

DIGICHEES

Rédaction d'un cahier des charges techniques

Cahier des charges

*Rédigé le 09/06/2023 par BASKA Benjamin, GUILLON Antonin et HOTTON Robin
Dernière mise à jour : 12/06/2023*

Sommaire

Sommaire.....	2
Glossaire.....	3
I- Descriptif du projet.....	4
II- Equipe du projet.....	4
III- Contexte du projet.....	5
1 - Exposé de la situation.....	5
2- Nos objectifs :.....	5
IV- Description fonctionnelle des besoins.....	6
1- Accueil et connexion.....	6
2- Architecture technique et logicielle.....	7
❖ Architecture technique :.....	7
❖ Architecture logicielle :.....	7
3- Les diagrammes fonctionnels.....	11
❖ Diagramme de classe :.....	11
❖ Les cas d'utilisations :.....	12
4- Fonctionnalités.....	14
❖ Fonctionnalité [1] : Créer une commande Boutique.....	14
❖ Fonctionnalité [2] : Visualiser le stock.....	15
❖ Fonctionnalité [3] : Ajouter un objet.....	16
❖ Fonctionnalité [4] : Modifier un objet.....	17
❖ Fonctionnalité [5] : Se connecter.....	18
V- Sécurité.....	19
❖ Gestion des rôles :.....	19
❖ Authentification :.....	19
❖ VPS (Virtual Private Server) pour intranet :.....	19
VI- Campagne de test.....	20
❖ Objectif.....	20
❖ types de tests.....	20
❖ Rapports de test.....	20
❖ Gestion des anomalies.....	20
❖ Plan de test.....	21
❖ Validation des tests.....	21
❖ Suivi et revue des tests.....	21
VII- Budget.....	21
VIII- Calendrier.....	21

Glossaire

TPE : Toute petite entreprise.

DIGICHEES : Nom de la fromagerie

VBA : Visual Basic for Applications (VBA) est une implémentation de Microsoft Visual Basic qui est intégrée dans toutes les applications de Microsoft Office

CRUD : L'acronyme informatique anglais CRUD (pour Create, Read, Update, Delete) (parfois appelé SCRUD avec un "S" pour Search) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.

SGBDR : Système de Gestion de Base de Données Relationnelles.

I- Descriptif du projet

La TPE de la Fromagerie DIGICHEES a besoin d'une refonte de l'application et de la base données existante qui ont été créés il y a 20 ans et qui ne sont plus à jour et manque d'efficacité. Ce document est un cahier des charges pour la refonte d'un SI dans une TPE pour gérer une boutique physique et en ligne à partir de points de fidélités d'une fromagerie nommée « Fromagerie DIGICHEES ». Le nom de domaine désiré est « DIGICHEES.com ».

II- Equipe du projet

Nom – Prénom	Rôle projet	Société	E-mail de contact
GUILLON Antonin	Chef de projet	Diginamic.fr	guillon.antonin@yahoo.com
BASKA Benjamin	Développeur	Diginamic.fr	bbaska@diginamic-formation.fr
HOTTON Robin	Développeur	Diginamic.fr	rhotton@diginamic-formation.fr
MOMIN Valentin	Product Owner	Diginamic.fr	vmomin@diginamic.fr
CREVAN Hervé	Product Owner	Diginamic.fr	hcrevan@diginamic.fr

III- Contexte du projet

Dans le cadre de nos activités, nous menons une réflexion quant à l'opportunité de virtualiser l'envoi de cadeau de fidélité au client de l'entreprise.

Pour atteindre cet objectif, nous envisageons de mettre à disposition un outil intranet destiné à tenir à jour les stocks, à faciliter l'envoi des colis et à administrer facilement l'application.

Nous nous intéresserons ici plus particulièrement à la gestion des stocks.

1 – Exposé de la situation

Aujourd'hui, l'application est centralisée en interne sur Microsoft Access et implémentée en VBA. L'application consomme entre 50 à 60 Mo. Cette conception pose plusieurs problèmes :

- **Forte instabilité (bugs réguliers),**
- **Problèmes de maintenance,**
- **Faible possibilité d'évolution de développement,**
- **Manque de fluidité, d'accessibilité et de visibilité pour les utilisateurs.**

L'objectif est de réaliser une refonte complète d'une application de gestion en prenant en compte les problèmes relevés ci-dessus.

