DIGICHEESE.com

Cahier des charges techniques

Rédigé le 28/10/2024 par Alexis Murail, David Meaux, Anne-Laure Lemaitre
Dernière mise à jour : 02/01/2025

Gestion documentaire

Versioning

Version	Date	Modifications	Rédacteur	Correcteur	Statut
1.0	28/10/2024	Création du document	Alexis Murail-David Meaux-Anne-Laure Lemaitre	Alexis Murail-David Meaux-Anne-Laure Lemaitre	Terminé
1.1	30/12/2024	Modification du document	Alexis Murail-David Meaux-Anne-Laure Lemaitre	Alexis Murail-David Meaux-Anne-Laure Lemaitre	En cours

Liste de diffusion

Nom et Prénom	Rôle	Actions
Fromagerie DIGICHEESE	Client principal	Signataire
Christophe Germain	Directeur des Projets, PO	Approbateur
Robin Hotton / Valentin Momin	PO	Approbateur
Anne-Laure Lemaitre	Cheffe de Projet	Rédacteur
Alexis Murail	Lead Développeur	Rédacteur
David Meaux	Développeur Backend	Rédacteur

Sommaire

Terminologie

API : (Application Programming Interface en anglais), une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

Authentication : vérifier qu'un utilisateur est bien celui qu'il prétend être, vérifier l'identité d'un utilisateur.

Authorisation : accorder à un utilisateur l'accès aux seules données et actions auxquelles il a été autorisé.

AZ : (Availability Zone) : La zone de disponibilité est où les centres du fournisseur cloud sont localisé.

Backend, back-end ou **back** : la partie de l'application non visible par l'utilisateur et contenant la logique de gestion.

CRUD : (Create, Read, Update, Delete) : créer, lire, modifier, supprimer.

Filtrage IP : Un pare-feu de filtrage des paquets IP permet de créer un ensemble de règles qui interdit ou autorise le trafic sur une connexion réseau.

Frontend, **front-end** ou **front** : la partie de l'application avec laquelle l'utilisateur interagit.

HTTPS : Le protocole HTTPS (Hyper-Text Transfer Protocol Secured) est une extension sécurisée du protocole HTTP, le « S » pour « Secured » (sécurisé) signifie que les données échangées entre le navigateur de l'internaute et le site web sont chiffrées et ne peuvent en aucun cas être espionnées (confidentialité) ou modifiées (intégrité).

IAM : IAM (Identity Access Management) est un cadre de politiques et de technologies visant à garantir que les utilisateurs appropriés ont un accès adéquat aux ressources technologiques.

RBAC : RBAC (Role-Based Access Control) veut dire classer les utilisateurs en groupes, chacun d'entre eux disposant d'un ensemble de privilèges définis lui permettant d'accéder à des champs de données spécifiques ou de les modifier.

SSL : (Secure Sockets Layer) Le SSL est une technologie standard de sécurisation des connexions Internet par le chiffrement des données transitant entre un navigateur et un site web (ou entre deux serveurs).

SSO : SSO (Single Sign-On) est un service d'authentification de session et d'utilisateur qui permet à un utilisateur d'utiliser un ensemble d'informations d'identification (par exemple, nom et mot de passe) pour accéder à plusieurs applications.

SQL : SQL (Structured Query Language) est un langage qui peut être utilisé pour écrire des requêtes afin de rechercher et d'extraire des informations spécifiques d'une base de données.

UML : UML (Unified Modeling Language) est un langage de modélisation qui permet de représenter visuellement les systèmes logiciels sous forme de diagrammes. Il aide à mieux comprendre la structure et le comportement d'un système dans son intégralité, et facilite la conception et le développement, par exemple d'une application, en offrant une vue globale et détaillée du système, de son architecture et de ses processus.

World Wide Web Consortium (W3C) : Le World Wide Web Consortium (W3C) est une organisation internationale d'intérêt public à but non lucratif où les organisations membres, une équipe à plein temps et le public travaillent ensemble à l'élaboration de normes Web. Les normes mondiales du W3C constituent la boîte à outils pour des solutions web évolutives, permettant aux innovateurs de résoudre des problèmes difficiles, en fournissant les bases appropriées pour répondre aux exigences en matière d'accessibilité, d'internationalisation, de respect de la vie privée et de sécurité sur le web. Les normes qui répondent aux besoins variés de la société ne sont pas créées par une seule entreprise, mais par le travail de la communauté du consortium Web.

WCAG 2.0 : Web Content Accessibility Guidelines 2.0 couvre un large éventail de recommandations visant à rendre le contenu du Web plus accessible. Le respect de ces lignes directrices permet de rendre le contenu accessible à un plus grand nombre de personnes en situation de handicap, qu'elles présentent des difficultés isolées ou combinées, telles que : malvoyance ou cécité, surdité et perte d'audition, les troubles de l'apprentissage, les difficultés cognitives, la limitation des mouvements, les troubles de la parole, la photosensibilité. Le respect de ces lignes directrices rendra aussi souvent votre contenu Web plus utilisable par les utilisateurs en général.

Verrues : En informatique, le terme "verrue" désigne une modification temporaire dans un système logiciel ou matériel pour résoudre un problème ponctuel ou une contrainte. Ce type d'ajout est considéré comme une solution de contournement qui ne respecte pas les bonnes pratiques de développement mais qui permet de faire fonctionner le système dans l'immédiat.

I – Descriptif du projet

Ce projet se concentre sur la transformation de l'application personnalisée de DIGICHEESE basée sur Microsoft Access en une application moderne associant une base de données avec une interface web (DIGICHEESE.com). Digi3 propose de créer une base de données SQL de qualité serveur qui interagit avec un frontend basé sur le web à travers une interface de programmation d'application (API). L'utilisation de l'expertise de Digi3 dans le développement d'un backend sécurisé séparé de l'interface web donnera à DIGICHEESE une solution moderne qui simplifier la gestion des commandes par les employés le suivi les points de fidélité, la gestion des données des clients, ainsi que le contrôle et la supervision des processus de commande du début à la fin.

En prévoyant de séparer le stockage des données de l'interaction avec l'utilisateur, le Digi3 prévoit de mettre en œuvre des processus pour stocker et transmettre en toute sécurité les données relatives aux commandes, aux produits et aux clients, tout en donnant à DIGICHEESE la flexibilité nécessaire pour entrer dans l'ère moderne avec une plate-forme qui est plus facile à utiliser et à maintenir, tout en fournissant la structure nécessaire pour adapter plus facilement l'interface, l'expérience de l'utilisateur et la structure des données en fonction de l'évolution des besoins. De plus, cette structure permettra à DIGICHEESE de se mettre en conformité avec le Règlement Général sur la Protection des Données (RGPD) en fournissant une sécurité des données, une gouvernance, et des politiques de lignage qui sont en ligne avec les réglementations actuelles de l'Union Européenne et de la France sur la protection des données et de la vie privée.

L'équipe projet sera composée de deux product owners, un chef de projet, un lead développeur (full stack) et un développeur backend, un testeur ainsi qu'un designer d'interface et d'expérience utilisateur. Digi3 sera en mesure de répondre aux attentes de DIGICHEESE en réalisant ce projet dans un délai de 45 jours maximum sur site et en ne dépassant pas le budget maximum de 45 000,00 €. Compte tenu de notre charge de travail actuelle et prévue, Digi3 estime qu'il sera en mesure de terminer ce projet d'ici le 4 juillet 2025 s'il est approuvé et démarré d'ici le 5 mai 2025. Cependant, en raison de problèmes imprévus, cette date peut changer, et Digi3 travaillera avec DIGICHEESE pour adapter le calendrier si nécessaire.

Nous vous remercions de lire ce document qui décrit les détails permettant de mener à bien ce projet dans le respect des délais, du budget et des modifications demandées incluant notamment la priorisation du développement des fonctionnalités ayant trait à la gestion des colis et du profil Administrateur. Digi3 souhaite profiter de cette occasion pour remercier DIGICHEESE de lui avoir confié ce projet ainsi que les rouages de son entreprise et les données de ses fidèles clients.

Des éléments complémentaires sont disponibles dans la note de cadrage.

II – Equipe du projet

Nom – Prénom	Rôle projet	Société	E-mail de contact
Fromagerie DIGICHEESE	Client principal	Fromagerie DIGICHEESE	contact@digicheese.com
Christophe GERMAIN	Directeur de Projets et PO	Diginamic.fr	cgermain@diginamic.fr
Robin HOTTON / Valentin MOMIN	PO	Diginamic.fr	rhotton@diginamic-formation.fr / vmomin@diginamic-formation.fr
Anne-Laure LEMAITRE	Cheffe de projet	Diginamic.fr	allemaitre@diginamic-formation.fr
Alexis MURAIL	Lead Dev / Développeur Full stack	Diginamic.fr	amurail@diginamic-formation.fr
David MEAUX	Développeur Backend	Diginamic.fr	dmeaux@diginamic-formation.fr
Marie Curie	Testeuse	Diginamic.fr	mcurie@diginamic-formation.fr
Jules VERNE	Designer UI/UX	Diginamic.fr	jverne@diginamic-formation.fr

III – Contexte du projet

Dans le cadre de nos activités, nous menons une réflexion sur la refonte d'un Système Informatique (SI) au sein d'une TPE : la fromagerie DIGICHEESE.

Cette entreprise familiale vend ses produits à des distributeurs de grandes enseignes ainsi qu'à des particuliers. Il propose un programme de fidélité sous forme de points à découper sur les produits, permettant aux clients de recevoir des goodies. La gestion de ce programme de fidélité nécessite une coordination entre différents intervenants. Actuellement, ces derniers utilisent une application développée sous Microsoft Access Office 2000, qui présente de nombreux problèmes (incompatibilité, rigidité, environnement peu ergonomique, *etc.*).

Dans ce contexte, nous envisageons de mettre à disposition un outil interne à destination des différents intervenants au sein de l'entreprise DIGICHEESE (administrateur, opérateur colis et opérateur stock).

