

|  |
| --- |
| DIGIDEV  Refonte de l’application et de la base de données existante de la fromagerie DIGICHEESE  **Cahier des charges**  *Rédigé le 18/12/2024 par Youssef Berhayla et Matthieu Roignant*  *Dernière mise à jour : 20/12/2024* |

[I. Introduction 3](#_Toc185583367)

[1. Contexte et enjeux du projet 3](#_Toc185583368)

[2. Objectifs de la refonte 4](#_Toc185583369)

[II. Parties prenantes 5](#_Toc185583370)

[1. DIGICHEESE 5](#_Toc185583371)

[2. DIGIDEV 6](#_Toc185583372)

[3. Organigramme 7](#_Toc185583373)

[III. Description fonctionnelle des besoins 7](#_Toc185583374)

[1. Accueil et connexion 7](#_Toc185583375)

[2. Fonctionnalités 9](#_Toc185583376)

[❖ **Partie Administrateur** (Figure 4) 9](#_Toc185583377)

[❖ **Gestion des colis** (Figure 6) 10](#_Toc185583378)

[❖ **Gestion des stocks** (Figure 8) 12](#_Toc185583379)

[IV. Architecture technique 13](#_Toc185583380)

[1. Choix de l’Environnement Informatique 13](#_Toc185583381)

[❖ **Frontend** 13](#_Toc185583382)

[❖ **Backend** 14](#_Toc185583383)

[❖ **Base de données** 14](#_Toc185583384)

[2. Serveur et hébergement 14](#_Toc185583385)

[3. Sécurité 14](#_Toc185583386)

[V. Modélisation des bases de données 15](#_Toc185583387)

[VI. Budget 17](#_Toc185583388)

[VII. Suivi du projet 17](#_Toc185583389)

[IX. Calendrier 18](#_Toc185583390)

[X. Annexes 19](#_Toc185583391)

[❖ **Structure des tables des bases de données** 19](#_Toc185583392)

## Introduction

#### Contexte et enjeux du projet

Avec plus d’un siècle d’existence, la fromagerie DIGICHEESE est une entreprise familiale composée d’environ 130 salariés. Elle vend ses produits exclusivement aux grands distributeurs et aux clients particuliers qui se rendent physiquement dans la boutique située proche de l’usine de production.

Actuellement, et depuis plus de 20 ans, DIGICHEESE utilise une ancienne application réalisée sur Microsoft Access 2000 et implémentée en VBA. C’est une technologie devenue obsolète qui n’est plus prise en charge depuis Windows Vista et ne bénéficie plus de dernières mises à jour depuis juillet 2009, empêchant toute modernisation ou amélioration de l’application.

DIGICHEESE rencontre ainsi plusieurs problèmes :

* Forte instabilité de l’application entrainant de nombreux bugs
* Problèmes de maintenance dus à l’arrêt des mises à jour
* Faible possibilité d’évolution de développement à la suite de la fin d’exploitation du système Access 2000 par Microsoft
* Manque de fluidité à cause du client lourd et peu performant
* Manque d’accessibilité et de visibilité pour les utilisateurs à cause d’une interface dépassée dans le temps

Une refonte de l’application est nécessaire afin d’apporter des évolutions visuelles, de nouvelles fonctionnalités et d’évoluer vers les technologies actuelles.

#### Objectifs de la refonte

La fromagerie DIGICHEESE a besoin d’une nouvelle application et de mettre à niveau les bases de données existantes.

L’objectif est de réaliser un site interne à l’entreprise, destiné aux salariés (Figure 1). Ce site se divisera en quatre parties :

* Authentification : permettant la connexion à son profil utilisateur
* Administration : accessible uniquement pour les rôles « Admin »
* Gestion des colis : accessible pour les rôles « OP-colis »
* Gestion des stocks : accessible pour les rôles « OP-stock »

Une image contenant texte, diagramme, capture d’écran, Plan

Description générée automatiquement

Figure 1 – Diagramme de contexte

Cette refonte s’inscrit dans la volonté de modernisation de l’entreprise. La nouvelle application devra ainsi avec une ergonomie plus moderne avec l’utilisation de technologie récente et permettant, par la suite, l’introduction de nouvelles fonctionnalités.

