

Nom

Anass Chiba

URL Github

<https://github.com/Anass1707/LOG430-labos>

1. Introduction et objectifs

Ce projet est une application web de gestion logistique pour une chaîne de magasins. Il permet la gestion centralisée des stocks, le suivi des ventes, le réapprovisionnement, la génération de rapports et la visualisation des performances via un tableau de bord.

Objectifs principaux :

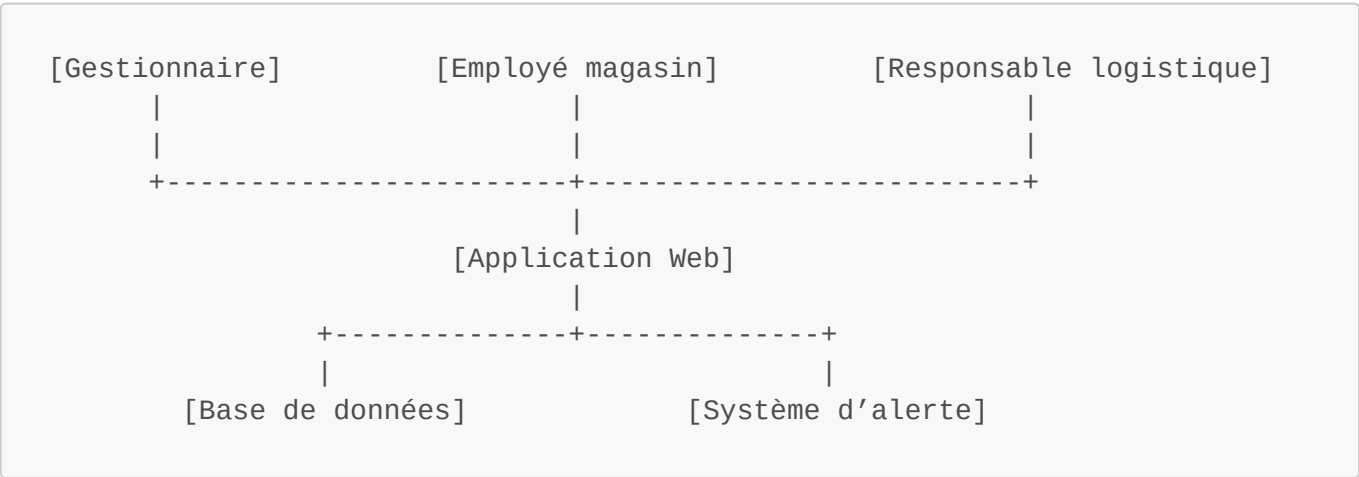
- Optimiser la gestion des stocks entre le centre logistique et les magasins.
- Faciliter le suivi des ventes.
- Offrir une interface web simple pour les gestionnaires et employés.

2. Contraintes

- Utilisation de Java, Spring Boot, Thymeleaf, Hibernate/JPA, PostgreSQL
- Interface web accessible sur navigateur web
- Synchronisation automatique des données entre le centre et les magasins

3. Contexte

Diagramme de contexte (simplifié) :



4. Solution conceptuelle

- **Gestion des stocks** : Stock central et stocks magasins synchronisés, gestion des ruptures et surstocks.
 - **Réapprovisionnement** : Demandes initiées par les magasins, validées par le centre logistique.
 - **Rapports** : Génération de rapports consolidés sur les ventes, stocks, produits les plus vendus.
 - **Tableau de bord** : Indicateurs clés (alertes, tendances) accessibles aux gestionnaires.
 - **Alertes automatiques** : Notification en cas de seuil critique de stock.
-

5. Scénarios d'utilisation

- Générer un rapport des ventes
 - Consulter le stock central
 - Déclencher un réapprovisionnement
 - Visualiser les performances des magasins dans un tableau de bord
 - Mettre à jour les produits depuis la maison mère
 - Approvisionner un magasin depuis le centre logistique
-

6. Architecture logicielle

- **Contrôleurs Spring MVC** : gestion des routes et des vues
 - **Services** : logique métier
 - **Entités JPA** : Magasin, Produit, StockCentral, StockMagasin, Vente, DemandeReapprovisionnement
 - **Repositories** : accès aux données
 - **Vues Thymeleaf** : pages web pour chaque fonctionnalité
-

7. Déploiement

- Application packagée en .jar
 - Déploiement sur serveur Linux avec Java 21+, PostgreSQL
 - Accès via navigateur web
-

