Sommaire

[Introduction 2](#_Toc159167029)

[Choix de l’Architecture 3](#_Toc159167030)

[1. Couche de Présentation : Interface Utilisateur 3](#_Toc159167031)

[2. Couche Métier : Logique Complexe de la Gestion de Production 3](#_Toc159167032)

[3. Couche de Données : Stockage Centralisé et Gestion des Informations 3](#_Toc159167033)

[Vue de cas d’utilisation 4](#_Toc159167034)

[Système étudié : Gestion de médiathèque. 4](#_Toc159167035)

[Diagramme de cas d’utilisation 5](#_Toc159167036)

[Diagramme de séquence 6](#_Toc159167037)

[Diagramme de classe 7](#_Toc159167038)

[Diagramme d’activité 8](#_Toc159167039)

[Diagramme d’état transition 9](#_Toc159167040)

[Diagramme de déploiement 10](#_Toc159167041)

[Diagramme de composant 11](#_Toc159167042)

[Diagramme de classe participant 12](#_Toc159167043)

# Introduction

Le présent rapport se concentre sur le développement d'un système de Gestion de la Production de Produits Industriels, avec une emphase particulière sur la célèbre marque de boissons gazeuses, Coca-Cola. Ce projet vise à concevoir et à mettre en œuvre un système informatisé qui optimisera la production et la gestion des produits de l'entreprise, en utilisant la méthode UML (Unified Modeling Language) comme principale approche de modélisation.

Coca-Cola est une entreprise emblématique dans l'industrie des boissons, reconnue à l'échelle mondiale pour ses produits innovants et sa forte présence sur le marché. La gestion efficace de sa production est essentielle pour maintenir sa position concurrentielle et répondre à la demande croissante des consommateurs.

Le choix de la méthode UML comme cadre de développement découle de sa capacité à fournir une représentation visuelle claire et précise des différentes composantes du système. En utilisant UML, nous pouvons modéliser les processus de production, les interactions entre les différents acteurs du système, ainsi que les différentes entités et leurs relations au sein de l'environnement de production de Coca-Cola.

Ce rapport sera structuré de manière à présenter une analyse détaillée des besoins du système, suivie par la conception UML du système de gestion de la production de Coca-Cola, incluant les diagrammes de cas d'utilisation, de séquence, de classes, et d'état, entre autres.

L'objectif ultime de ce projet est de fournir à Coca-Cola un outil de gestion efficace qui améliorera la productivité, réduira les coûts de production et assurera une meilleure qualité de ses produits. En suivant une approche méthodique basée sur UML, nous visons à garantir la cohérence et la compréhension commune du système à tous les niveaux de développement et d'utilisation.

Dans les sections suivantes de ce rapport, nous explorerons en détail les différentes phases du projet, en mettant l'accent sur la méthodologie utilisée, les décisions de conception prises et les résultats attendus pour le système de Gestion de la Production de Produits Industriels de Coca-Cola

# Choix de l’Architecture

Dans le cadre de notre projet axé sur la gestion de la production de produits industriels, il est impératif de sélectionner une architecture logicielle qui favorise l'efficacité, la maintenance simplifiée et l'évolutivité. Parmi les différentes options, l'Architecture Three-Tier se démarque comme un choix judicieux en offrant une structure organisée et une séparation claire des responsabilités entre les différentes couches.

## 1. Couche de Présentation : Interface Utilisateur

La première couche de notre architecture est dédiée à l'interface utilisateur, où l'utilisateur interagit avec le système pour surveiller et gérer la production. Cette couche offre une expérience intuitive, permettant aux utilisateurs de visualiser facilement les données pertinentes et de prendre des décisions informées sur la base des informations en temps réel.

## 2. Couche Métier : Logique Complexe de la Gestion de Production

La couche métier constitue le cœur du système, abritant la logique métier complexe liée à la gestion de la production. Elle prend en charge la coordination entre la production de bouchons, de bouteilles et du liquide Coca-Cola. Cette séparation de la logique métier garantit une gestion efficace des processus de production tout en facilitant la maintenance et les mises à jour sans perturber l'ensemble du système.