1 – Exposé de la situation

a. Description générale de l'application actuelle

Actuellement, l'entreprise DIGICHEESE rencontre divers problématiques d'évolution, de maintenance et de gestion de l'application motivant la migration vers un client plus léger. Dans un premier temps, d'un point de vue gestion des colis, les opérateurs (colis ou stock) disposent d'une application en interne développée sous Access Office 2000 implémenté en VBA. L'analyse de l'application a identifié divers problèmes incluant entre autres :

- *Forte instabilité :* bugs Access réguliers. De surcroit, le support Microsoft n'est plus assuré depuis 2009 et Access 2000 n'est plus compatible à partir de Windows Vista.
- *L'application en mode client lourd* (consommation estimée entre 50 à 60 Mo) est peu performante par rapport aux technologies actuelles.
- *L'interface visuelle est rigide* et dépassée (nécessité de revoir le design et l'ergonomie)

Il est quasi *impossible de maintenir et de faire évoluer* l'application qui a subie de nombreuses "verrues" au fil du temps.

Dans un second temps, l'analyse de l'architecture globale du système d'information a montré que d'autres aspects pouvaient être améliorés ou des fonctionnalités ajoutées. Toutefois, en l'état actuel, les problèmes d'instabilité, de maintenance, de navigabilité pour les utilisateurs et l'impossibilité de faire évoluer la solution ne permettent pas ces améliorations. A titre d'illustration, une modélisation de l'architecture du système actuel est disponible en Annexe 1.

b. Description technique de l'application actuelle

Description technique de l'application actuelle	**Limites atteintes par l'application aujourd'hui**
Application a été créée via Access Office 2000 il y a plus de 20 ans	Il n'est officiellement pas pris en charge sur Windows Vista ou les versions ultérieures de Windows
Mise à jour au fur et mesure dans le temps avec des compétences limitées	Le support standard pour Office 2000 a pris fin le 30 juin 2004 et le support étendu a pris fin le 14 juillet 2009
	Client lourd peu performant en rapport aux technologies actuelles
	Interface rigide et dépassée
	Peu/pas de modification et/ou amélioration possibles
	Design et graphisme mal structuré, inesthétique et peu ergonomique

Figure 1 : Description technique de l'application actuelle.

c. Identification des besoins

Au sein de l'application, trois rôles (cumulables) ont été identifiés (les deux premiers étant à prioriser) :

- *Administrateur* : Initialement, la fonction d'administrateur du système avait un rôle de paramétrage (notamment la gestion des emballages, le calcul du montant de l'affranchissement, les poids des articles, etc.). Dans la refonte de l'application, l'administrateur devra pouvoir gérer les utilisateurs, les commandes, les objets (articles), les conditionnements, les poids, et le poids des vignettes.
- *Opérateur colis* : L'opérateur colis traitait manuellement les colis réceptionnés et entrait les informations dans l'application Access. Dans la refonte, il devra pouvoir gérer les commandes, le mailing, les statistiques, créer des fiches clients et accéder à une interface d'impression.
- *Opérateur stock* : Il était responsable de la gestion des stocks de goodies et des inventaires. Lors de la refonte, il devra pouvoir en priorité gérer les stocks et dans une seconde version saisir les inventaires, accéder à une interface d'impression.

Ces rôles pouvant être cumulable, ce cas d'utilisation devra être pris en compte lors de la connexion à l'application. Une analyse des packages est disponible en Annexe 2.

DIGICHEESE fait donc appel à notre entreprise afin de réaliser une refonte complète d'une application de gestion en prenant en compte les problèmes précédemment cités mais également pour répondre au besoin de maintenance et d'ergonomie dans l'application. Il est à noter que DigiCheese a demandé que la gestion des colis et du profil Administrateur soient développés en priorité. Afin de répondre à cette problématique nous faisons le choix de consacrer plus de temps de développement à ces deux rôles et nous avons ainsi revu le calendrier et les coûts afin de respecter les délais impartis et le budget.

Pour répondre à cette problématique, nous choisissons de répartir la conception de l'application en différentes étapes qui permettrons de déterminer les fonctionnalités à développer.

La première étape vise à réaliser une modélisation fonctionnelle des besoins avec UML qui contribuera à reprendre les fonctionnalités existantes en l'état et de poser les bases des futurs besoins, cela inclut des étapes intermédiaires décrites ci-après :

- Déterminer les acteurs et de décrire leurs rôles respectifs.
- Réaliser un schéma global de l'architecture du système (acteurs et grandes fonctionnalités) avec les interactions des acteurs.
- Représenter les acteurs et le système sous forme de diagrammes de cas d'usage
- Décrire le scénario de la gestion des commandes
- Créer le diagramme de classe concernant la partie de gestion des colis qui nous servira de base pour créer ensuite la base de données.

La seconde étape vise à développer les principales fonctionnalités demandées. Celles-ci regroupent :

- Fonctionnalité – Gestion des Clients
- Fonctionnalité - Gestion des Commandes
- Fonctionnalité - Gestion des Objets
- Fonctionnalité – Gestion des Utilisateurs

La dernière étape visera à réaliser le frontend de l'application en optimisant l'expérience utilisateur.

2 – Nos objectifs :

L'objectif est donc de migrer vers une solution de type client léger (intranet par exemple), plus souple, facilement évolutive et maintenable. Il est important de noter qu'un aspect important de cette refonte sera porté sur la fluidité de la navigation dans l'application.

Nos objectifs sont donc centrés sur :

- **OBJECTIF 1** : Réaliser une modélisation fonctionnelle des besoins avec UML
- **OBJECTIF 2** : Réaliser le backend de l'application (connexion à la base de données, fonctionnalité demandée pour créer, modifié, supprimer ou mettre à jour les fiches clients, utilisateur, objet ou commandes)
- **OBJECTIF 3** : Réaliser le frontend en mettant l'accent sur les visuels afin de permettre une interaction optimale entre l'utilisateur et l'application.

3 – Résumé des exigences techniques

Les exigences techniques à prendre en compte regroupent :

- **Exigences de performance** : Demande d'un client léger, permettant une navigabilité optimale.
- **Exigences Ergonomique** : l'accent devra être porté sur la souplesse et la fluidité lors de la navigation dans l'application. Facilité d'accès avec un paramétrage selon le rôle (administrateur, opérateur colis et/ou stock.
- **Exigences d'extensibilité et de maintenance** Utilisation d'outils moderne afin de faciliter les futures évolutions. Maintenance facilitée de l'application
- **Exigences de sécurité** : Sécurité des données relatives aux données des clients et utilisateurs (en adéquation avec la RGPD).

IV – Description fonctionnelle des besoins

1 – Résumé de la solution proposée

La solution proposée se basera sur le langage de programmation Python, choisi pour sa polyvalence, notamment dans l'automatisation d'interactions avec les navigateurs et interfaces graphiques. Au niveau des frameworks, nous avons opté pour un frontend en JavaScript, Backend avec Flask API (programmé en Python). Flask est un micro-framework léger et extensible, idéal pour la performance et l'ergonomie. Les bases de donnes seront de type relationnelle (SQL) et nous utiliserons SQLAlchemy qui est une bibliothèque facilitant la communication entre l'API et la base de données en convertissant les appels en instructions SQL. Au niveau de l'infrastructure, nous proposons à DIGICHEESE d'utiliser Microsoft Access ou de migrer vers une infrastructure cloud (Azure, AWS ou Google) pour améliorer la sécurité et la gestion des ressources.

2 – Environnement informatique choisi, architecture logicielle, protocole de sécurité

a. Environnement informatique technique

Pour répondre à l'objectif de refonte de l'application, nous proposons un environnement informatique présentant les caractéristiques suivantes :

- **Langage de programmation :** Python. La force de ce langage réside dans la polyvalence. Il permet notamment d'automatiser les interactions avec les navigateurs web ou les interfaces graphiques des applications.

- **Frameworks**: frontend (JavaScript), backend (Flask API, Python).

JavaScript est un langage de programmation universel (dans le sens supporté par tous les navigateurs web), un écosystème riche avec des bibliothèques comme bootstrap qui fournit des composants d'interface déjà stylisée et permet ainsi un gain de temps dans la création d'une interface attrayante et fonctionnelle.
L'API Flask est un micro-framework qui constituera le noyau de notre application web. Celle-ci présente l'avantage d'être légère et extensible ce qui pourra permettre de faire évoluer les fonctionnalités par la suite. Dans ce contexte, cette API pourra répondre au besoin de performance et de recherche d'ergonomie.

- **Base de données :** Relationnelle (SQL) et utilisation de SQLAlchemy. SQLAlchemy est une bibliothèque Python qui établit une liaison entre le langage de programmation et les bases de

données SQL. Elle convertit automatiquement les appels de classes Python en instructions SQL, facilitant ainsi la communication entre notre API et la base de données.

- **Infrastructure** : l'infrastructure interne au sein de DIGICHEESE est associée avec une utilisation de Microsoft Access. Nous pouvons proposer une infrastructure cloud géré soit par Microsoft (Azure) selon les licences déjà en court au sein de DIGICHEESE ou alors auprès d'autre service (AWS, Google). L'infrastructure cloud permettra de gérer à la fois les éléments relatifs à la sécurité mais également à la gestion des ressources.

Au niveau des outils de développements :

- **Environnement de développement intégré (IDE) :** VS Code, JetBrains
- **Outils de gestions des versions :** GitHub
- **Environnement de test :** test unitaires (pytest), Swagger (vérification des routes de l'API), MyPy (tests statiques pour vérifier les types dans le code).
- **Intégration continue et déploiement continu (sur site) :** Tests unitaires et votre serveur Windows 2000. Nous prévoyons d'utiliser votre serveur et votre environnement configuré avec Windows 2000 dans lequel l'application sera développée et testée.
- **Serveur SQL :** MariaDB (gratuit) a des exigences matérielles très minimales et fonctionnera sur presque tout, mais vous devriez avoir au moins 8Go de RAM et 16Go est notre recommandation générale pour cette application.
- **Server Web :** Apache HTTP Server (gratuit) Le serveur HTTP Apache a des exigences matérielles très minimales et fonctionnera sur presque tout ce qui est équipé de Windows 2000 ou d'une version ultérieure, mais vous devez disposer d'au moins 8Go de RAM et 16Go est notre recommandation générale pour cette application.