Les principaux axes de développements pour répondre à ces problématiques sont :

- ☐ Créer une commande Boutique
- ☐ Lancer mise à jour historique commande
- ☐ Visualiser le stock

- ☐ Gestion d'objet (CRUD sans le delete)
- ☐ Modification d'un prix
- ☐ Modification d'inventaire

2- Nos objectifs :

- **OBJECTIF 1** : Virtualiser la gestion des commandes fidélité
- **OBJECTIF 2** : Faire un stock comptable

IV- Description fonctionnelle des besoins

1- Accueil et connexion

Après consultation de la DSI/équipe technique, la plateforme doit être accessible selon les modalités suivantes : via un intranet de l'entreprise.

Aucun contenu ou fonctionnalité de la plateforme n'est accessible hors authentification.

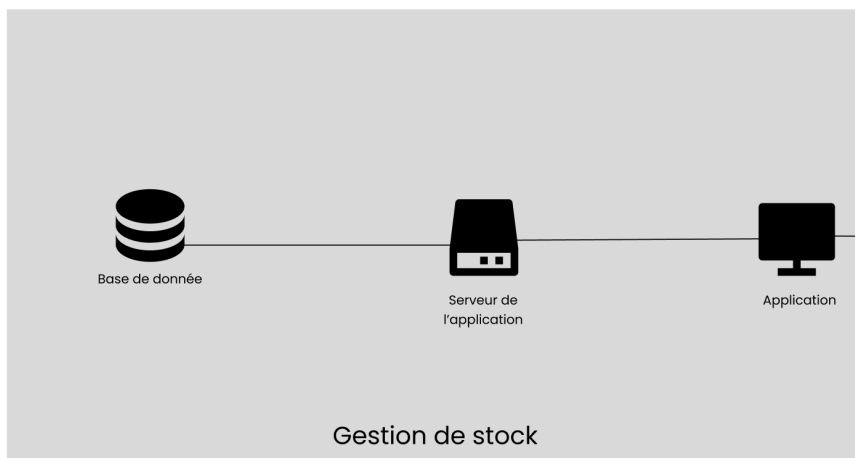
Il est possible d'accéder à la plateforme selon les rôles suivants :

- ❖ **Administrateur** : administre l'application
- ❖ **OP-colis** : Gère l'envoi des colis
- ❖ **OP-stock** : Gère la gestion des stocks

Nous nous intéresserons ici uniquement à OP-stock

2- Architecture technique et logicielle

❖ Architecture technique :



❖ Architecture logicielle :

Langages, frameworks et bibliothèques spécifiques

NOM	VERSION
Python	3.11
SqlAlchemy	2.0.15
Fastapi	0.95.2
Passlib	1.7.4
Pydantic	1.10.8
PyMySQL	1.0.3
Python-jose	3.3.0
Selenium	4.9.1
SQLAlchemy	2.0.15

Uvicorn	0.22.0
TypeScript	5.1.3
React	18
MUI	5
Jest	29.5.0

Pourquoi on utilise Python pour le back-end

Le python présente plusieurs avantages d'utilisation pour le back-end d'une application qui font qu'on l'a choisi plutôt qu'un autre langage :

- **Syntaxe claire et lisible** : Python a une syntaxe simple et facile à lire, ce qui en fait un langage idéal pour développer des applications backend.
- **Large écosystème et bibliothèques** : Python dispose d'une vaste bibliothèque standard qui offre de nombreuses fonctionnalités prêtes à l'emploi pour le développement web. De plus, il existe une multitude de bibliothèques tierces spécialisées dans différents domaines tels que le développement web, la manipulation de données, la sécurité, etc.
- **Rapidité de développement** : Grâce à sa syntaxe concise et à ses nombreuses bibliothèques, Python permet de développer rapidement des applications backend.
- **Portabilité** : Python est un langage interprété, ce qui signifie qu'il peut être exécuté sur différents systèmes d'exploitation sans nécessiter de modifications importantes.
- **Frameworks web populaires** : Python offre plusieurs frameworks web populaires tels que Django et Flask, qui simplifient le développement web en fournissant des fonctionnalités et des outils prêts à l'emploi.
- **Scalabilité** : Bien que Python ne soit pas aussi rapide que certains autres langages, il est souvent préféré pour les projets qui nécessitent une évolutivité horizontale (scaling out). Cela signifie que l'application peut être exécutée sur plusieurs serveurs, ce qui permet de gérer une charge de trafic plus importante.