## Parties prenantes

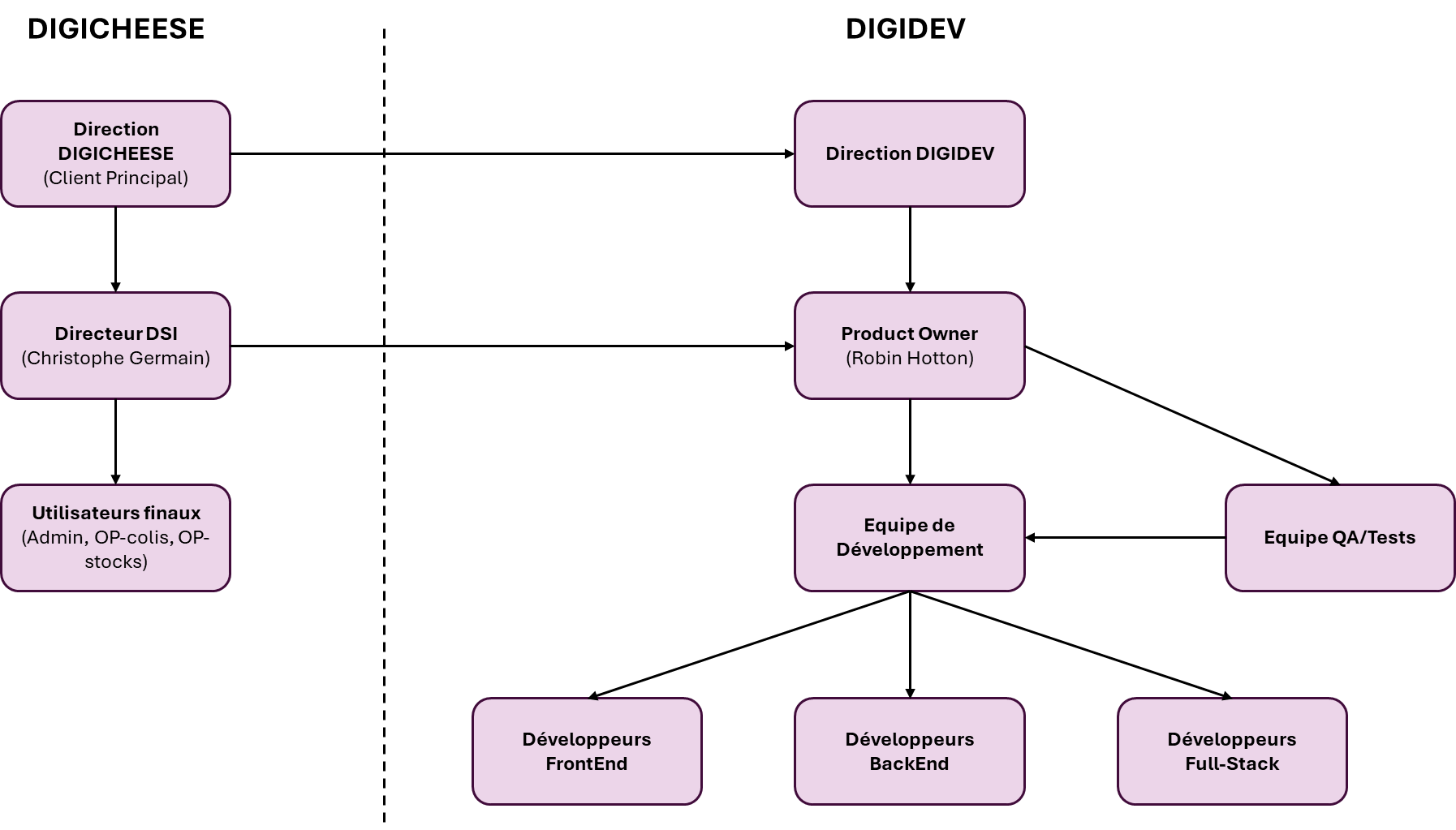
#### DIGICHEESE

|  |  |  |
| --- | --- | --- |
| **DIGICHEESE** | | |
| **Nom** | **Rôle** | **Responsabilités** |
| **Fromagerie DIGICHEESE** | Commanditaire du projet | * Définir les besoins et les attentes en matière de fonctionnalités et de performance * Valider les différentes étapes du projet (spécifications, prototypes, livrables finaux) * Participer aux phases de test * Fournir les ressources nécessaires (données, accès, etc.) |
| **Christophe Germain** | Directeur DSI de DIGICHEESE | * Superviser le projet de refonte au nom de  DIGICHEESE * Gérer la relation avec le sous-traitant **DIGIDEV** * S’assurer que le projet respecte les exigences techniques et les objectifs stratégiques * Coordonner avec les utilisateurs finaux pour s’assurer de l’adéquation de l’application avec leurs besoins |
| **Utilisateurs Finaux** | Utilisateurs de l’application au sein de DIGICHEESE | * **Administrateurs (Admin)** : Gérer les utilisateurs et les paramètres de l’application * **Opérateurs Colis (OP-colis)** : Créer et suivre les commandes de goodies * **Opérateurs Stocks (OP-stocks)** : Gérer les stocks de goodies et effectuer les inventaires * Fournir des retours sur l’application pour améliorer les fonctionnalités et l’ergonomie |

#### DIGIDEV

|  |  |  |
| --- | --- | --- |
| **DIGIDEV** | | |
| **Nom** | **Rôle** | **Responsabilités** |
| **DIGIDEV** | Entreprise sous-traitante chargée du développement de l’application | * Fournir les ressources techniques nécessaires pour développer l’application * Assurer la qualité et les performances de l’application * Livrer le projet dans le respect des délais et du budget convenus avec DIGICHEESE |
| **Robin Hotton** | Product Owner | * Recueillir et prioriser les besoins des utilisateurs finaux * Élaborer et gérer le Backlog du produit * Communiquer régulièrement avec le DSI de  DIGICHEESE pour s'assurer de la satisfaction du client * Valider les livrables à chaque étape |