8. Qualité et sécurité

- Gestion des erreurs et des exceptions
 - Sauvegarde régulière de la base de données
-

9. Glossaire

- **Stock central** : stock principal au centre logistique
 - **Stock magasin** : stock local à chaque magasin
 - **Réapprovisionnement** : processus de demande et de transfert de stock
 - **Gestionnaire** : utilisateur ayant accès aux rapports et au tableau de bord
 - **Responsable logistique** : valide les demandes de réapprovisionnement
-

10. Annexes

- Diagrammes UML (cas d'utilisation, séquence, classes)
- Scripts SQL d'initialisation

Diagrammes

Diagramme de classes

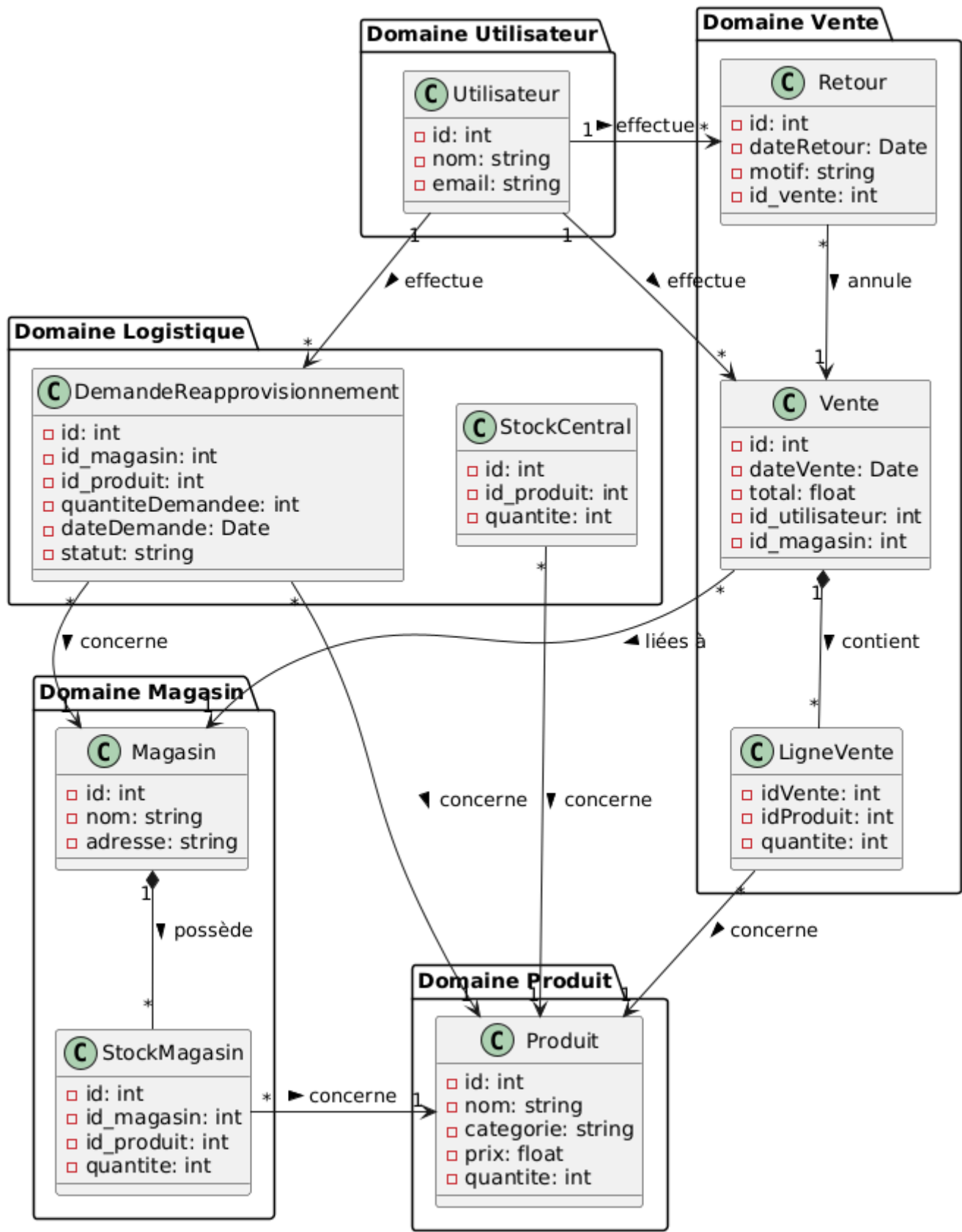


Diagramme de composants

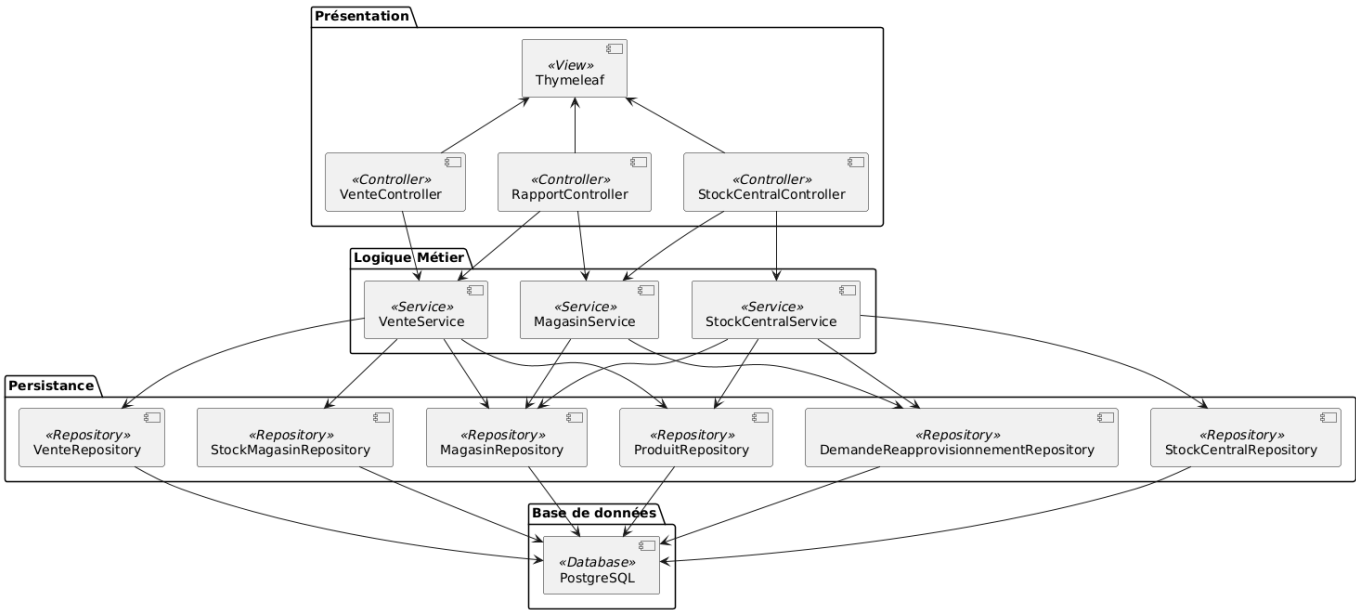