## 3. Couche de Données : Stockage Centralisé et Gestion des Informations

La troisième couche est dédiée à la gestion des données, où les informations relatives à la production, telles que les niveaux de stock, les rapports de production et les données de qualité, sont stockées de manière centralisée. Cette structure de données centralisée assure la cohérence des informations et simplifie la gestion des bases de données, facilitant ainsi la maintenance et la récupération rapide des données nécessaires.

# Vue de cas d’utilisation

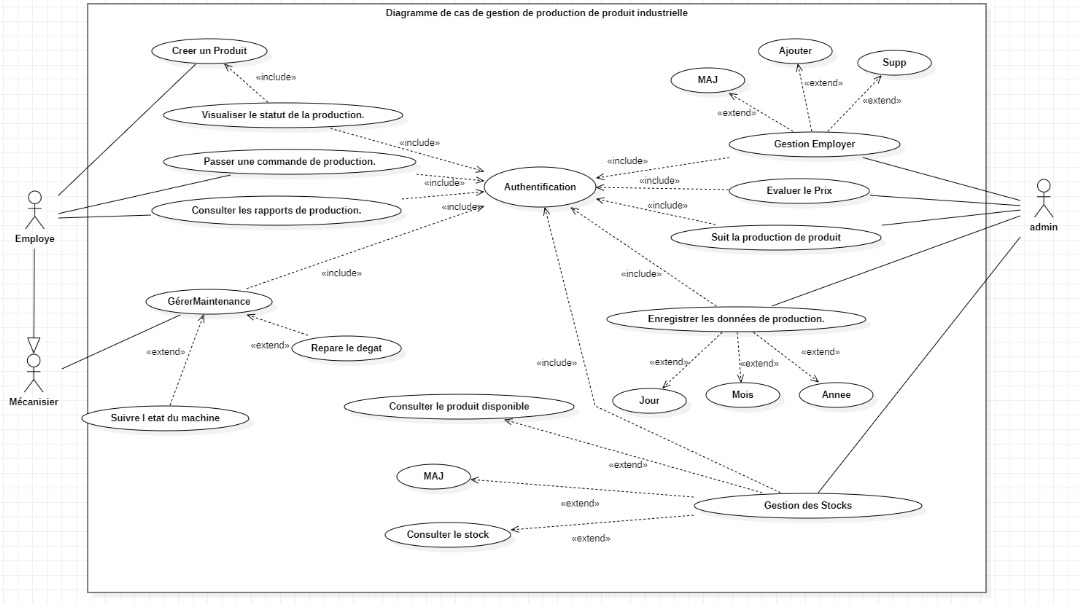
## Système étudié : Gestion de médiathèque.

La technique employée pour identifier les acteurs, les fonctionnalités et les classes consiste à analyser le texte décrivant le problème à résoudre.

|  |  |  |
| --- | --- | --- |
| **Objectives** | **Auteurs** | **Cas d’utilisation** |
| - Fournir à Coca-Cola un outil de gestion efficace qui améliorera la productivité, réduira les coûts de production et assurera une meilleure qualité de ses produits. | -Employé (EPL)  -Mécanicien (MCN)  -Admin (ADN) | **1**. Créer un produit ***(EPL)***  **2**. Visualiser le statut de la production ***(EPL)***  **3**. passer une commande de production ***(EPL)***  **4**. consulter les rapports de production ***(EPL)***  **5.** Gérer la maintenance ***(MCN)***  **6**. Gestion des employé ***(ADN)***  **7**. Evaluer les prix ***(ADN)***  **8**. Suit la production de produit ***(ADN)***  **9**. Enregistrer les données de productions ***(ADN)***  **10**. Gestion des Stocks ***(ADN)*** |