Pourquoi on utilise React/Typescript pour le front-end

Notre choix de technologie pour le front était constitué de Angular, Vue et React, cependant comme nous avons plus d'expérience en React, nous l'avons logiquement choisi. React et Typescript présente plusieurs avantages pour le développement front-end d'une application qui font qu'on les a choisis :

- **Performances élevées** : React est une bibliothèque JavaScript optimisée pour la création d'interfaces utilisateur réactives et performantes. Son approche de rendu virtuel et sa capacité à mettre à jour uniquement les parties nécessaires de l'interface utilisateur permettent d'obtenir de bonnes performances, même avec des applications complexes.
- **Réutilisabilité des composants** : React repose sur un modèle de composants réutilisables. Il permet de diviser l'interface utilisateur en petits morceaux autonomes appelés composants, ce qui facilite la réutilisation du code et la maintenance de l'application.
- **Gestion efficace de l'état** : React offre une gestion simplifiée de l'état de l'application grâce à son architecture basée sur les composants. Il utilise un concept appelé "state" pour stocker et gérer les données de l'application, permettant ainsi de créer des interfaces utilisateur dynamiques et interactives.
- **Communauté et écosystème robustes** : React bénéficie d'une communauté active et d'un écosystème riche en outils, bibliothèques et frameworks complémentaires.
- **TypeScript** : TypeScript est un sur-ensemble de JavaScript qui ajoute le typage statique au langage. L'utilisation de TypeScript avec React permet de détecter les erreurs de manière précoce et d'améliorer la qualité du code. TypeScript fournit également des fonctionnalités de refactoring avancées, une meilleure intégration avec les environnements de développement et une documentation automatique.
- **Développement collaboratif** : L'utilisation de TypeScript avec React facilite le travail en équipe. Grâce au typage statique, il est plus facile de comprendre et de communiquer les intentions du code entre les membres de l'équipe.

Langages, frameworks et librairies spécifiques

NOM	VERSION
MySQL	8.0.33

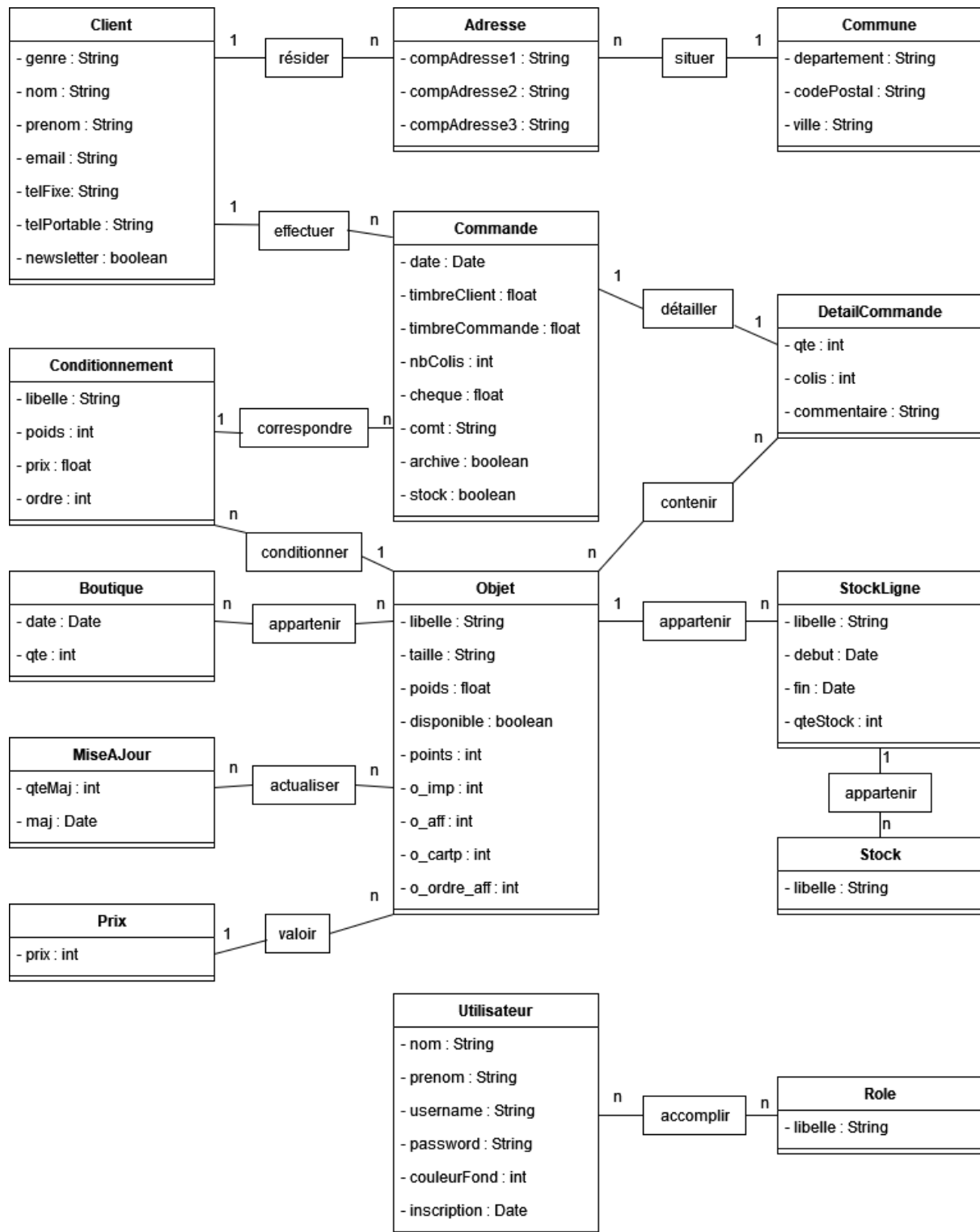
Pourquoi on utilise MySQL pour la base de donnée