| **Équipe de Développement** | Équipe technique en charge du développement de l’application | * Implémenter les fonctionnalités conformément aux spécifications * Assurer la qualité du code et respecter les bonnes pratiques de développement * Corriger les anomalies identifiées pendant les phases de test |
| **Équipe QA/Tests** | Assurer la qualité de l’application | * Rédiger et exécuter les plans de test * Identifier et documenter les anomalies * Valider les corrections avant la mise en production |

#### Organigramme



## Description fonctionnelle des besoins

#### Accueil et connexion

La nouvelle application ne sera accessible que via le réseau interne de l’entreprise. Aucun contenu ou fonctionnalité ne sera accessible hors authentification.

Plusieurs rôles seront définis pour accéder à l’application (Figure 2) :

* **Rôle 1** : Administrateur
* **Rôle 2** : OP-colis
* **Rôle 3** : OP-stock

Une image contenant texte, diagramme, Plan, Dessin technique

Description générée automatiquement

Figure 2 – Diagramme de package

Ces rôles pourront être cumulatifs. Par exemple, un « Admin » pourra également être « OP‑colis » ou « OP-stock.

Sur l’écran d’accueil, il faudra rentrer son identifiant ou mail d’entreprise et se connecter avec un mot de passe (Figure 3). Si ce dernier est oublié, un lien permettra de redéfinir ce mot de passe. Un lien « Contacter l’administrateur » permettra de rapporter tout problème survenu lors de la connexion.



Figure 3 – Maquette écran d’accueil de l’application

#### Fonctionnalités

##### **Partie Administrateur** (Figure 4)

* Gestion des opérations du CRUD (ajout, modification, suppression) sur les utilisateurs et les bases de données.
* Modification les bases de données (communes, objets, conditionnements, poids, poids-vignette).
* Possibilité d’impression au format papier.