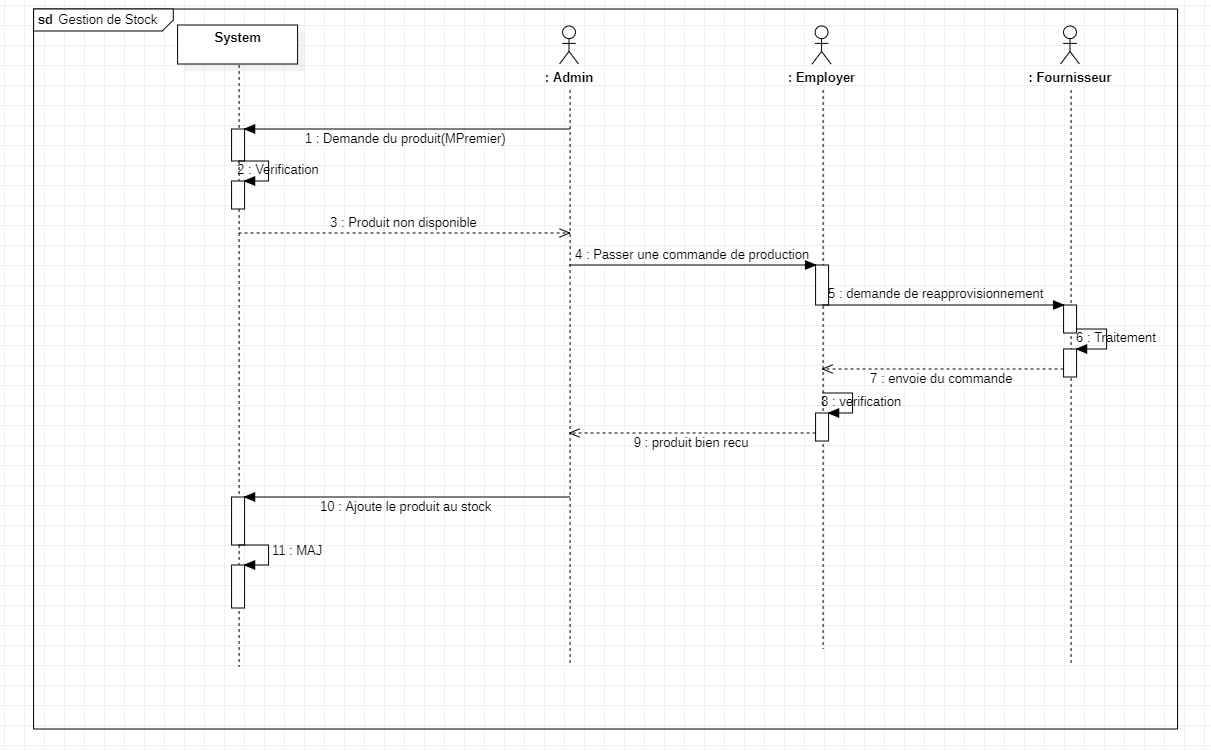
# Diagramme de cas d’utilisation

Ce diagramme statique met en évidence les relations entre les acteurs et les cas d'utilisation, illustrant ainsi les besoins des acteurs envers le système.



# Diagramme de séquence

Ce diagramme permet de détailler les scénarios associés à chaque cas d'utilisation en représentant les collaborations entre objets dans une perspective temporelle. L'accent est mis sur la séquence chronologique des messages échangés entre les objets, sans aborder le contexte ni l'état des objets, qui relèvent du domaine du diagramme de collaboration. La disposition des objets le long de l'axe vertical, appelé ligne de vie, détermine l'ordre chronologique des messages, avec le temps s'écoulant de haut en bas sur cet axe. En revanche, la disposition horizontale des objets n'a pas d'incidence sur le contenu ou l'interprétation du diagramme.