On a décidé de choisir une base de donnée de type relationnelle (SGBDR), ici MySQL, pour plusieurs raisons :

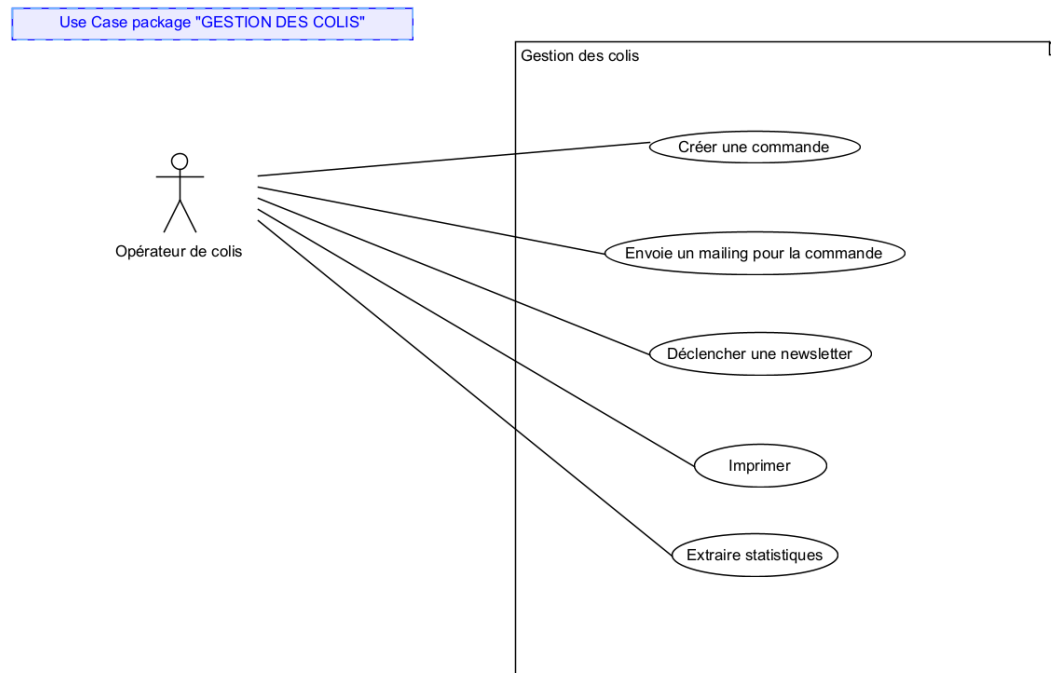
- **Performance** : MySQL est reconnu pour ses performances élevées et sa capacité à gérer des charges de travail importantes. Il est capable de traiter efficacement de grandes quantités de données et de répondre rapidement aux requêtes.
- **Stabilité et fiabilité** : MySQL est utilisé depuis de nombreuses années et est considéré comme un SGBD mature et stable. Il est fiable et bien testé, ce qui en fait un choix sûr pour les applications qui nécessitent une disponibilité et une stabilité élevées.
- **Large adoption et support communautaire** : MySQL est l'un des SGBD les plus populaires et largement utilisés dans l'industrie. Il dispose d'une vaste communauté d'utilisateurs et de développeurs qui fournissent un support actif, des ressources en ligne et des mises à jour régulières. Cette large adoption facilite également le recrutement de développeurs ayant une expérience avec MySQL.
- **Évolutivité** : MySQL offre des fonctionnalités avancées pour la gestion de bases de données de grande taille. Il prend en charge la réplication, le partitionnement, les clusters et d'autres techniques qui permettent de faire évoluer la base de données pour gérer des charges de travail croissantes.
- **Facilité d'utilisation** : MySQL est relativement facile à installer, à configurer et à utiliser. Il fournit des outils d'administration conviviaux et une syntaxe SQL claire et bien documentée. De plus, il est compatible avec de nombreux langages de programmation et frameworks, ce qui facilite son intégration dans des applications existantes.
- **Coût** : MySQL est un SGBD open source et gratuit.