En termes d'impact général, le développement des fonctionnalités demandé au travers de ces outils permettra de faciliter la maintenabilité et l'évolution du projet.

b. Architecture logicielle et protocole de sécurité

En ce qui concerne l'architecture logicielle, nous proposons une architecture incluant :

- **Modèle :** Architecture Microservice basée sur des API.
- **Interaction entre les composants** : API (Flask API, SQLAlchemy)
- **Sécurité** : communication sécurisée via HTTPS, chiffrement des données sensibles (données personnelles des clients), intégration de services tiers, par exemple pour l'authentification
- **Gestion des rôles :** gestions différenciées selon les droits et le profils octroyé (Administrateur, Opérateur colis ou stock)

Pour illustrer l'architecture proposée, la synthèse suivantes est proposée :

- **Frontend** : l'utilisateur interagit avec une interface web développée JS.
- **Backend :** cette interface envoie des requêtes à l'API développée avec **Flask API (Python)** pour traiter les demandes de l'utilisateur.
- **Accès aux données** : le backend envoie des requêtes à une **base de données MySQL** pour récupérer ou enregistrer des informations.
- **Sécurité** : l'authentification des utilisateurs est gérée via l'application et toutes les communications entre le client et le serveur sont chiffrées avec **HTTPS**.

3 -Spécification fonctionnelle

a. Résumé des spécification fonctionnelles.

En tant qu'employé de DIGICHEESE, je me connecte à l'application DIGICHEESE.com via un système d'authentification. Selon que SSO est activé dans l'entreprise, je m'authentifie à l'aide d'un système comme OpenID ou via le SSO. Mon niveau d'accès dépend des droits associés à mon compte (référencé en base de données). Selon mon rôle, j'accède aux fonctionnalités correspondantes :

→ En tant qu'Administrateur, je veux pouvoir
Gérer les utilisateurs de l'application (éditer, ajouter, modifier, supprimer).
Gérer les profils clients (éditer, ajouter, modifier, supprimer).
Gérer les commandes et les objets (éditer, ajouter, modifier, supprimer).
Gérer le conditionnement des objets et les règles associées aux calculs de conditionnement (éditer, ajouter, modifier, supprimer).
Accéder aux statistiques de l'application (éditer).

→ En tant qu'Opérateur colis, je veux pouvoir
Gérer les commandes et les clients (éditer, ajouter, modifier, supprimer).
Envoyer des emails aux clients (éditer, ajouter, modifier, supprimer).
Gérer les objets (éditer).
Calculer les coûts de conditionnement
Imprimer les bordereaux d'expédition

→ En tant qu'Opérateur stock, je veux pouvoir à terme
Gérer les stocks et les objets (éditer, ajouter, modifier, supprimer).
Consulter les commandes (éditer, éditer, ajouter, modifier, supprimer).
Imprimer les bordereaux d'expédition.

Afin de répondre à la demande de priorisation des développements liés à la gestion des colis et au rôle de l'Administrateur, nous avons décidé de concentrer nos efforts sur une seule fonctionnalité clé pour

l'Opérateur Stock : la gestion des stocks et des objets. Les fonctionnalités relatives aux bordereaux d'expédition et à la gestion des commandes étant également accessibles via l'Opérateur Colis, nous avons choisi de donner la priorité à cette approche. De cette façon, nous pourrons ainsi gagner un jour de développement et un jour de test (tests consacrés à la gestion des stocks).

Une fois les tâches terminées, les différents utilisateurs peuvent se déconnecter de l'application.

Une description visuelle des cas d'utilisation selon les rôles est disponible en annexe 3, pour l'Administrateur, en annexe 4 pour l'opérateur colis et en annexe 5 pour l'opérateur stock.

b. Fonctionnalités pour les personnes en situation de handicap

Nos interfaces utilisateur sont construites à l'aide des attributs et des rôles ARIA (Accessible Rich Internet Applications), conformément aux techniques des recommandations WCAG 2.0 du World Wide Web Consortium (W3C) pour rendre le contenu plus accessible, ce qui permet un meilleur accès aux personnes en situation de handicap. Cela permet également aux personnes souffrant de déficiences visuelles, telles que le daltonisme ou une faible acuité visuelle, d'appliquer leurs propres feuilles de style personnalisées (CSS) à l'application afin de mieux répondre à leurs besoins. En option, DIGI3 peut fournir une interface de paramétrage de l'application avec des options d'affichage qui appliqueraient automatiquement des mises en page et des palettes de couleurs pour répondre à vos besoins spécifiques. Nous devrions étendre la planification, le temps de développement et les coûts en fonction de ces besoins.

c. Principaux acteurs

Au sein du système de gestion des colis, trois acteurs sont impliqués : l'opérateur colis, l'administrateur système et l'opérateur stock. Une description de leur rôle est proposée dans le tableau suivant :

Acteurs	Rôles
Opérateur colis	Réceptionne et traite manuellement les courriers envoyés par les clients. Il gère les commandes, les clients, le conditionnement, l'affranchissement et l'envoi de mail (optionnel).
Administrateur système	Paramétrage du système.
Opérateur stock	Gestion des stocks et de l'inventaire

Tableau 1 : Acteurs et rôles respectifs.

4 – Accueil et connexion

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

Opérateur de colis : responsable de tout ce qui concerne la commande et le colis

Administrateur : responsable de la configuration du système et de la base de données

Opérateur de stock : responsable du stock

A noter que l'application respectera les recommandes de la **WCAG 2.0** : Web Content Accessibility Guidelines 2.0 visant à rendre le contenu du Web accessible pour les personnes en situation de handicap, qu'elles présentent des difficultés isolées ou combinées (malvoyance, surdité et perte d'audition, les troubles de l'apprentissage, les difficultés cognitives, la limitation des mouvements). Le respect de ces lignes directrices rendra aussi souvent votre contenu Web plus utilisable par les utilisateurs en général.

Une illustration des écrans d'accueil est disponible dans la figure 2 ci-dessous.

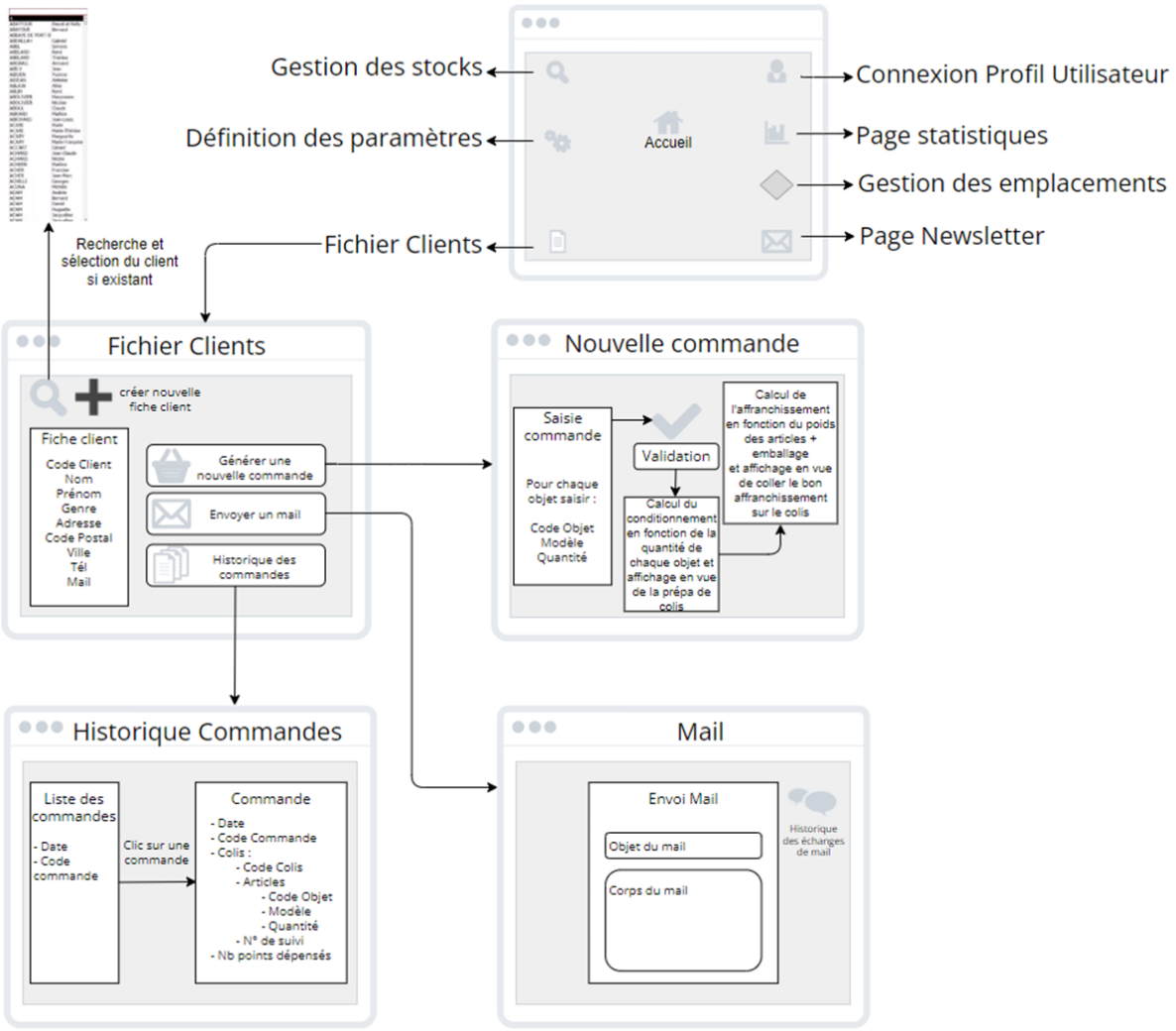


Figure 2 : Maquette illustrative

5 – Fonctionnalités

a. Fonctionnalité [Gestion des Commandes]

→ Sous fonctionnalité [Saisie de commande]

- **Objectif** : Saisir une commande
- **Description** : En tant qu'**Opérateur de colis**, je peux saisir une commande.

→ Sous fonctionnalité [Consultation des commandes passées]

- **Objectif** : Consulter l'historique de commande
- **Description** : En tant qu'**Opérateur de colis**, je peux consulter la liste de toutes les commandes passées et consulter le détail de chaque commande.

→ Sous fonctionnalité [Modification de commande]

- **Objectif** : Modifier une commande
- **Description** : En tant qu'**Opérateur de colis**, je peux modifier une commande saisie.

→ Sous fonctionnalité [Suppression de commande]
- **Objectif** : Supprimer une commande
- **Description** : En tant qu'**Opérateur de colis**, je peux supprimer une commande saisie.

