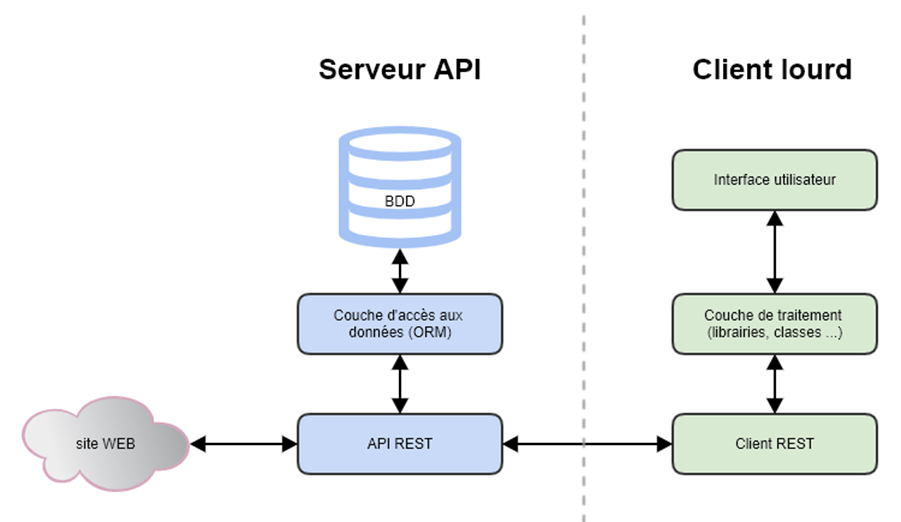
# Cahier des charges fonctionnels

## 1. Introduction

L'objectif de ce document est de définir les fonctionnalités et les exigences du système d'information pour une application de gestion de stock. Cette application permettra aux gérants de rechercher, commander, réserver et gérer des produits, les fournisseurs, et les clients.

## 2. Description générale et architecture



L'application de gestion de stock offrira les fonctionnalités suivantes :

### 2.1 Client Lourd

* Interface utilisateur pour la gestion des stocks, des commandes, les fournisseurs, les familles et les article à disposition sur le site web. (Commande automatique lorsque le stock arrive à X articles).
* L’application dépend d’une API développée par l’équipe pour accéder et gérer les données.

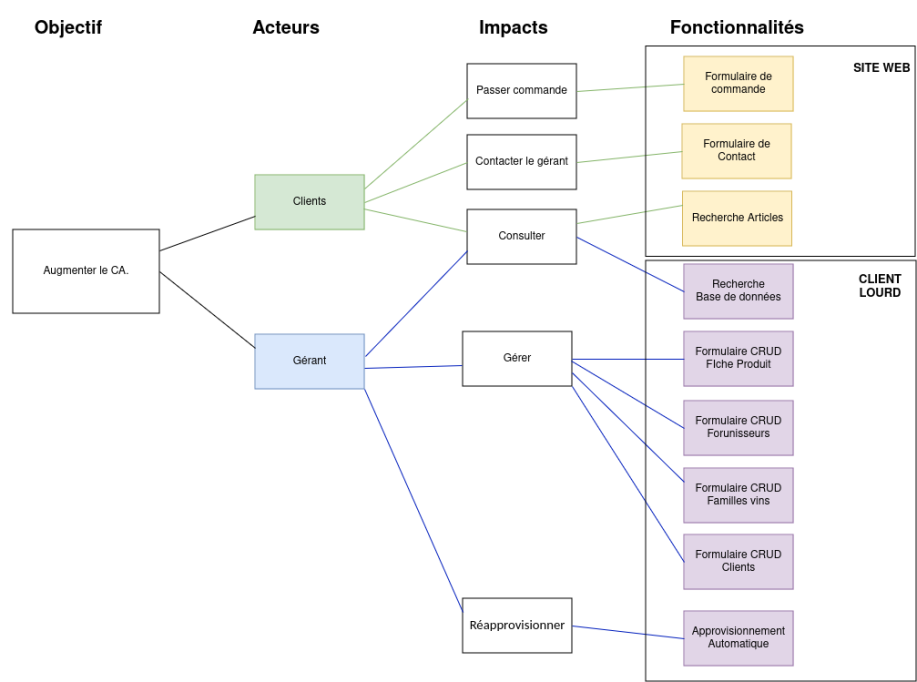
### 2.2 API

* Inscription et authentification des clients.
* Expose les données présentes dans la base à l’application et au site web.
* Propose et sécurise des routes accessibles.

### 2.3 Site Web

* Inscription et authentification des clients sur le site web.
* Recherche avancée de vins par catégorie, maisons ou année.
* Possibilité de commander des vins (en stock ou pas – pas visible pour le client) sur le site web via paiement en ligne.
* Gestion du profil client (modification des coordonnées, historique des commandes, etc.).