3- Les diagrammes fonctionnels

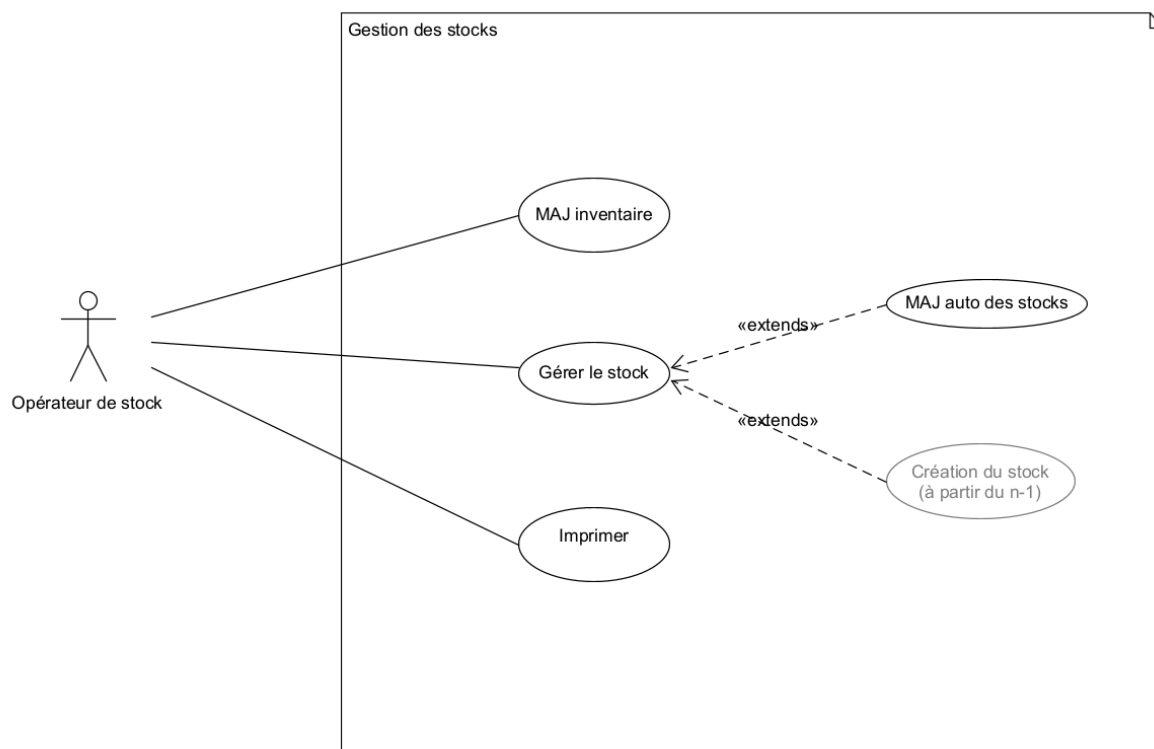
❖ Diagramme de classe :