→ Sous fonctionnalité [Calcul du prix de la commande]
- **Objectif** : Calculer le prix d'une commande
- **Description** : En tant qu'**Opérateur de colis**, je peux calculer le prix d'une commande.

→ Sous fonctionnalité [Envoi d'un mailing pour la commande]
- **Objectif** : Envoi de mail
- **Description** : En tant qu'**Opérateur de colis**, je peux envoyer un mail au client concernant sa commande.

b. Fonctionnalité [Gestion des Clients]

→ Sous fonctionnalité [Consulter la liste des clients]
- **Objectif** : Consulter la liste des clients
- **Description :** En tant qu'**Opérateur de colis**, je peux afficher la liste des clients

→ Sous fonctionnalité [Création d'une fiche client]
- **Objectif** : Créer une fiche client
- **Description :** En tant qu'**Opérateur de colis**, je peux créer une fiche client et saisir les informations relatives au client.

→ Sous fonctionnalité [Consultation d'une fiche client]
- **Objectif** : Consulter une fiche client
- **Description :** En tant qu'**Opérateur de colis**, je peux consulter une fiche client

→ Sous fonctionnalité [Modification d'une fiche client]
- **Objectif** : Modifier une fiche client
- **Description :** En tant qu'**Opérateur de colis**, je peux modifier les informations contenues dans la fiche client et saisir de nouvelles informations relatives au client.

→ Sous fonctionnalité [Suppression d'une fiche client]
- **Objectif** : Supprimer une fiche client
- **Description :** En tant qu'**Opérateur de colis**, je peux supprimer une fiche client.

c. Fonctionnalité [Gestion des Colis]

→ Sous fonctionnalité [Consulter la liste des colis en cours]

- **Objectif** : Visualiser les colis en cours
- **Description :** En tant qu'**Opérateur de colis**, je peux afficher la liste des colis en cours de préparation

→ Sous fonctionnalité [Consulter l'historique des mouvements de colis]

- **Objectif** : Visualiser l'historique des mouvements de colis
- **Description :** En tant qu'**Opérateur de colis**, je peux visualiser tous les mouvements de colis passés

d. Fonctionnalité [Mailing]

→ Sous fonctionnalité [Envoi de mail]

- **Objectif** : Envoyer un mail au client
- **Description :** En tant qu'**Opérateur de colis**, je peux envoyer un mail personnalisé au client

→ Sous fonctionnalité [Envoi d'une newsletter]

- **Objectif** : Déclencher une newsletter
- **Description :** En tant qu'**Opérateur de colis**, je peux éditer une newsletter et l'envoyer à l'ensemble ou à une partie ciblée du fichier client.

e. Fonctionnalité [Statistiques]

Objectif : Extraire des statistiques

Description : En tant qu'**Opérateur de colis**, je peux obtenir des statistiques en fonction d'un intervalle de dates au format mois/année.

f. Fonctionnalité [Interface d'impression]

Objectif : Imprimer

Description :

- En tant qu'**Opérateur de colis**, je peux imprimer au format papier.

Développement futur (non priorisé dans le contexte du projet mais pouvant faire l'objet de futurs développements au besoin) :

- En tant qu'**Opérateur de stock**, je peux imprimer au format papier.

g. Fonctionnalité [Gestion des utilisateurs]

→ Sous fonctionnalité [Création d'un utilisateur]
- **Objectif** : Saisir un nouvel utilisateur
- **Description :** En tant qu'**Administrateur**, je peux créer un nouveau profil utilisateur et lui attribuer un ou plusieurs rôles
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Consulter une fiche utilisateur]
- **Objectif** : Consulter les informations d'un utilisateur
- **Description :** En tant qu'**Administrateur**, je peux consulter un profil utilisateur
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Modifier une fiche utilisateur]
- **Objectif** : Modifier les informations d'un utilisateur
- **Description :** En tant qu'**Administrateur**, je peux modifier un profil utilisateur
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Supprimer une fiche utilisateur]
- **Objectif** : Supprimer un utilisateur
- **Description :** En tant qu'**Administrateur**, je peux supprimer un profil utilisateur
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

h. Fonctionnalité [Gestion des Communes]

→ Sous fonctionnalité [Création d'une commune]
- **Objectif** : Saisir une nouvelle commune
- **Description :** En tant qu'**Administrateur**, je peux créer une nouvelle commune
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Consulter les communes]
- **Objectif** : Consulter la liste des communes
- **Description :** En tant qu'**Administrateur**, je peux consulter la liste des communes
- **Contraintes et règles de gestion :**

- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Modifier une commune]
- **Objectif** : Modifier les informations d'une commune
- **Description :** En tant qu'**Administrateur**, je peux modifier une commune
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Supprimer une commune]
- **Objectif** : Supprimer une commune
- **Description :** En tant qu'**Administrateur**, je peux supprimer une commune
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

i. Fonctionnalité [Gestion des Objets]

→ Sous fonctionnalité [Création d'un objet]
- **Objectif** : Saisir un nouvel objet
- **Description :** En tant qu'**Administrateur**, je peux créer une fiche objet et lui attribuer un prix en vignettes et en argent et un poids
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Consulter un objet]
- **Objectif** : Consulter les informations d'un objet
- **Description :** En tant qu'**Administrateur**, je peux consulter les informations d'un objet
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Modifier une fiche objet]
- **Objectif** : Modifier les informations d'un objet
- **Description :** En tant qu'**Administrateur**, je peux modifier une fiche objet, notamment son prix et son poids
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Supprimer une fiche objet]
- **Objectif** : Supprimer un objet
- **Description :** En tant qu'**Administrateur**, je peux supprimer une fiche objet

- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

j. Fonctionnalité [Gestion des Conditionnements]

→ Sous fonctionnalité [Consulter la liste des emballages]
- **Objectif** : Visualiser la liste de tous les emballages existants
- **Description :** En tant qu'**Administrateur**, je peux visualiser la liste de tous les emballages existants

→ Sous fonctionnalité [Création d'un nouveau conditionnement]
- **Objectif** : Saisir un nouveau conditionnement
- **Description :** En tant qu'**Administrateur**, je peux créer un nouveau conditionnement et lui attribuer une capacité en kg
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Consulter un conditionnement]
- **Objectif** : Consulter les informations d'un conditionnement
- **Description :** En tant qu'**Administrateur**, je peux consulter une fiche conditionnement
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Modifier un conditionnement]
- **Objectif** : Modifier les informations d'un conditionnement
- **Description :** En tant qu'**Administrateur**, je peux modifier une fiche conditionnement.
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Supprimer un conditionnement]
- **Objectif** : Supprimer un conditionnement
- **Description :** En tant qu'**Administrateur**, je peux supprimer une fiche conditionnement
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Calcul conditionnement]
- **Objectif** : Calculer un conditionnement
- **Description :** En tant qu'**Opérateur de colis**, je peux calculer quel conditionnement est nécessaire pour ma commande et son prix

- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

k. Fonctionnalité [Gestion des Relations poids/vignette/colis]

→ Sous fonctionnalité [Création d'une règle de relation poids/vignette/colis]

- **Objectif** : Saisir une nouvelle règle
- **Description :** En tant qu'**Administrateur**, je peux créer une nouvelle règle concernant la relation entre le poids des objets, leur prix et le conditionnement associé
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Consulter une règle de relation poids/vignette/colis]

- **Objectif** : Consulter les règles
- **Description :** En tant qu'**Administrateur**, je peux consulter les règles concernant la relation entre le poids des objets, leur prix et le conditionnement associé
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Modifier une règle]

- **Objectif** : Modifier les règles
- **Description :** En tant qu'**Administrateur**, je peux modifier les règles concernant la relation entre le poids des objets, leur prix et le conditionnement associé
- **Contraintes et règles de gestion :**
- **Niveau de priorité : Haut**

→ Sous fonctionnalité [Supprimer une règle]

- **Objectif** : Supprimer une règle
- **Description :** En tant qu'**Administrateur**, je peux supprimer une règle concernant la relation entre le poids des objets, leur prix et le conditionnement associé.

a. Fonctionnalité [Gestion des Stocks]

Objectif : Gérer efficacement les stocks
Description : En tant qu'**Opérateur de stock**, je peux consulter et mettre à jour les stocks.

b. Fonctionnalité [Gestion de l'inventaire]

Développement futur (non priorisé dans le contexte du projet mais pouvant faire l'objet de futurs développements au besoin)

Objectif : Enregistrer les inventaires
Description : En tant qu'**Opérateur de stock**, je peux saisir mes résultats d'inventaire.

→ Sous fonctionnalité [Consulter une fiche client]
- **Objectif** : Consulter les informations d'un client
- **Description :** En tant qu'**Opérateur de colis**, je peux consulter les informations d'un client via son numéro de client et accéder à son historique de commandes.

6 – Flux de donnée

Afin d'illustrer l'interaction entre les processus et les différentes étapes, un diagramme de flux a été réalisé.

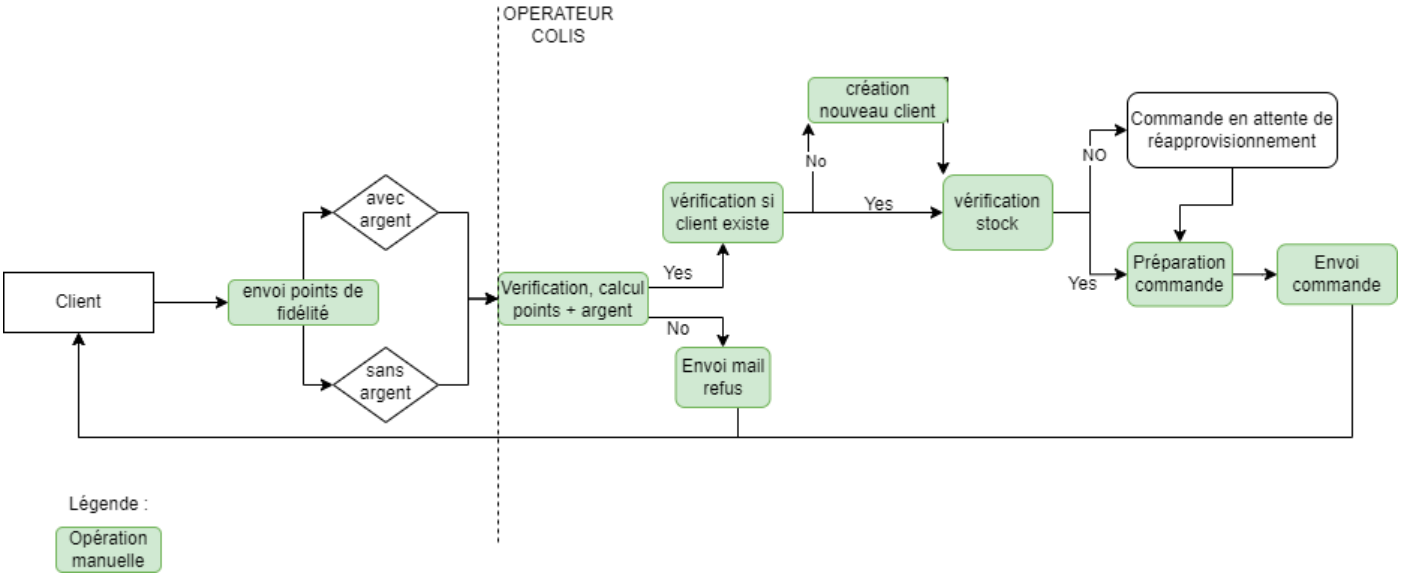


Figure 3 : Diagramme de flux.