**Description Textuelle : Scenario Nominal (succès) du Cas d’Utilisation « Commande Produit »**

* Cas d’Utilisation : « gestion de stock »
* Scenario: Nominal
* Actor: Employer
* **Pré condition** : L’Admin doit être connecté au système.
* **Action:**

1. Le système affiche l’interface des services « Recherche un produit »

2. Le (ADN) saisie le produit recherche (Nom)

3. Le système a trouvé le produit recherche

4.le système affiche le produit

* **Les variants:**
* En 3 le système n’a pas trouvé le produit
* Le (ADN) demande a l employer de passer une commande au fournisseur
* Apres que le produit a été bien réceptionne (ADN) l’ajoute au stock

Et enfin il enregistre

# Diagramme de classe

Le diagramme de classes offre une vision statique du système à modéliser, mettant en scène des classes, des associations et des cardinalités.

1. Identification des classes.

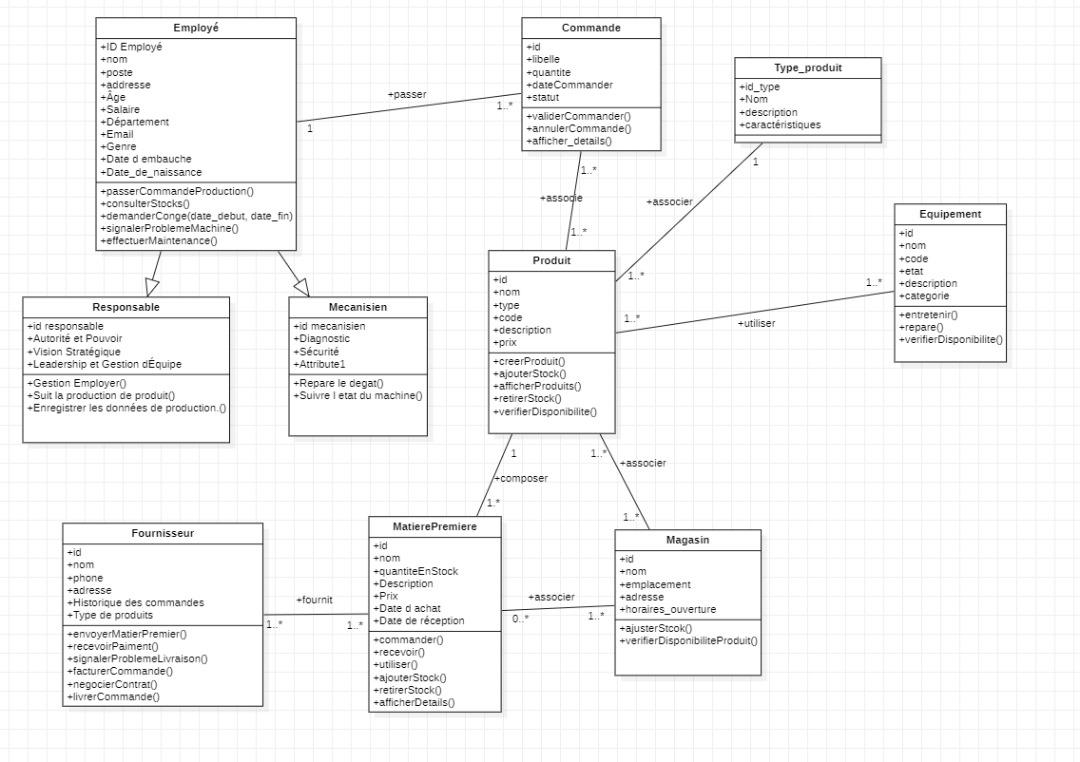
Voici les diffèrent objet du digramme :