Cas d'utilisation OP-colis



Cas d'utilisation OP-stock

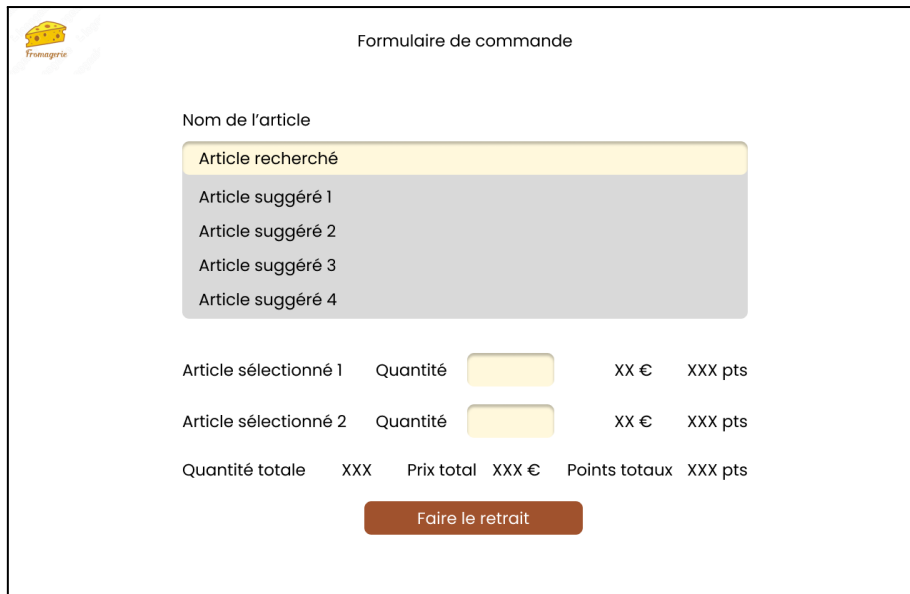


4- Fonctionnalités

❖ Fonctionnalité [1] : Créer une commande Boutique

Description :

En tant que **OP-stock**, je peux retirer un objet du stock



Formulaire de commande

Nom de l'article

Article recherché

Article suggéré 1

Article suggéré 2

Article suggéré 3

Article suggéré 4

Article sélectionné 1 Quantité XX € XXX pts

Article sélectionné 2 Quantité XX € XXX pts

Quantité totale XXX Prix total XXX € Points totaux XXX pts

Faire le retrait

Contraintes et règles de gestion : être connecté, avoir l'objet en stock

Niveau de priorité : 1

input de recherche: champs de recherche qui propose des suggestions au fur et à mesure de la saisie

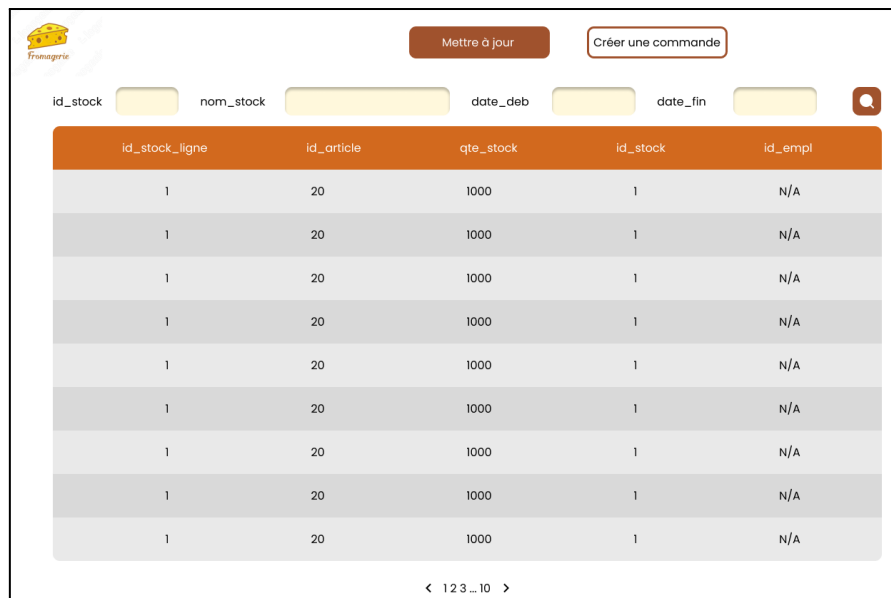
input quantité d'un article: champs où l'on saisie la quantité commandée de l'article en question

button "Faire le retrait" : Bouton qui valide la saisie de la commande

❖ Fonctionnalité [2] : Visualiser le stock

Description :

En tant que **OP-stock**, je peux visualiser le stock



id_stock_ligne	id_article	qte_stock	id_stock	id_empl
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A
1	20	1000	1	N/A

Contraintes et règles de gestion : être connecté et avoir des objet en stock

Niveau de priorité : 1

input de filtre (id_stock, nom_stock, date_debut, date_fin) : champs qui filtreront les données du tableau

button "Loupe" : Bouton qui valide la saisie des filtres, et les applique au tableau

button "Mettre à jour" : Bouton qui lance la fonctionnalité de mise à jour de l'historique des commandes

button "Créer un commande" : Bouton qui lance la fonctionnalité de création de commande en boutique

input "Flèche et chiffres sous le tableau" : Boutons de navigation du tableau

❖ Fonctionnalité [3] : Ajouter un objet

Description :