### 3 Périmètre Fonctionnel



## 3.1 Fonctionnalités du client lourd

### Gestion globale :

* L’utilisateur doit pouvoir afficher l'historique des commandes passer aux fournisseurs ou aux clients.

### Gestion des commandes :

* Le gérant gère son stock.
* Le gérant à la possibilité d’envoyer un bon de commande aux fournisseurs quand son stock à atteint un niveau minimum.

### Interface d’utilisateurs :

* Le gérant doit pouvoir ajouter/supprimer/modifier les fournisseurs, les familles, les articles (prix etc.…).
* Visualiser l’historique de ces commandes fournisseurs/clients.
* Gérer manuellement les stocks au besoin.

## 3.2 Fonctionnalités du site web

### Gestion globale :

* Les clients doivent pouvoir créer un compte en fournissant leur nom, leur adresse e-mail et un mot de passe sécurisé ainsi que pouvoir modifier leurs coordonnées.
* Le système doit permettre aux clients de se connecter en utilisant leur adresse e-mail et leur mot de passe.

### Recherche de vins :

* Les clients doivent pouvoir rechercher des vins par maisons, année, prix ou familles.
* Le site doit afficher les résultats de recherche de manière claire et organisée.

### Commande de vins :

* Les clients doivent pouvoir commander des vins, qu’ils soient en stocks ou pas

### Gestion des commandes :

* Les clients peuvent voir l’étape de leur commande.
* Les clients doivent pouvoir afficher l'historique des commandes.

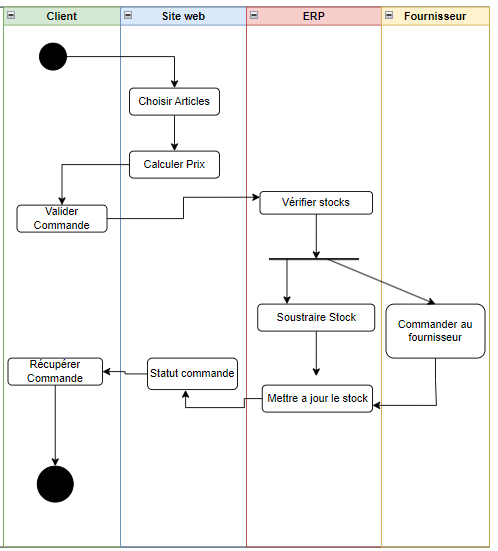
## 3.3 Fonctionnalités de l’API

* L’API expose différentes routes donnant accès aux données en base.
* Les routes disponibles dépendent des fonctionnalités attendues au site web et au client lourd.
* Certaines routes permettant de modifier le stock ne seront accessibles qu’au client lourd après authentification du gérant.

### 4. Contraintes techniques

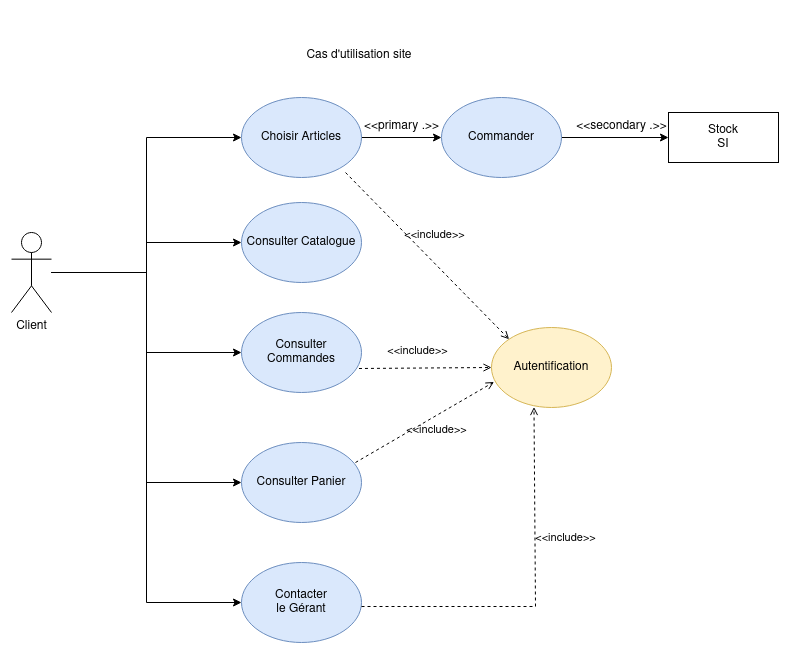
* Le Client Lourd doit être développée en utilisant le langage de programmation C#, avec WPF pour l’interface graphique.
* La base de données doit être MySQL.