Une image contenant texte, diagramme, capture d’écran, motif

Description générée automatiquement

Figure 4 – Use case administrateur

Sur cet écran, les administrateurs auront la possibilité de créer, modifier, supprimer et consulter les bases de données et les utilisateurs (Figure 5). Pour la partie « UTILISATEUR », le bouton « Création » renverra vers un formulaire à compléter. Le bouton « Modificateur » permettra de rentrer le nom d’un utilisateur pour modifier ses informations et ses accréditations. Le bouton « Supprimer » sera rediriger vers une barre de recherche et une liste pour sélectionner l’utilisateur à supprimer. Les administrateurs auront accès à une liste de tous les utilisateurs en cliquant sur « Liste utilisateurs » regroupant toutes les informations de connexion et pourront également imprimer cette liste.

Des fonctionnalités similaires seront utilisées pour la partie « BASE DE DONNEES ». Un bouton « STATISTIQUES » pourrait être mis en place pour analyser les datas des connexions des utilisateurs.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Figure 5 – Maquette écran administrateur

##### **Gestion des colis** (Figure 6)

* Gestion des clients, des commandes et des conditionnements.
* Accès à la liste des clients, aux commandes en cours, à diverses listes (emballages, relation poids/colis, relation poids-vignette/colis) et à différentes statistiques.
* Possibilité de contacter les clients par mail.
* Possibilité d’impression au format papier.

Une image contenant texte, diagramme, capture d’écran, conception

Description générée automatiquement

Figure 6 – Use case OP-colis

Sur l’écran d’accueil de l’utilisateur OP-colis, en cliquant sur le bouton « CLIENTS », la personne aura la possibilité d’aller vers un nouvel écran « CLIENTS » regroupant la liste des clients, les fiches clients et la possibilité d’envoi de mails personnalisés (Figure 7). Par le bouton « COMMANDES », l’utilisateur aura accès aux commandes en cours, archivés et aux stocks de goodies. Le bouton « LISTES » renverra vers des listes correspondantes aux emballages, aux relations poids/colis et poids-vignette/colis. Ces listes pourront être imprimées tout comme les fiches client et les commandes. Un bouton « STATISTIQUES » pourra être mise en place pour étudier les ventes des goodies.

Une image contenant texte, capture d’écran, Rectangle, Police

Description générée automatiquement

Figure 7 – Maquette écran d’accueil OP-colis

##### **Gestion des stocks** (Figure 8)

* Gestion des stocks.
* Possibilité de mise à jour des stocks.
* Possibilité d’impression au format papier.

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquementUne image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement

Figure – Use case OP-stock

En se connectant, l’utilisateur OP-stock aura la possibilité de visualiser le stock de goodies et de le mettre à jour après chaque inventaire. Il pourra imprimer la liste et les fiches des goodies (Figure 9).

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Figure 9 – Maquette écran d’accueil OP-stock

## Architecture technique

#### Choix de l’Environnement Informatique

La refonte de l’application de gestion de la fromagerie DIGICHEESE s’inscrit dans une envie de modernisation des outils actuels et d’offrir des meilleures performances, une maintenance active et un potentiel d’évolutions.

La nouvelle application sera développée selon une architecture en trois axes à savoir le frontend, le backend et les bases de données afin de séparer clairement les responsabilités et de faciliter les opérations de maintenance et l’évolution du système.

La partie **frontend** est le développement de l’interface utilisateur destinée aux opérateurs (colis, stocks) et aux administrateurs.

Le **backend** gère les fonctionnalités de l’application et permet à la liaison aux **bases de données** qui permettent le stockage des informations correspondant aux clients, commandes et aux stocks.

##### **Frontend**

 Tableau 1 - Comparatif des technologies de développement du frontend

|  |  |  |  |
| --- | --- | --- | --- |
| **Framework** | **Langage** | **Points Positifs** | **Points négatifs** |
| React | JavaScript | - Composants réutilisables   - DOM virtuel pour des performances optimisées   - Large communauté et écosystème | - Nécessite une configuration initiale complexe |
| Vue.js | JavaScript | - Facile à apprendre   - Documentation claire   - Intégration simple avec des projets existants | - Moins populaire que React ou Angular   - Écosystème plus petit |

Pour le développement de l’interface utilisateur, nous recommandons l’utilisation de **React** ou **Vue.js**, deux bibliothèques JavaScript modernes reconnues pour leur performance et leur flexibilité.