Diagramme de séquence : Création d'une demande

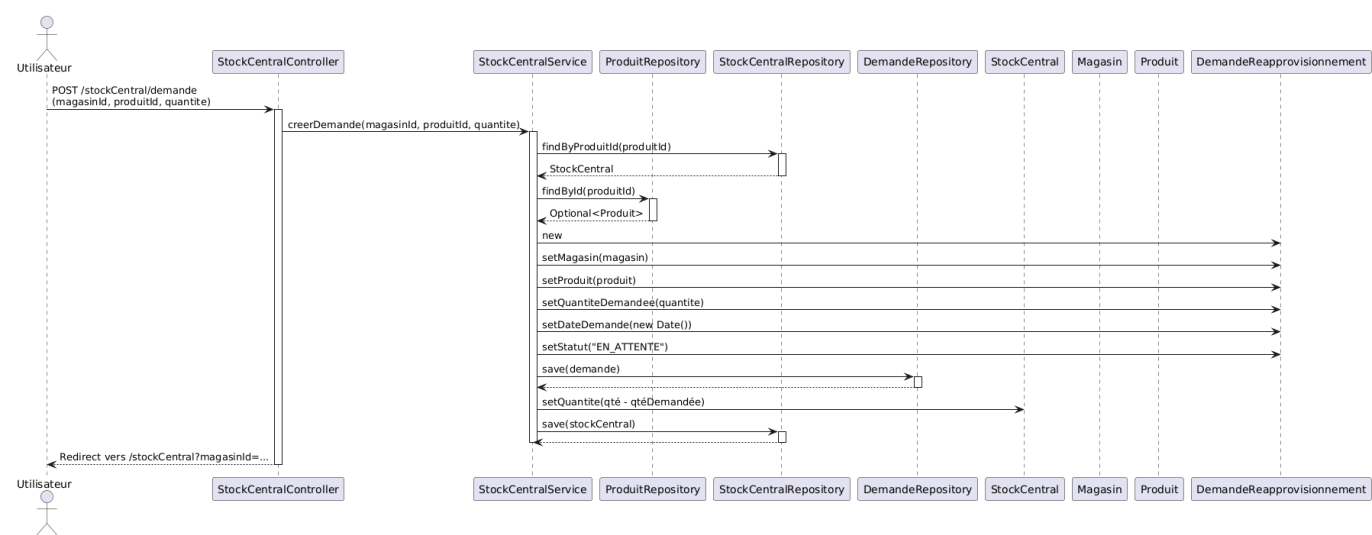


Diagramme de séquence : Rapport détaillé

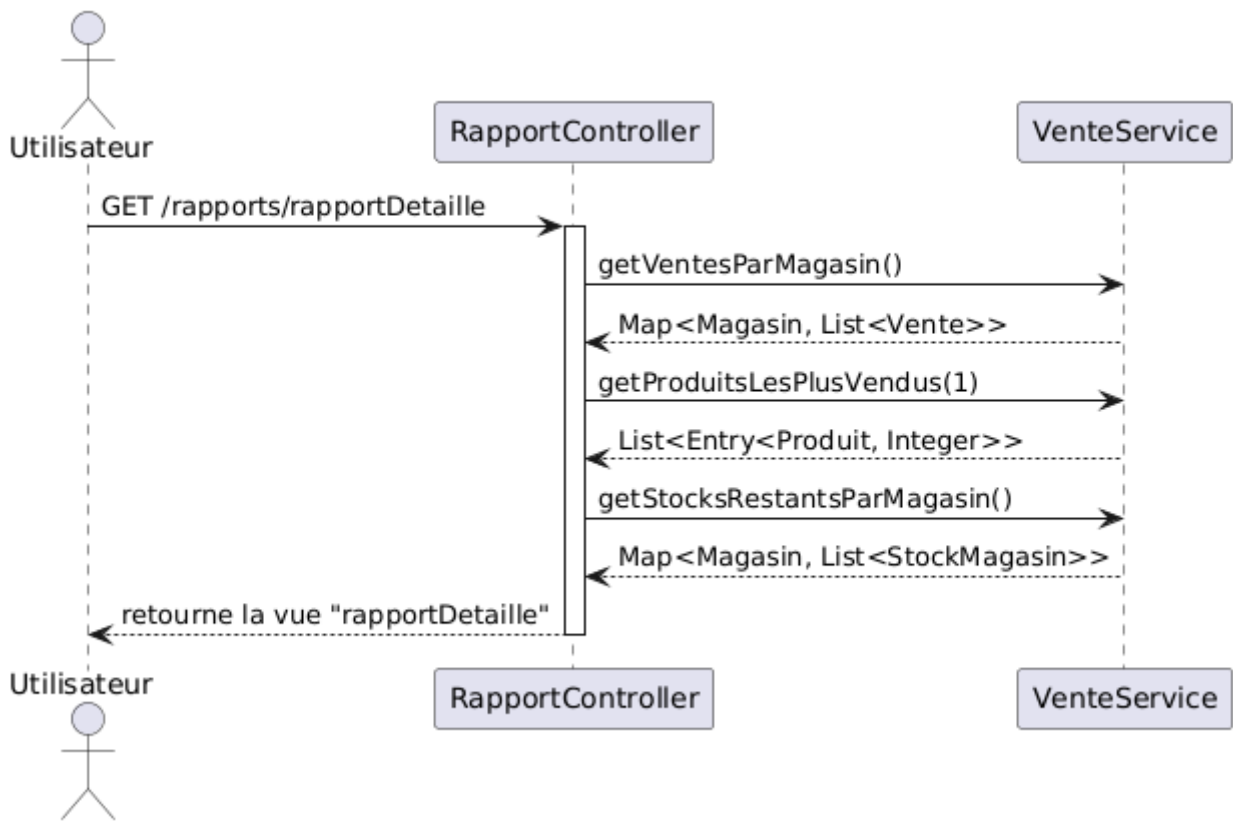


Diagramme de séquence : Consulter Stock Central