7 – Contenu des bases de données

La base de données sera implémentée sous SQL avec une architecture relationnelle. Elle sera composée de plusieurs tables principales, reliées entre elles par des relations de type "une à plusieurs" ou "une à une". Voici un aperçu des principales tables et de leurs attributs :

- **Client** : Cette table rassemble les informations relatives aux clients. Chaque enregistrement contiendra des détails tels que le nom, le prénom, l'adresse, l'email, etc. Les données contenues dans cette table seront synchronisées avec le système externe Open Access pour assurer la cohérence des informations clients.

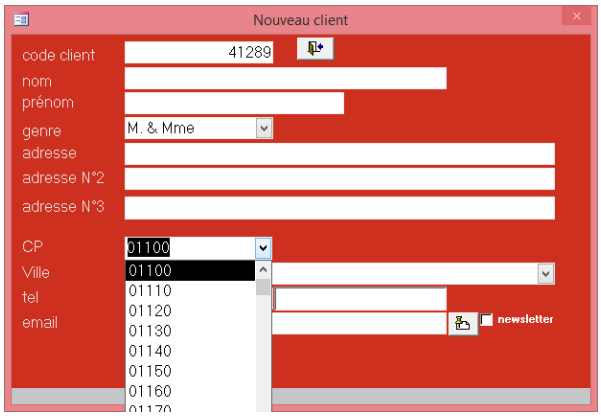


Figure 4 : Fiche client sous Microsoft Access.

- **Commande** : La table commande stocke les informations sur les commandes passées par les clients. Les champs incluent un identifiant unique pour chaque commande, la date de la commande, le nombre de colis, ainsi que des détails sur les produits commandés. Chaque commande est liée à un client via une relation "une à plusieurs" (un client peut passer plusieurs commandes).
- **Objet** : Cette table contient les informations relatives aux articles ou goodies disponibles pour les clients dans le cadre du programme de fidélité. Chaque article est défini par un identifiant unique, un nom, un poids, une disponibilité, etc. Les objets peuvent être liés aux commandes via une relation "une à plusieurs" (une commande peut inclure plusieurs objets).
- **Utilisateur** : Cette table gère les rôles et les informations des utilisateurs qui interagissent avec l'application. Chaque utilisateur a un identifiant unique, un nom, un email et un rôle (exemple : administrateur, opérateur). Cette table est cruciale pour la gestion des permissions et des accès.

En plus de ces tables principales, d'autres tables secondaires seront créées pour gérer des aspects spécifiques tels que le conditionnement (liens entre objets et leur packaging) et le stock (quantités disponibles pour chaque objet). Les relations entre ces tables et les principales suivront également des modèles de relations clairs (ex. : un objet est conditionné dans plusieurs types d'emballages, relation "une à plusieurs").

L'architecture de la base de données est présentée dans la figure ci-après.

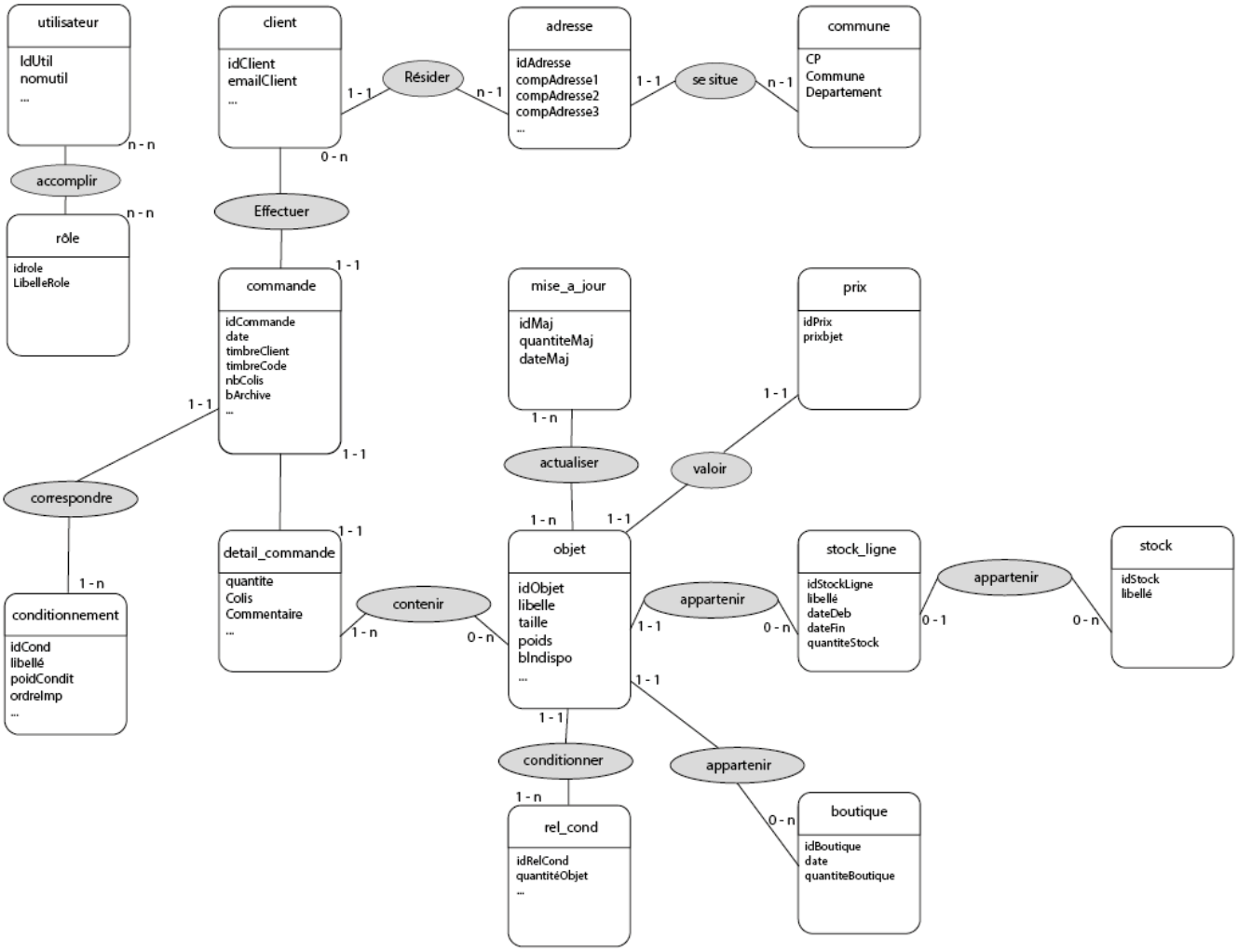


Figure 5 : Modèle de la base de données

En ce qui concerne les données clients, utilisateur le respect des standards de sécurité (chiffrement des données sensibles, respect des normes RGPD) sera respecté.

8 – Plan de développement logiciel

Afin de répondre aux objectifs et exigences techniques du projet, nous adopterons une **méthodologie Agile** avec la méthode **Scrum**. L'avantage de cette approche est qu'elle permettra une collaboration continue avec l'équipe de DIGICHEESE à chaque étape du développement, facilitant ainsi la validation progressive des livrables et la gestion des priorités. Le développement sera structuré autour de **sprints** d'une durée de deux semaines (à l'exception du Sprint 4, d'une durée d'une semaine, qui a pour but de garder des jours de développement en réserve selon les modifications souhaitées par le client). Les livrables seront proposés à la fin de chaque sprint.

Découpage des sprints :

Les sprints seront organisés comme suit :

- **Sprint 1** : Phase d'analyse des besoins et d'analyse fonctionnelle. Il s'agira de définir les spécifications techniques et fonctionnelles avec une validation du Product Backlog.
- **Sprint 2** : Conception de l'UX (maquettes) et début du développement du backend et du frontend.
- **Sprint 3** : Poursuite du développement du backend et du frontend, avec implémentation de l'API et des fonctionnalités majeures.
- **Sprint 4** : Phase de révision et rétrospective, correction des bugs et ajustements fonctionnels en fonction des retours des tests.
- **Sprint 5** : Déploiement final de l'application et formation des utilisateurs finaux à son utilisation.

Chaque sprint inclura une phase de tests réalisée par un testeur indépendant sur des journées spécifiques. Ces tests comprendront des tests unitaires, des tests d'intégration, de régression et des tests fonctionnels pour s'assurer que chaque fonctionnalité développée est conforme aux attentes. À la fin de chaque sprint, un COPIL sera organisé avec le client, le chef de projet, le Product Owner (PO), et le lead développeur, afin de valider le travail accompli et d'identifier les modifications nécessaires pour le sprint suivant.

Livrables intermédiaires : À partir du **Sprint 2**, des livrables partiels seront fournis toutes les deux semaines. Ces livrables incluront des versions fonctionnelles de certaines parties de l'application, permettant de valider progressivement l'avancement du projet. Une **livraison finale** sera effectuée à la fin du **Sprint 5**, après validation des dernières modifications et déploiement en production.

Environnement de développement :
Le projet utilisera plusieurs environnements :

- **Environnement de développement** : où les nouvelles fonctionnalités seront développées.
- **Environnement de test** : où les tests effectués.
- **Environnement de production** : pour le déploiement final.