Ces deux framework facilitent la création de composants modulaires et réutilisables, réduisant ainsi le temps de développement. Leurs performances sont optimisées grâce au DOM virtuel qui permet des rendus rapides et fluides.

##### **Backend**

Tableau 2 - Comparatif des technologies de développement du frontend

|  |  |  |  |
| --- | --- | --- | --- |
| **Framework** | **Langage** | **Points Positifs** | **Points négatifs** |
| Django | Python | - Framework complet (ORM, authentification)   - Sécurité intégrée   - Rapide pour le développement | - Monolithique   - Moins adapté aux applications asynchrones |
| FastAPI | Python | - Très rapide et performant   - Documentation automatique   - Support natif de l’asynchrone | - Moins de fonctionnalités intégrées que Django   - Moins adapté aux gros projets monolithiques |

La partie backend sera développée soit en **Django** ou **FastAPI**, deux frameworks Python robustes et performants.

Django fournit un ensemble intégré de fonctionnalités (authentification, ORM, administration), protège contre les failles courantes comme le CSRF et l’injectif SQL et est adapté aux applications de taille moyenne à grande.

FastAPI est un framework asynchrone rapide, génère automatique une documentation interactive avec Swagger et utilise les annotations Python

##### **Base de données**

La base de données retenue est **MySQL**, une solution éprouvée et performante pour les applications de gestion de données transactionnelles, utilisée en entreprise et compatible avec Django notamment.

#### Serveur et hébergement

L’application sera déployée sur un **serveur interne** au réseau intranet de DIGICHEESE pour des raisons de sécurité et de confidentialité.

* **Système d’Exploitation** : **Linux (Ubuntu Server)** pour sa stabilité, sa sécurité et ses performances.
* **Serveur Web** : **Nginx** pour servir le frontend et le backend de manière efficace.
* **Conteneurisation** : **Docker** pour isoler les composants de l’application et faciliter le déploiement.
* **Orchestration** : Possibilité d’utiliser **Docker Compose** pour simplifier le lancement des différents services.

#### Sécurité

Des mesures de sécurité seront appliquées pour garantir l’intégrité et la confidentialité des données :

* **Chiffrement HTTPS** : Pour sécuriser les échanges entre le frontend et le backend.
* **Authentification JWT** : Pour contrôler l’accès à l’application.
* **Sauvegardes Régulières** : Planification de sauvegardes automatiques des données critiques.
* **Protection contre les Attaques** : Implémentation de protections contre les attaques courantes (CSRF, XSS, injection SQL).

Plus de détails pourront être fourni après consultation de l’équipe technique.