-Employé -Equipement -

-Commande -Magasin

-Responsable -Fournisseur

-Mécanicien -MatierePremiere



1. **Identification des associations**

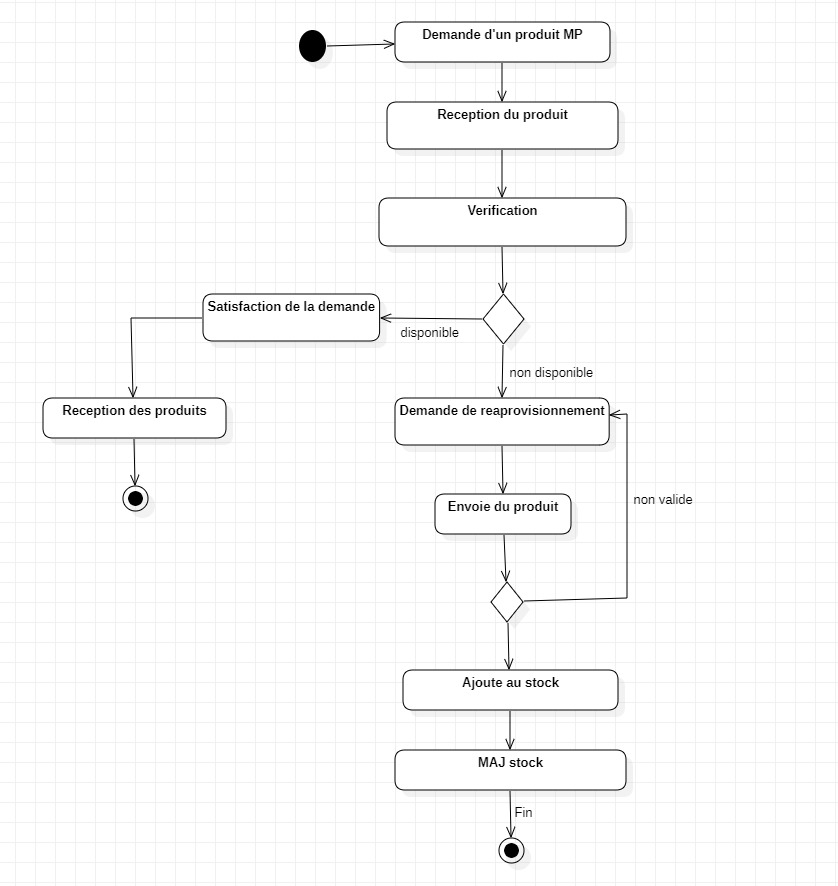
* Relation d’héritage (responsable, Mécanicien, Employé)
* Association « passer » (un a plusieurs) entre Employé et Commande.
* Association « associer » (plusieurs à plusieurs) entre produit et Commande.
* Association « associer » (un à plusieurs) entre produit et type produit.
* Association « utiliser » (plusieurs à plusieurs) entre produit et Equipement.
* Association « associer » (plusieurs à plusieurs) entre produit et Magasin.
* Association « composer » (un à plusieurs) entre produit et MatierPremier.
* Association « fournit » (plusieurs à plusieurs) entre Fournisseur et MatierePremiere.

Association « associer » entre MatierePremiere et Magasin.

# Diagramme d’activité

Un diagramme d'activité en UML est un outil de modélisation visuelle qui représente le flux de contrôle d'un processus ou d'un système. Il est particulièrement utile pour décrire les étapes séquentielles d'un algorithme, d'un workflow ou d'un scénario d'utilisation dans un système logiciel.

Dans ce diagramme, les activités sont représentées par des formes rectangulaires avec des coins arrondis, et les transitions entre les activités sont indiquées par des flèches. Chaque activité peut être annotée avec des actions spécifiques qui se produisent à cet état, et des conditions de transition peuvent être définies pour déterminer le passage d'une activité à une autre.