La gestion du code source se fera via **GitHub**, avec une **intégration continue** afin d'assurer un déploiement fluide et régulier des versions testées. Une branche sera créée pour chaque développeur et l'ajout du développement à l'environnement de test se fera par une validation d'un autre développeur.

Sécurité et conformité :
Les aspects de sécurité seront intégrés dès le début du développement, en particulier :

- **Authentification et autorisation** : gestion des droits d'accès selon les rôles définis dans l'application.
- **Sécurisation des données** : respect des standards de sécurité (chiffrement des données sensibles, respect des normes RGPD).

9 – Tests

Afin de garantir la qualité de l'application, des tests seront réalisés régulièrement à la fin de chaque sprint par un testeur indépendant. Le recours à un testeur externe permet d'assurer une analyse objective et complète du développement effectué, évitant ainsi que les développeurs ne testent leurs propres développements, ce qui pourrait limiter la détection d'anomalies. Cette approche permet d'identifier rapidement les problèmes, d'améliorer la qualité du code, et d'assurer un retour constant sur les fonctionnalités livrées.

Les types de tests effectués incluront :

- Tests unitaires : Pour valider chaque fonctionnalité de manière isolée.
- Tests d'intégration : Pour vérifier que les différentes fonctionnalités fonctionnent bien ensemble.
- Tests fonctionnels : Pour s'assurer que l'application se comporte conformément aux spécifications fonctionnelles.
- Tests de non-régression : Pour garantir que les nouvelles fonctionnalités n'ont pas impacté les fonctionnalités existantes.
- Tests de performance : Pour évaluer la réactivité et la stabilité de l'application sous charge.
- Tests de montée en charge : Pour tester la capacité de l'application à supporter un grand nombre d'utilisateurs simultanés sans perte de performance.

Les retours issus de ces tests seront analysés et, en cas de détection d'anomalies ou d'améliorations nécessaires, des tickets de correction seront créés et priorisés pour le sprint suivant. Cela garantit que les problèmes soient résolus rapidement et que les livraisons sont conformes aux attentes.

10 – Alertes

En ce qui concerne les alertes, un système de surveillance continu sera mis en place pour détecter et signaler les anomalies critiques et mineures dans l'application. Ces alertes seront classées selon leur **niveau de priorité** (bloquant, critique, mineur), afin de permettre une réaction rapide et appropriée en fonction de la gravité du problème.

Les alertes seront gérées comme suit :

Priorité	Type Alerte	Détail	Destinataire
Bloquant	Alerte serveur	Erreurs de connexion à la base de données (temps >5 minutes	Admin (notification mail)
Critique	Alerte de performance	Latence serveur excessive (> 2 secondes pour une requête	Admin (notification mail)

Mineur	Alerte utilisateur	Problèmes de connexion des utilisateurs	Admin
Mineur	Exceptions non gérées	Erreurs non capturées dans le code	Admin
Critique	Anomalies signalées par les utilisateurs (bug)	Bugs rapportés via l'interface ou par remontée manuelle	Admin, équipe de développement

Chaque alerte sera associée à un délai de traitement en fonction de sa priorité :

- **Alerte bloquante** : Doit être résolue dans un délai d'1 heure maximum.
- **Alerte critique** : Résolution attendue sous 4 heures.
- **Alerte mineure** : Résolution sous 24 heures.

Les notifications d'alerte seront transmises par email ou via un outil de messagerie (ex. : Microsoft Outlook, Gmail), garantissant que l'équipe concernée soit immédiatement informée et puisse intervenir rapidement. Les administrateurs seront principalement responsables de la gestion des alertes serveurs, tandis que les anomalies liées aux utilisateurs ou aux performances seront partagées avec l'équipe de développement pour un suivi et une résolution adéquate.

V – Sécurité

Digi3 fournira un service d'autorisation et d'authentification des utilisateurs avec un contrôle d'accès basé sur les rôles sur l'un de vos serveurs locaux hébergeant les logiciels MariaDB SQL et Apache HTTP Server, ces services ne seront pas aussi robustes ni aussi redondants que les services d'un fournisseur de cloud, tels que Amazon Web Services (AWS), Microsoft Azure, ou Google Cloud Platform (GCP). En outre, votre serveur autonome sera moins souple pour s'adapter à une augmentation saisonnière soudaine des commandes des clients ou à une croissance rapide due, par exemple, à une campagne de marketing remarquable que vous pourriez mener. (Voir l'annexe 7 pour une discussion sur les avantages optionnels de l'utilisation d'un fournisseur de cloud public pour l'hébergement de votre base de données et de votre application front).

Pour ce service auto-hébergé, Digi3 configurera l'autorisation de travailler avec un nom d'utilisateur standard et un processus de mot de passe qui sera géré par toute personne ayant le rôle d'utilisateur administratif. La communication entre le frontend et le backend utilisera le protocole HTTPS et un certificat de sécurité SSL qui ***devra être fourni par DIGICHEESE***. Compte tenu de ce niveau de sécurité relativement faible, ***L'administrateur du réseau de DIGICHEESE devra s'assurer que le serveur intranet est bien isolé*** de l'Internet et qu'un pare-feu avec filtrage IP est en place pour interdire l'exposition de l'application à tout type d'accès provenant de l'extérieur du réseau interne de DIGICHEESE. Nous suggérons fortement de mettre en place un air-gapping sur le réseau si cela est possible.

En ce qui concerne l'autorisation des utilisateurs, Digi3 configurera les privilèges d'accès à l'application et à la base de données en fonction de chacun de vos trois rôles d'utilisateur : **administration** (admin), **gestion des expéditions** (OP-colis) et **gestion des stocks** (OP-stocks). Chacun de ces trois rôles se verra accorder les accès et privilèges suivants que DIGICHEESE a demandés.

VI – Budget

Tâches	Intervenants	Coût / J	Nb Jours Homme	Total	Commentaires
Analyse des besoins			**3**	**2 400€**	
Réunions de cadrage	Chef de projet, Lead Dev, PO, utilisateurs finaux	800€	1	800€	Recueillir les besoins des utilisateurs finaux
Analyse Fonctionnelle	Chef de projet, Lead Dev, PO	800€	2	1 600€	Écrire les spécifications fonctionnelles
Conception Fonctionnelle et Technique			**7**	**4 200€**	
Définition des fonctionnalités	Chef de Projet, Lead Dev	600€	1	600€	Définir les fonctionnalités minimales pour une première version
Conception de l'architecture	Chef de Projet, Lead Dev	600€	2	1 200€	Concevoir l'architecture technique
Rédaction des documents	Chef de Projet, Lead Dev	600€	4	2 400€	Écriture du CDC & Note de cadrage
Design UX/UI			**5**	**2 000€**	
Conception	Designer UI/UX	400€	4	1 600€	Design de l'interface utilisateur, Conception des écrans
Validation	Chef de Projet	400€	1	400€	Validation des maquettes
Développement			**34**	**25 200€**	Développer les fonctionnalités de l'application
Backend	Dév Backend	800€	19	15 200€	Gestion des données, API, Algo de calcul des coûts
Frontend	Dév Frontend	800€	10	8 000€	Interface utilisateur, écrans de suivi des commandes et de gestion des stocks
Supervision	Chef de Projet	400€	5	2 000€	Suivi du développement, Supervision
Tests et Validation			**5**	**2 000€**	
Tests	Testeur	400€	4	1 600€	Tests unitaires, Tests d'intégration, Validation des utilisateurs

Coordination	Chef de projet	400€	1	400€	Coordination des tests
Déploiement			**11**	**6 500€**	
Déploiement	Lead Dev	600€	4	2 400€	Déploiement sur infra client
Configuration des utilisateurs	Lead Dev	600€	1	600€	Configuration des principaux utilisateurs
Formation	Lead Dev	900€	1	900€	Formation des utilisateurs à l'outil
Suivi de fonctionnement	Lead Dev	600€	3	1 800€	Assurance du bon fonctionnement de l'outil
Supervision	Chef de projet	400€	2	800€	Suivi du déploiement, Supervision
Comités de Pilotage	Chef de projet, Lead Dev, PO	800€	**3**	**2 400€**	Réunions de suivi après chaque grande étape
(Maintenance Facultative)			**(6)**	**(2 400€)**	Assurer une maintenance corrective sur la première année
Interventions techniques	Développeur	400€	5	2 000€	Interventions correctives et évolutives
Suivi	Chef de projet	400€	1	400€	Suivi des demandes et des correctifs
Total			**68 jrs h**	**44 700€**	**(+ 2400 facultatifs)**

À noter que le budget n'inclut pas les frais de déplacement qui seront estimés à part.

VII – Calendrier

En ce qui concerne le calendrier, nous proposons un découpage global comme suit :

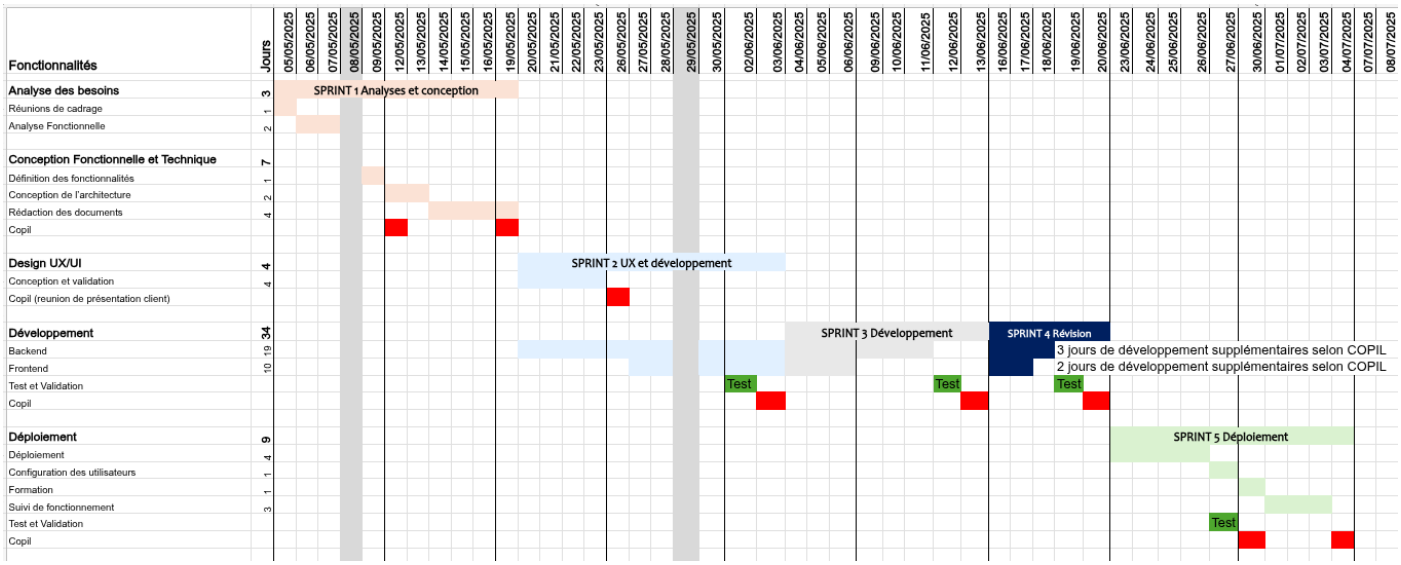


Figure 6 : Calendrier modifié selon les demandes de DIGICHEESE.