## Modélisation des bases de données

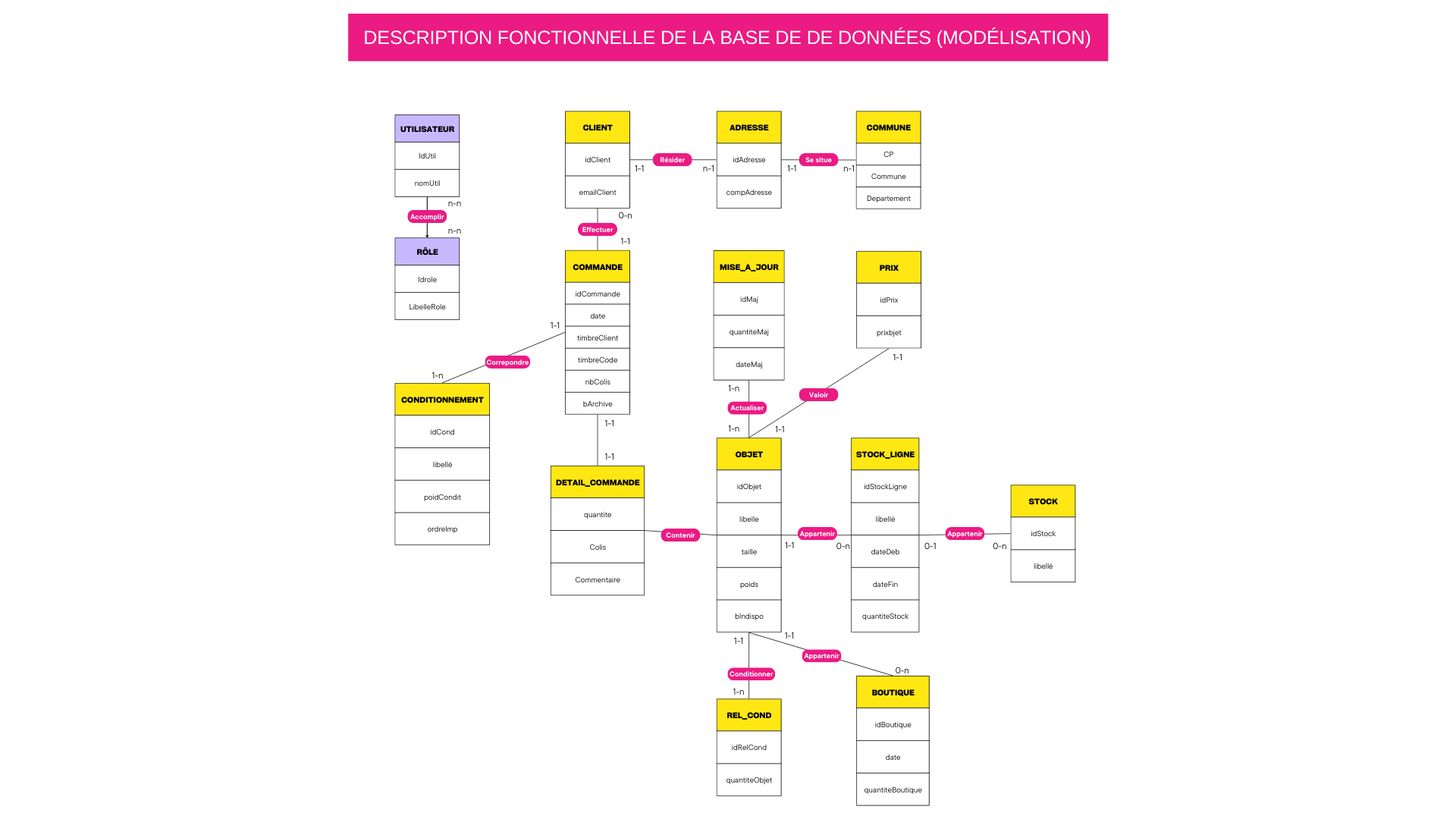
La modélisation des bases de données pour la refonte de l'application DIGICHEESE repose sur une structure claire et cohérente, permettant de gérer efficacement les informations liées aux clients, aux commandes, aux stocks, aux utilisateurs et aux différents paramètres. Cette modélisation assure une organisation logique des données, facilite les opérations de requêtes, et garantit l'intégrité et la cohérence des informations.

**Objectifs :**

* **Organiser les données de manière structurée** pour répondre aux besoins de l'application
* **Faciliter les requêtes et les opérations de gestion** des clients, des commandes, des stocks et des utilisateurs
* **Garantir l’intégrité des données** à travers des relations et des contraintes appropriées
* **Assurer la scalabilité** pour s’adapter aux futures évolutions du système
* **Optimiser les performances** pour garantir une application rapide et réactive

*Voir le diagramme page suivante (plus de détails en annexe)*

Cette modélisation des bases de données offre une **structure robuste et évolutive** pour l’application DigiCheese. Grâce à cette organisation, les opérations de gestion des clients, commandes, stocks et utilisateurs seront efficaces et sécurisées. La clarté des relations facilite également le développement du backend et garantit la cohérence des données à long terme.