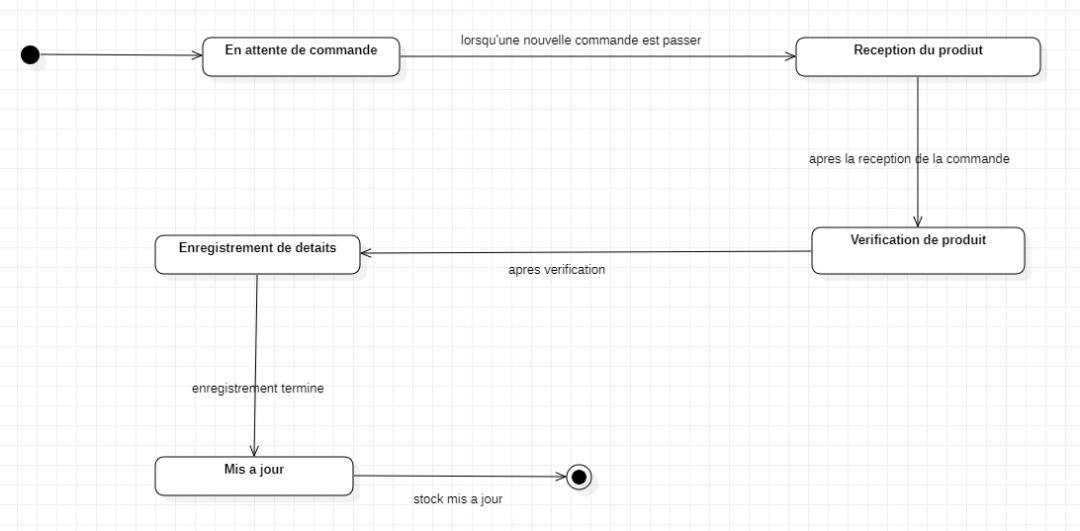


# Diagramme d’état transition

Le diagramme d'état de transition en UML est un autre outil de modélisation qui représente le comportement dynamique d'un système en décrivant les différentes étapes de ses objets ou entités et les transitions entre ces étapes en réponse à des événements.

Dans ce diagramme, chaque état possible d'un objet est représenté par une forme rectangulaire avec le nom de l'état à l'intérieur. Les transitions entre les états sont représentées par des flèches dirigées, indiquant le passage d'un état à un autre en réponse à un événement spécifique. Ces transitions peuvent être déclenchées par des événements externes ou des conditions internes.

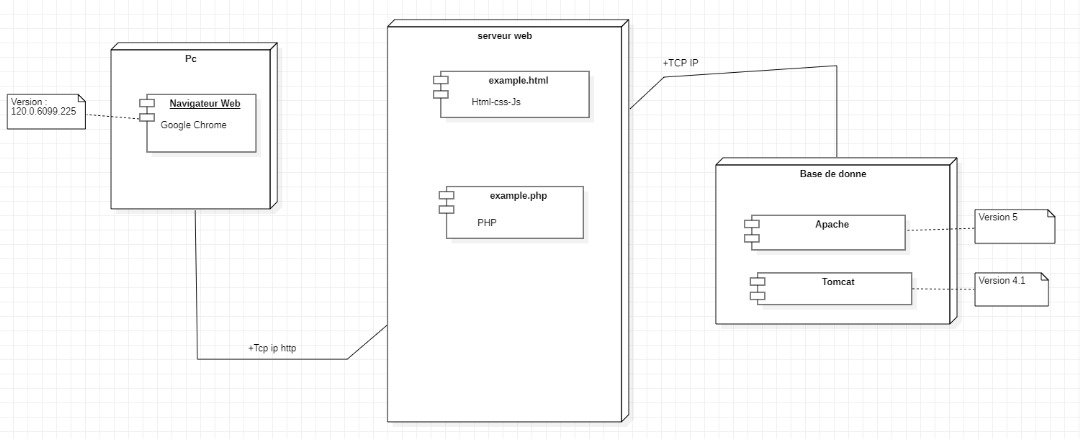
Chaque transition peut être annotée avec des actions ou des comportements associés à cette transition, ainsi que des gardes conditionnelles qui spécifient les conditions sous lesquelles la transition peut se produire. Par exemple, une transition peut avoir une garde qui spécifie que l'événement ne déclenche la transition que si certaines conditions sont remplies.



# Diagramme de déploiement

Le diagramme de déploiement en UML est un outil de modélisation qui décrit la configuration matérielle d'un système logiciel et la manière dont les composants logiciels sont déployés sur cette infrastructure matérielle.