Cette version modifiée du calendrier prend en compte les demandes de DIGICHEESE, à savoir :

- Livraison décalée de 4 mois par rapport au planning initial : afin de répondre à ce critère, le projet débutera donc le 05/05/2025 (au lieu du 06/01/2025) pour s'achever au plus tard le 04/07/2025.
- Livraison sur site en 45 jours maximum : livraison prévue en 43 jours sur site.
- Budget de 45 K € HT : budget estimé à 44 700 € HT.
- La Gestion des colis et du profil "Administrateur" doivent être développés en priorité.

Afin de respecter le planning et le budget, le développement sera axé en priorité sur les fonctionnalités nécessaires à la gestion des colis et le profil Administrateur. Comme énoncé dans la partie IV-3 (page 14), afin de répondre à la demande de priorisation des développements liés à la gestion des colis et au rôle de l'Administrateur, nous avons décidé de concentrer nos efforts sur une seule fonctionnalité clé pour l'Opérateur Stock : la gestion des stocks et des objets. Les fonctionnalités relatives aux bordereaux d'expédition et à la gestion des commandes étant également accessibles via l'Opérateur Colis, nous avons choisi de donner la priorité à la fonctionnalité de gestion des stocks de l'opérateur stock. De cette façon, nous pourrons ainsi gagner un jour de développement et un jour de test (jour qui était censé être consacré aux tests de la gestion du stock) lors du Sprint 3. Également, le développement étant réalisé sur site, nous pourrons gagner une journée de déploiement lors du Sprint 5.

Au niveau du Backlog, nous proposons un premier découpage des Sprints de ce type (ce découpage pourra bien sûr être modifié lors du premier COPIL qui permettra de valider le Product Backlog) :

Figure 7 : Composition du Sprint 1

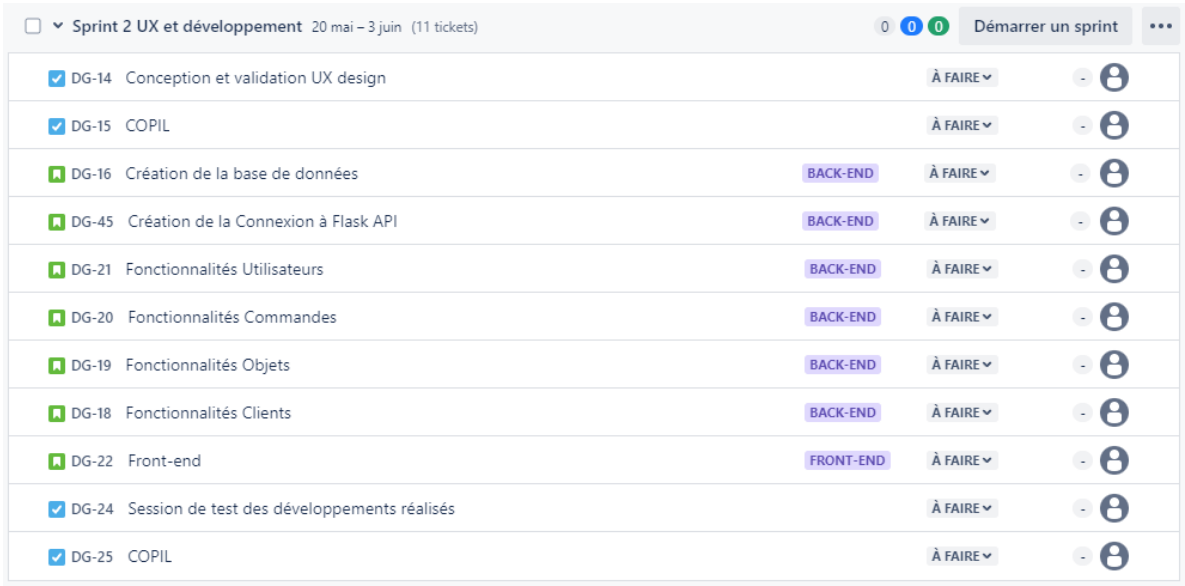


Figure 8 : Composition du Sprint 2

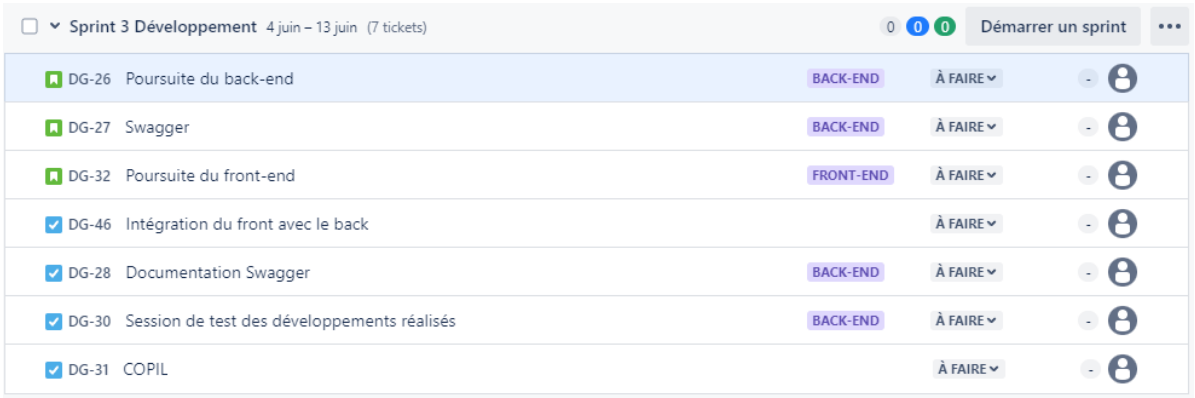


Figure 9 : Composition du Sprint 3

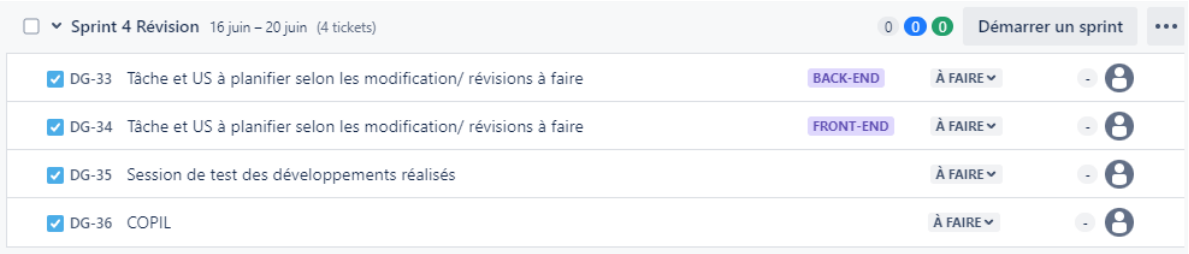


Figure 10 : Composition du Sprint 4 (servant à apporter les modifications et révisions nécessaires selon le souhait du client)

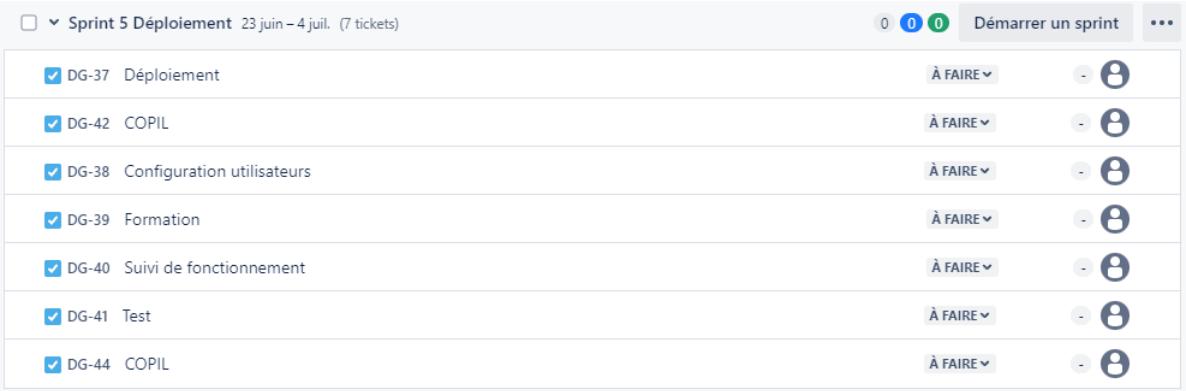


Figure 11 : Composition du Sprint 5

VIII – Conclusion et remerciement

En conclusion, ce cahier des charges technique formalise les exigences essentielles pour la réussite du projet de refonte de l'application DIGICHEESE.com. Il a pour but d'assurer une vision partagée tout au long du développement. Nous invitons les parties prenantes à valider ce document afin de pouvoir entamer les prochaines étapes du projet dans les meilleures conditions.

Nous tenons à remercier DIGICHEESE de la confiance qu'il nous témoigne en faisant appel à notre équipe. Nous remercions également toutes parties prenantes pour leur implication dans la définition de ce projet.

IX – Clauses

Clause d'acceptation :

Je soussigné(e), ... ; (nom du représentant de DIGICHEESE) agissant en tant que représentant autorisé de DIGICHEESE, déclare avoir pris connaissance et accepte les termes et conditions du présent cahier des charges.