## Budget

Le budget prévisionnel pour la refonte de l’application de DIGICHEESE sera de **50000 € HT**.

|  |  |  |
| --- | --- | --- |
| **Catégorie** | **Description** | **Budget (€)** |
| **1. Développement** |  |  |
| Fontend | Création d'une interface utilisateur adaptée | *10000* |
| Backend | Développement des fonctionnalités internes | *12000* |
| Base de données | Configuration et optimisation | *3000* |
| Tests | Mise en place des tests automatisés | *5000* |
| Authentification et sécurité | Intégration d'un système de connexion sécurisé | *5000* |
| Sous-total |  | **35000** |
| **2. Hébergement et infrastructure** |  |  |
| Hébergement | Serveurs internes | *2000* |
| Configuration initiale | Déploiement sur serveurs | *2000* |
| Maintenance |  | *2000* |
| Sous-total |  | **6000** |
| **3. Formation et documentation** |  |  |
| Documentation |  | *2000* |
| Formation |  | *2000* |
| Sous-total |  | **4000** |
| **4. Maintenance initiale** |  |  |
| Support technique | Correction des bugs | *3000* |
| Améliorations |  | *2000* |
| Sous-total |  | **5000** |
| Total |  | **50000** |

## Suivi du projet

Le suivi de l’avancée du projet se fera sur le logiciel Jira en suivant la méthode Scrum.

Chaque tâche sera répartie entre les différents sprints du projet. Un sprint aura une durée comprise entre 2 et 4 semaines.

Des daily scrum sont prévues pour suivre l’avancée du projet entre le product owner et chaque acteur appliqué.

Tout en appliquant la méthode Agile, des réunions régulièrement seront organisées avec le client (toutes les 2 semaines) afin d’exposer les avancées et les problèmes rencontrés.

Une première version de l’application sera livrée au bout des trois premiers mois pour un test en condition réelle. Cette phase permettra un retour des utilisateurs sur les bugs rencontrés et les améliorations à apporter.

A la fin du cinquième mois, l’application finale sera livrée au client DIGICHEESE.

## Calendrier

La durée prévisionnelle pour la réalisation de cette refonte est de **5 mois**.

|  |  |
| --- | --- |
| **Sprint 1** | **Développement du module Administration** |
| 23/12/2024 au 12/01/2025 |
| *Création du module de gestion des utilisateurs, des bases de données et des paramètres* |
| **Sprint 2** | **Développement du module Gestion des Colis** |
| 13/01/2025 au 2/02/2025 |
| *Création du module de gestion des commandes et des colis* |
| **Sprint 3** | **Développement du module Gestion des Stocks** |
| 3/02/2025 au 16/02/2025 |
| *Création du module de gestion des stocks* |
| **Sprint 4** | **Intégration Front-End** |
| 10/02/2025 au 2/03/2025 |
| *Intégrer les composants front-end avec le back-end* |
| **Sprint 5** | **Tests et Déploiement** |
| 3/03/2025 au 23/03/2025 |
| *Finaliser les tests et préparer le déploiement* |
| **Sprint 6** | **Maintenance et améliorations** |
| 25/03/2025 au 23/05/2025 |
| *Assurer le support post-développement et apporter des améliorations* |
| **Date clé** | **23 décembre 2024 :** Début du développement  **3 mars 2025 :** Fin de l’intégration front-end et back-end  **24 mars 2025** : Livraison de la première version de l’application  **23 mai 2025 :** Fin du support initial et clôture du projet |

## 

## Annexes

##### **Structure des tables des bases de données**

Les entités identifiées dans le diagramme de classes sont représentées sous forme de **tables relationnelles** dans la base de données MySQL. Voici une description détaillée de chaque table, de ses attributs et de ses relations avec d’autres tables.

|  |  |
| --- | --- |
| Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement | Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement |
| Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement | Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement |

|  |  |
| --- | --- |
| Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement | Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement |

|  |  |
| --- | --- |
| Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement | Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement |

|  |  |
| --- | --- |
| Une image contenant texte, capture d’écran, Police, nombre  Description générée automatiquement |  |