Dans ce diagramme, les nœuds représentent les différentes entités matérielles, telles que des serveurs, des ordinateurs, des périphériques réseau, etc. Ces nœuds sont généralement représentés par des boîtes avec leur nom à l'intérieur.

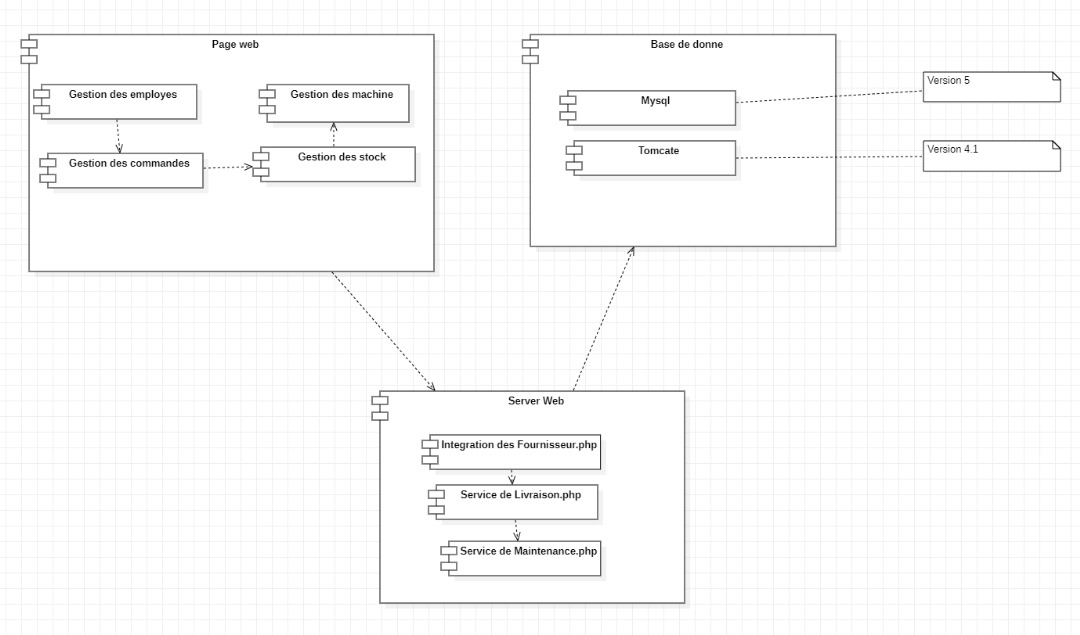


# Diagramme de composant

Le diagramme de composant en UML est un outil de modélisation qui représente les dépendances entre les composants logiciels d'un système, ainsi que les interfaces qu'ils exposent pour communiquer les uns avec les autres.

Dans ce diagramme, les composants sont représentés par des boîtes rectangulaires avec leur nom à l'intérieur. Chaque composant peut encapsuler une ou plusieurs fonctionnalités et peut dépendre d'autres composants pour fonctionner correctement. Les dépendances entre les composants sont représentées par des flèches dirigées indiquant la direction de la dépendance.

Les interfaces exposées par chaque composant sont représentées par des points de connexion (ports) sur les bords du composant. Ces ports sont des points d'entrée ou de sortie qui définissent les services ou les fonctionnalités accessibles à partir de ce composant. Les relations entre les ports de différents composants représentent les interactions et les communications entre ces composants.



# Diagramme de classe participant

1. Demande de produit

|  |  |
| --- | --- |
| Recherche :  Demande | |
| Nom du produit | Coca |
| Nom du produit | fanta |

|  |
| --- |
| Recherche :  Demande |
| Le produit recherché n’a pas été trouver. |

1. Ajout de produit sur le système

|  |
| --- |
| Ajouter  Description  Date de réception  Date d’achat  prix  Quantite  Nom  Ajout de MatierePremiere |