## 5. Diagramme d’activité

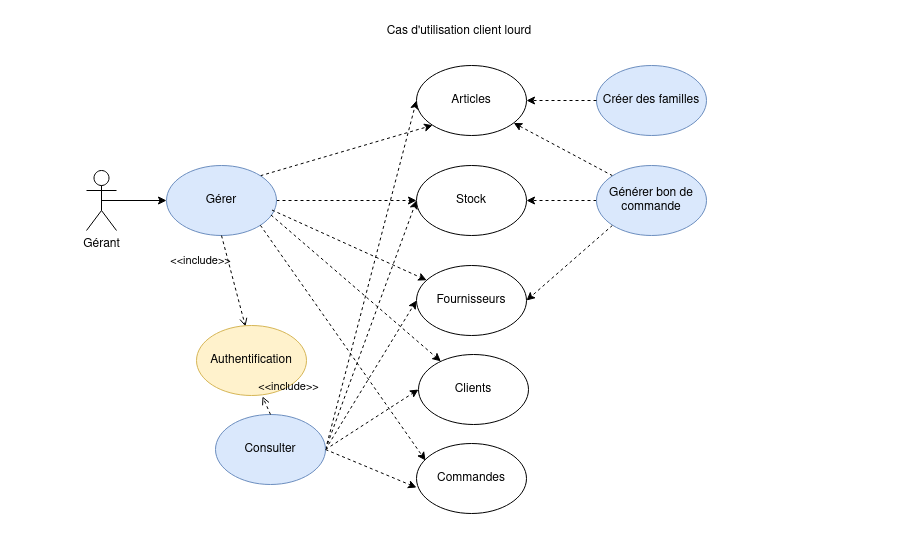


## 6. Diagrammes de cas d’utilisations

### 6.1 Site Web



### 6.2 Client Lourd



## 7. Description textuelle du cas d’utilisation

## « Passer commande »

### 7.1 Site Web

**Résumé :** Ce cas d'utilisation permet aux utilisateurs connectés de passer commande sur le site internet.

**Acteur principal :** Un utilisateur qui est déjà inscrit.

**Acteurs secondaires :** l'API

**Date :**

**Description des scénarios**

**Pré-conditions :**

* L'utilisateur possède un compte.
* L'utilisateur est identifié sur le site web.

**Scénario nominale :**

1. L'utilisateur parcourt le catalogue de produits.
2. L'utilisateur consulte le détail des articles.
3. Il ajoute les articles qui l'intéressent dans son panier.
4. Il valide son panier.
5. La commande est passée.

**Scénario alternatif 1 : Dysfonctionnement de l'API lors du passage de commande**

1. L'utilisateur parcourt le catalogue de produits.
2. Il consulte le détail des articles.
3. Il ajoute les articles qui l'intéressent dans son panier.
4. Il valide son panier.
5. Lors de la tentative de passage de commande, le système envoie une requête à l'API pour valider les informations de commande et de paiement.
6. L'API rencontre un problème (exemple : crash, problème de réseau, indisponibilité du serveur).
7. Le système affiche un message d'erreur à l'utilisateur, indiquant que le traitement de la commande a échoué en raison d'un problème technique.
8. Le système propose plusieurs options :
   * Retenter la validation de la commande (si le problème est temporaire).
   * Enregistrer la commande dans un état "en attente" pour une tentative ultérieure.
   * Annuler la commande et revenir au catalogue.
9. Si l'utilisateur choisit de retenter, le système fait une nouvelle tentative de connexion à l'API.
10. Si l'API devient de nouveau fonctionnelle :

* La commande est validée et confirmée.

Si le problème persiste, l'utilisateur est informé et peut consulter les options disponibles.

### 7.2 Client Lourd

**Résumé :** Ce cas d'utilisation permet au gérant connecté à son espace de passer commande dans le cas où le stock atteint le niveau minimal.

**Acteur principal :** Le gérant qui est admin de son système.

**Acteurs secondaires :** l'API

**Date :**

**Description des scénarios**

**Pré-conditions :**

* Le gérant est connecté sur son espace personnel.
* Le système et l’API sont opérationnels.

**Scénario nominale :**

1. Le gérant consulte l’inventaire de ses stocks.
2. Il souhaite passer une nouvelle commande à son fournisseur.
3. Il ajoute les articles.
4. La commande est passée.
5. Un bordereau de commande est envoyé au fournisseur.

**Scénario alternatif 1 : Échec de communication avec l'API**

1. Le gérant valide son panier après avoir ajouté les articles.
2. Le système tente d'envoyer le bordereau de commande à l’API fournisseur.
3. L'API n'est pas disponible ou renvoie une erreur.
4. Le système affiche un message d'erreur indiquant l'échec de la transmission de la commande.
5. Le gérant peut choisir de réessayer plus tard ou d'annuler la commande.
6. Fin du cas d'utilisation.