:marie.martin@mail.com),06 98 76 54 32

- **id** : Identifiant unique du client
- **name** : Nom complet du client
- **email** : Adresse email
- **phone** : Numéro de téléphone

### `data/orders.csv`

id,customerId,date,total

O001,C001,2025-01-15,45.50

O002,C002,2025-01-16,78.90

- **id** : Identifiant unique de la commande
- **customerId** : Référence au client ayant passé la commande
- **date** : Date de la commande (format YYYY-MM-DD)
- **total** : Montant total de la commande en euros

### `data/order_items.csv`

orderId,productId,quantity,unitPrice

O001,1,2,2.50

O001,3,5,0.80

O002,2,10,1.20

- **orderId** : Identifiant de la commande
- **productId** : Identifiant du produit commandé
- **quantity** : Quantité commandée
- **unitPrice** : Prix unitaire au moment de la commande

## 6.2 Conventions de Format

- Tous les nombres décimaux utilisent le **point (.)** comme séparateur
- Les dates utilisent le format **ISO 8601** (YYYY-MM-DD)
- Les identifiants doivent être uniques dans leur catégorie
- Les en-têtes (première ligne) doivent être présents et respecter la casse

## 7. GUIDE D'UTILISATION

### 7.1 Prérequis

- Java JDK 8 ou supérieur installé et configuré
- Fichiers CSV dans le dossier data/
- Bibliothèque OpenCSV incluse dans le classpath

### 7.2 Lancement de l'Application

# Compilation

```
javac -cp lib/opencsv-*.jar src/app/Main.java src/csv/*.java src/model/*.java  
src/service/*.java
```

# Exécution

```
java -cp lib:src app.Main
```

### 7.3 Menu Principal

L'application présente un menu interactif avec les options suivantes :

=== Supermarché ===

1. Lister produits
2. Ajouter produit
3. Modifier stock
4. Lister clients
5. Ajouter client
6. Lister commandes
7. Créer commande
8. Afficher alertes de stock
9. Sauvegarder et Quitter

## 7.4 Exemples d'Utilisation

### ***Ajouter un Produit***

1. Sélectionner l'option 2
2. Entrer l'ID, le nom, la catégorie, le prix et le stock initial
3. Le produit est ajouté à la liste

### ***Créer une Commande***

1. Sélectionner l'option 7
2. Entrer l'ID du client
3. Ajouter les produits et quantités souhaitées
4. La commande est enregistrée avec le montant total calculé

### ***Consulter les Alertes***

1. Sélectionner l'option 8
2. Voir la liste des produits dont le stock est inférieur au seuil minimum
3. Possibilité de réapprovisionner les stocks

## 8. CONCLUSION

Le projet **Supermarché CSV** démontre une implémentation complète et fonctionnelle d'un système de gestion commerciale. Grâce à son architecture modulaire et son utilisation de fichiers CSV pour la persistance, l'application offre une solution simple mais efficace pour gérer les opérations essentielles d'un petit supermarché.

## Points Forts

- ✓ Architecture en couches bien organisée
- ✓ Séparation claire des responsabilités
- ✓ Interface utilisateur intuitive
- ✓ Gestion robuste des données CSV
- ✓ Fonctionnalités complètes de gestion de stock
- ✓ Alertes automatiques de stock faible

## Possibilités d'Amélioration Futures

- Migration vers une base de données relationnelle (MySQL, PostgreSQL)
- Interface graphique (GUI) pour améliorer l'expérience utilisateur
- Système de rapports et statistiques de ventes
- Gestion des fournisseurs et des approvisionnements
- Intégration d'un système de facturation
- Authentification et contrôle d'accès par rôle