En tant que **Admin**, je peux pouvoir ajouter un objet



The screenshot shows a web form titled "Ajouter un objet" with a "Tronagrie" logo in the top left corner. The form contains the following fields:

- Nom de l'article**: A single-line text input field.
- Prix**: A text input field with a "€" symbol on the right.
- Coupons**: A text input field with a "pts" symbol on the right.
- Taille**: A text input field with a "cm" symbol on the right.
- Poids**: A text input field with a "kg" symbol on the right.
- Description**: A multi-line text area.

At the bottom center of the form is a brown button labeled "Ajout de l'objet".

Contraintes et règles de gestion : être connecté et les champs doivent être remplis.

Niveau de priorité : 1

input "Nom de l'article" : champs où l'on renseigne le nom de l'article à ajouter en base de donnée

input "Prix" : champs où l'on renseigne le prix de l'article à ajouter en base de donnée

input "Taille" : champs où l'on renseigne la taille de l'article à ajouter en base de donnée

input "Coupons" : champs où l'on renseigne le nombre de coupons de l'article à ajouter en bdd afin de pouvoir le commander

input "Poids" : champs où l'on renseigne le poids de l'article à ajouter en base de donnée

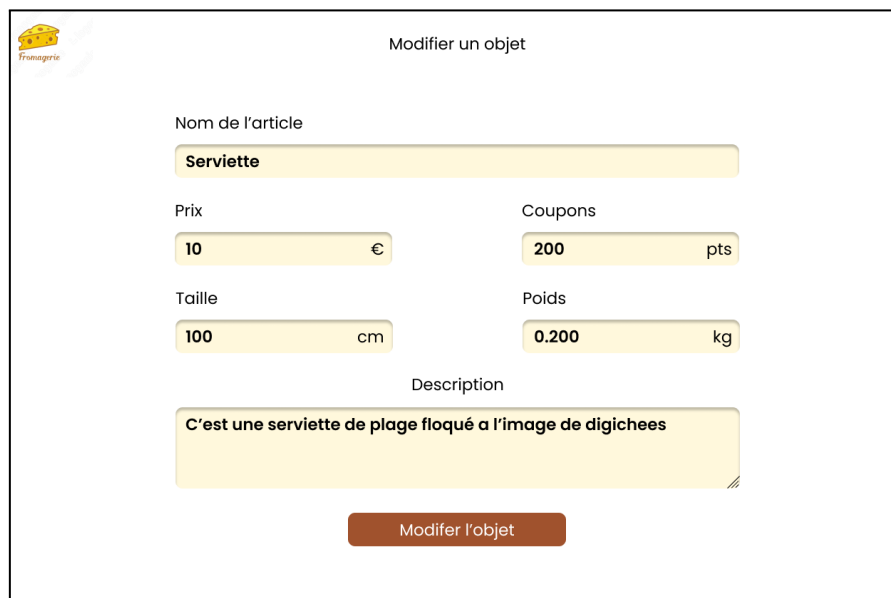
input "Description" : champs où l'on renseigne la description de l'article à ajouter en base de donnée

bouton "Ajout de l'objet" : Bouton qui valide la création de l'objet

❖ Fonctionnalité [4] : Modifier un objet

Description :

En tant que **Admin**, je peux pouvoir modifier un objet



The screenshot shows a web form titled "Modifier un objet" with a "Trimagrie" logo in the top left. The form contains several input fields and a description area:

- Nom de l'article**: A text input field containing "Serviette".
- Prix**: A numeric input field containing "10" with a "€" symbol to its right.
- Coupons**: A numeric input field containing "200" with a "pts" symbol to its right.
- Taille**: A numeric input field containing "100" with a "cm" symbol to its right.
- Poids**: A numeric input field containing "0.200" with a "kg" symbol to its right.
- Description**: A text area containing the text "C'est une serviette de plage floqué a l'image de digichees".
- Modifier l'objet**: A brown button at the bottom center of the form.

Contraintes et règles de gestion : être connecté et les champs doivent être remplis.