Responsabilité des Parties :

En signant ce document, DIGICHEESE et Digi3 reconnaissent que les informations fournies sont exactes et s'engagent à respecter les exigences définies. Toute modification ultérieure pourra faire l'objet d'un avenant nécessitant une acceptation écrite des deux parties.

Clause d'Engagement et de Conformité :

DIGICHEESE, s'engage à fournir toutes les informations nécessaires à la bonne réalisation de ce projet et à valider les livrables selon les termes précisés dans le présent document.

Fait à ..... le .....

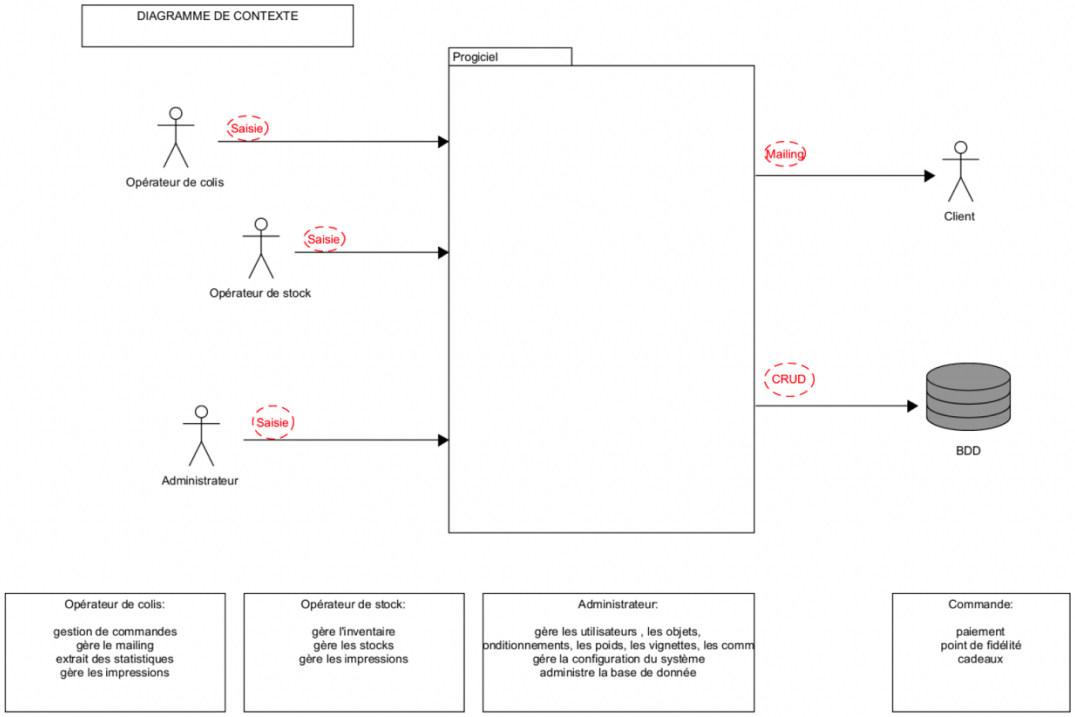
Signature du représentant de DIGICHEESE
Nom et fonction du signataire

Fait à ..... le .....

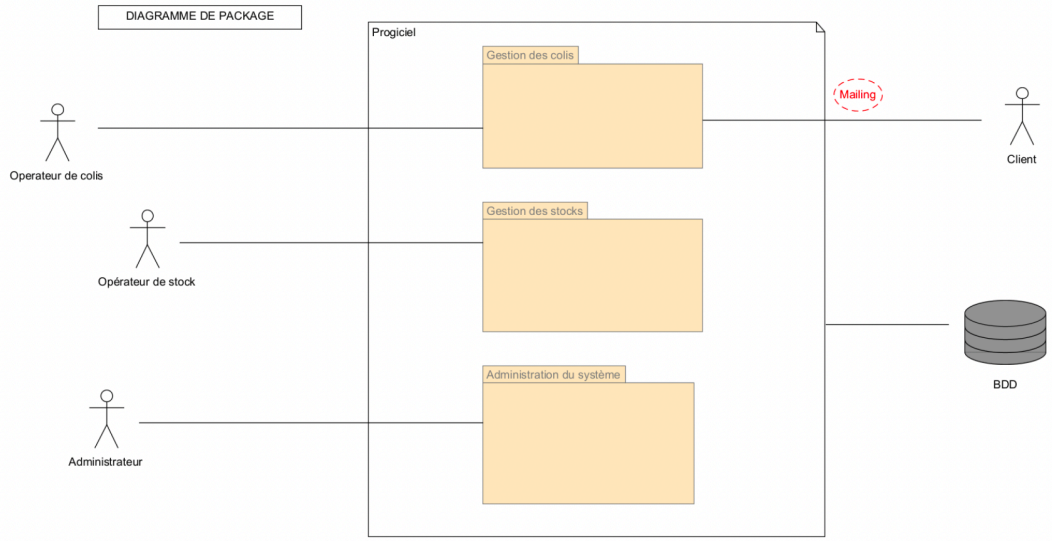
Signature du représentant de Digi3
Nom et fonction du signataire

X – Annexes

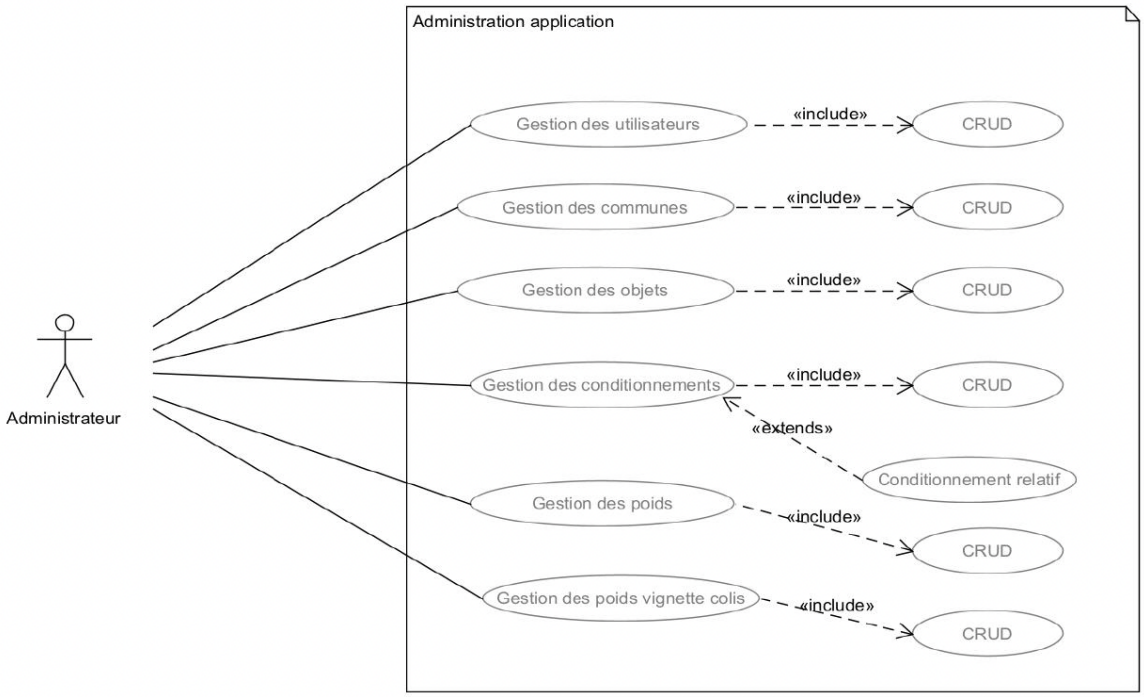
Annexe 1 – Diagramme du contexte général.



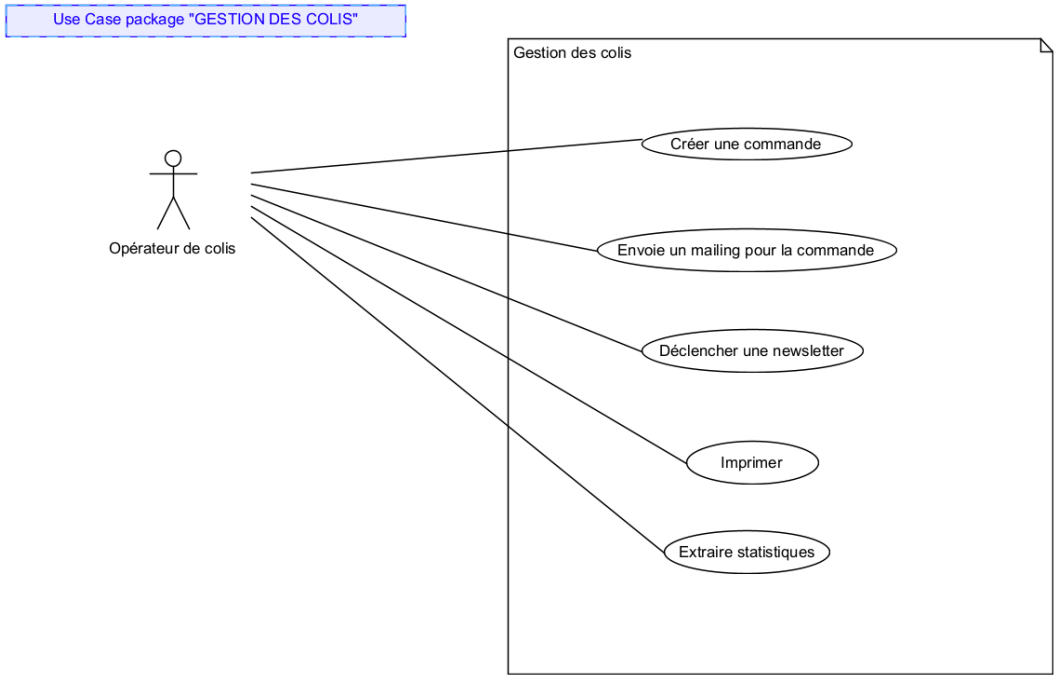
Annexe 2 – Diagramme de package, dépendant des rôles utilisateurs.



Annexe 3 – Diagramme de Use Case, rôle Administrateur



Annexe 4 – Diagramme de Use Case, rôle Opérateur Colis.



Annexe 5 – Diagramme de Use Case, rôle Opérateur Stock.