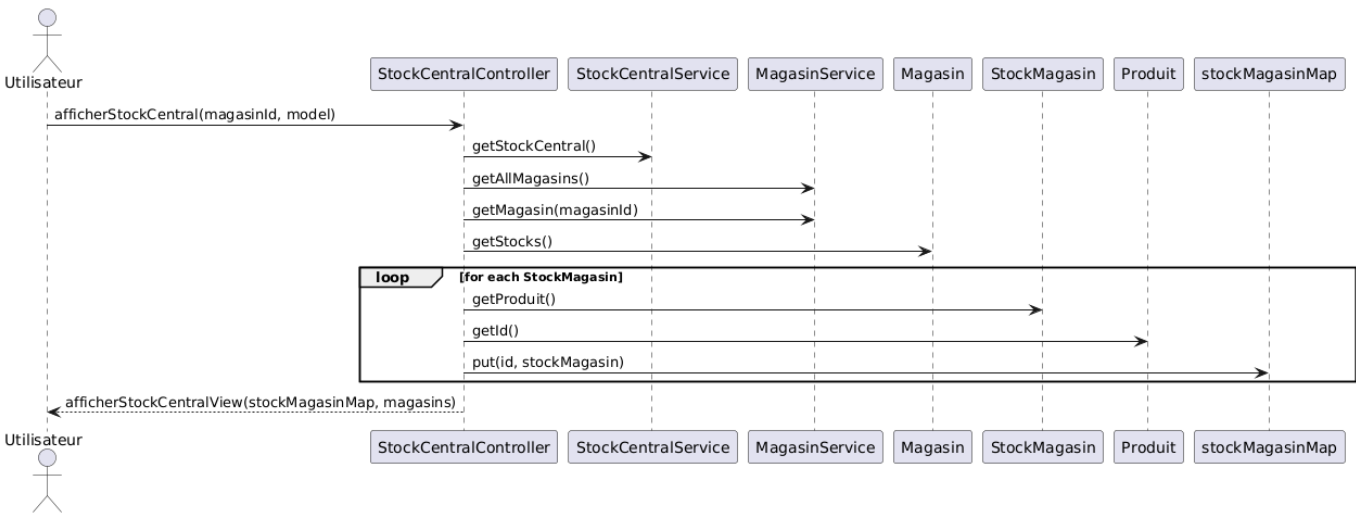


Diagramme de séquence : Visualiser Tableau Bord

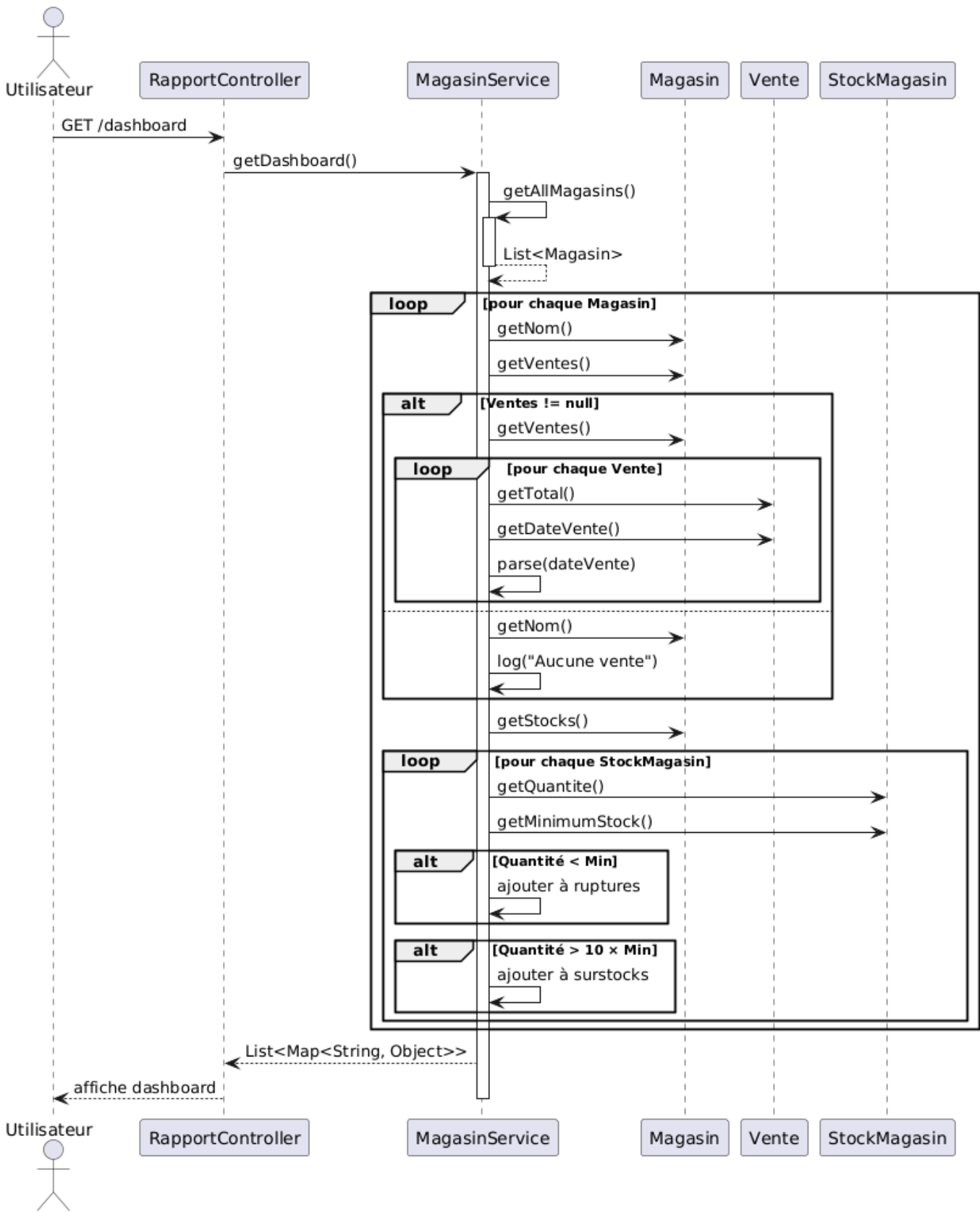


Diagramme de déploiement

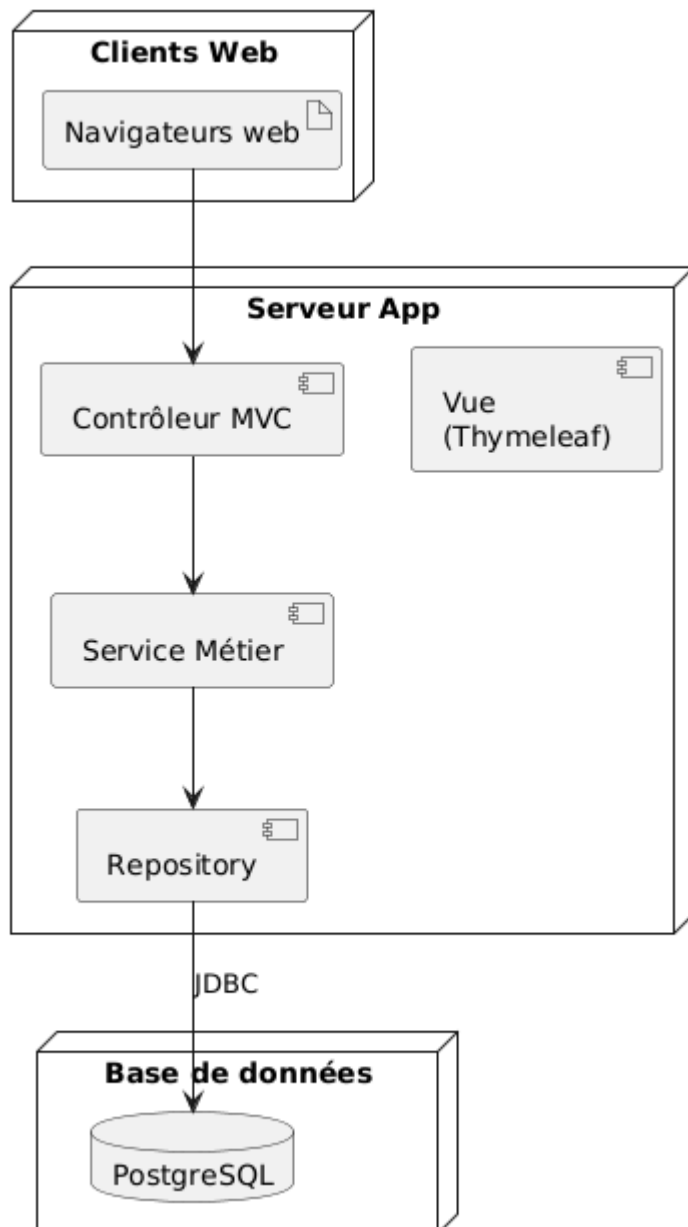


Diagramme de cas d'utilisation