Niveau de priorité : 2

inputs : formulaire identique à celui de la création d'objet

bouton “Modifier de l’objet” : Bouton qui valide la modification de l’objet

❖ Fonctionnalité [5] : Se connecter

Description :

En tant que **Admin**, je peux pouvoir modifier un objet



The image shows a login form for a website called 'Fromagerie'. At the top center is a logo featuring a wedge of Swiss cheese with holes, with the word 'Fromagerie' written in a cursive font below it. Below the logo, there are two input fields. The first is labeled 'Email' and the second is labeled 'Mot de passe'. Both fields are empty and have a light yellow background. Below the input fields is a brown button with the text 'Se connecter' in white.

Contraintes et règles de gestion : être connecté et les champs doivent être remplis.

Niveau de priorité : 2

input “Email” : endroit où il faut mettre son email

input “Mot de passe” : endroit où il faut mettre son mot de passe

bouton “Se connecter” : Bouton qui valide la connexion

V- Sécurité

❖ Gestion des rôles :

La plateforme informatique devra comporter un système de gestion des rôles permettant de définir des niveaux d'accès et de privilèges pour les différents utilisateurs. Ce système devra garantir que chaque utilisateur possède uniquement les autorisations nécessaires pour effectuer ses tâches. Les rôles sont Administrateur, OP-colis et OP-stock. Un utilisateur peut avoir un ou plusieurs rôles.

❖ Authentification :

L'authentification est un élément essentiel de la sécurité du système. Tous les utilisateurs devront être soumis à un processus d'authentification avant d'accéder à la plateforme. Ce processus devra s'appuyer sur des mécanismes robustes, tels que l'identification par mot de passe fort. De plus, les informations d'authentification des utilisateurs devront être stockées de manière sécurisée, en utilisant des techniques de hachage et de chiffrement appropriées. Ici, nous utiliserons bcrypt de passlib, qui est une fonction de hachage utilisant l'algorithme de chiffrement Blowfish et nous utilisons OAuth2 de fastapi pour la génération et la gestion de token.

❖ VPS (Virtual Private Server) pour intranet :

La mise en place d'un VPS pour l'intranet permettra de renforcer la sécurité en isolant le système de la plateforme informatique du reste du réseau. Le VPS devra

être configuré de manière à empêcher les accès non autorisés de l'extérieur et à assurer une protection contre les attaques potentielles.

VI- Campagne de test

❖ Objectif

Grâce à Jest pour React et Pytest pour Python, nous cherchons à réaliser un maximum de tests pour couvrir l'ensemble des fonctionnalités et des cas d'utilisation.

❖ types de tests

- **Tests unitaires** : Vérifier le bon fonctionnement des composants, fonctions ou modules individuels.
- **Tests d'intégration** : Vérifier le bon fonctionnement des différentes parties de l'application lorsqu'elles sont combinées.
- **Tests d'interface utilisateur** : S'assurer que l'application réagit correctement aux actions de l'utilisateur.
- **Tests de performance** : Évaluer la réactivité, la capacité de charge et la stabilité du système.

❖ Rapports de test

Les résultats des tests, y compris les échecs, les erreurs et les performances, doivent être consignés dans des rapports détaillés.

❖ Gestion des anomalies

Toutes les anomalies identifiées lors des tests doivent être enregistrées dans un système de suivi des problèmes.

❖ Plan de test

Un plan de test détaillé doit être élaboré, décrivant les différentes étapes des tests à effectuer, les ressources nécessaires, les délais et les responsabilités.

❖ Validation des tests

Les tests doivent être validés par l'équipe de développement pour s'assurer de leur adéquation aux fonctionnalités et exigences du système.

❖ Suivi et revue des tests

Des revues régulières des tests doivent être effectuées pour évaluer leur efficacité et apporter des améliorations continues.

VII- Budget

Coût	
Poste: Développeur	300€/jour
Poste: Product Owner	300€/jour
Poste: Chef de projet	500€/jour

VIII- Calendrier

Pour le développement de l'application nous avons décidé d'utiliser une méthodologie AGILE, pour permettre une mise à jour plus régulière des fonctionnalités en fonction du besoin du client. Des dailys et rétrospectives seront donc mis en place et un planning prévisionnel vous est fournis ci-dessous :

Pré-requis	Sprint 1	Sprint 2	Sprint 3
[v1]	[v1]	[v1]	[v2]
Création de la BDD	Gestion d'objet (CRUD sans le delete)	Connection	Créer une commande Boutique
Création du projet	Gestion des stocks (CRUD sans le delete et le create)	Lancer mise à jour stocks	Impression OP_stock
Installation des dépendances	Gestion utilisateur (CRUD)		Impression Admin

Chaque sprint a une durée de deux semaines, les prérequis, eux, se font sur une semaine.