

RAPPORT DE STAGE DE PROJET DE FIN D'ETUDES INGENIEURS EN INFORMATIQUE

Option : Génie Logiciel/DevOps

Intitulé du stage

Conception et développement d'une application web de gestion de collecte et de remise en bon état d'usage du matériel médical

Réalisé par
Amal Grami

Entreprise d'accueil



THE CODING MACHINE

Encadrant Entreprise
Louis CAILLE

Encadrant SESAME
Amal BOUAZIZ

Année Universitaire
2022-2023

Dédicace

“ Louange à **DIEU** tout puissant qui m'a permis de voir ce jour tant attendu.

Du profond de mon cœur, je dedie ce travail à tous ceux qui me sont chers.

A MA FAMILLE

À mon père, tu as toujours été mon modèle de persévérance et de détermination. Grâce à toi, j'ai pu me surpasser et relever les défis avec confiance. Je te suis infiniment reconnaissante pour ta présence rassurante et ton amour inconditionnel.

À ma mère, ton amour incommensurable et ta bienveillance ont été mes sources d'inspiration quotidiennes. Tu m'as toujours encouragée à poursuivre mes rêves et à donner le meilleur de moi-même. Je te suis profondément reconnaissante pour ton soutien indéfectible et ta foi en moi.

À mon frère, même si nous sommes éloignés géographiquement, ton amour et ton soutien ont toujours été proches de moi. Les appels, les messages d'encouragement et les paroles réconfortantes que tu m'as adressés ont été des bouées d'espoir dans les moments de doute et de fatigue.

A MON ENCADRANTE

À mon encadrante universitaire, je tiens à exprimer ma reconnaissance pour votre expertise, votre patience et votre investissement dans mon développement professionnel. Vous avez été une guide exceptionnelle, toujours prête à m'écouter, à répondre à mes questions et à me guider sur la voie de la réussite. Vos conseils éclairés et vos encouragements m'ont permis de m'épanouir dans ce stage et d'acquérir des compétences précieuses pour mon avenir. Je vous suis sincèrement reconnaissante pour votre confiance, votre disponibilité et votre dévouement.

A MES AMIS

À mes amis, vous êtes les piliers de mon équilibre. Vos encouragements, vos conseils et votre soutien ont été les forces motrices qui m'ont poussée à donner le meilleur de moi-même. Votre amitié sincère et vos encouragements constants m'ont donné la confiance nécessaire pour explorer de nouvelles voies et me surpasser. Je vous suis infiniment reconnaissante d'être toujours à l'écoute, prêts à soutenir mes rêves et à partager mes réussites.

A toute personne qui a contribué de près ou de loin à la réalisation de ce travail. Cette dédicace est une humble expression de ma gratitude envers vous. Votre soutien, votre amour et votre présence à distance ont été des trésors inestimables pour moi. Je vous remercie du fond du cœur pour tout ce que vous avez fait et continuez de faire pour moi.

”

Amal GRAMI

Remerciements

Je tiens, avant de présenter mon travail, à exprimer ma grande reconnaissance envers les personnes qui m'ont de près ou de loin apporté leurs soutiens.

Je tiens à exprimer ma profonde reconnaissance pour la directrice du projet **Soumaya GOUIDER**, votre engagement et votre soutien tout au long de ce projet. Votre capacité à diriger et à inspirer l'équipe a été exemplaire. Votre leadership visionnaire nous a permis de relever les défis avec confiance et de réaliser des résultats remarquables.

J'exprime mes sincères reconnaissances à mon encadrant professionnel Monsieur **Louis CAILLE**, votre soutien constant et vos encouragements ont été un moteur pour ma motivation et ma confiance en moi. Vous avez cru en mon potentiel et m'avez poussé à me surpasser. Vos retours constructifs et vos conseils éclairés ont été précieux pour mon développement professionnel. Grâce à vous, j'ai pu acquérir de nouvelles compétences, relever des défis et atteindre mes objectifs.

Ma profonde gratitude s'adresse également à mon encadrante académique Madame **AMAL BOUAZIZ**, Votre rôle d'encadrante académique a été bien au-delà de simples conseils et directives. Vous avez été une source d'inspiration et de motivation pour moi, m'encourageant à repousser mes limites et à viser l'excellence. Merci du fond du cœur pour votre engagement, je suis honorée d'avoir été guidée par une personne aussi exceptionnelle que vous.

À chaque collègue de mon équipe, je souhaite exprimer ma sincère gratitude à chacun d'entre vous pour votre collaboration exceptionnelle tout au long de notre parcours ensemble. Votre engagement, votre professionnalisme et votre esprit d'équipe ont été des éléments clés de notre succès collectif.

Que les membres de jury trouvent, ici, l'expression de mes remerciements pour l'honneur qu'ils me font en prenant le temps de lire et d'évaluer ce travail.

Table des matières

INTRODUCTION GENERALE	1
1 Cadre général du projet	2
Introduction	3
1.1 Présentation de l'organisme d'accueil	3
1.1.1 Présentation de The coding Machine [1]	3
1.1.2 Structure de l'organisme	4
1.2 Cadre du Projet	4
1.2.1 Présentation du projet	5
1.2.2 Etude de l'existant	5
1.3 Méthodologie de travail	6
1.4 Langage de modélisation utilisé	7
Conclusion	7
2 Planification du Projet	8
Introduction	9
2.1 Spécifications des Besoins	9
2.1.1 Spécifications des Besoins Fonctionnels	9
2.1.2 Spécifications des Besoins non fonctionnels	10
2.1.3 Diagramme de cas d'utilisation global	10
2.2 Pilotage du Projet avec Scrum	11
2.2.1 Equipe et rôle	11
2.2.2 Backlog produit	12
2.2.3 Planification des sprints	13

Table des matières

2.3	Architecture de l'application	13
2.3.1	Architecture client-serveur	13
2.3.2	Architecture Logicielle	14
2.4	Environnement de travail	15
2.4.1	Environnement matériel	16
2.4.2	Environnement Logiciel	16
2.4.3	Langages de programmation	17
	Conclusion	18
3	Sprint 1 : Authentification et gestion des articles, collectes et des tournées	19
	Introduction	20
3.1	Backlog du sprint 1	20
3.2	Analyse et conception	20
3.2.1	Diagramme de cas d'utilisation	20
3.2.2	Raffinement de cas d'utilisation Gérer collecte	21
3.2.3	Raffinement de cas d'utilisation Gérer article	22
3.2.4	Raffinement de cas d'utilisation Gérer tournée	22
3.2.5	Diagrammes de séquence	23
3.3	Conception	28
3.4	Réalisation	30
3.4.1	Interface d'authentification	30
3.4.2	Gestion des articles	31
3.4.3	Gestion des collectes	33
3.4.4	Gestion Tournées	35
	Conclusion	38
4	Sprint 2 : Gestion des rôles ,des agences et suivi des statistiques	39
	Introduction	40
4.1	Backlog du sprint 2	40
4.2	Analyse et conception	40
4.2.1	Diagramme de cas d'utilisation	40

4.2.2	Raffinement de cas d'utilisation Gérer rôle	41
4.2.3	Raffinement de cas d'utilisation Gérer agence	42
4.2.4	Diagrammes de séquence	42
4.3	Conception	48
4.4	Réalisation	49
4.4.1	Liste des rôles	49
4.4.2	Modifier un rôle	50
4.4.3	Consulter agence :	51
4.4.4	Tableau de bord :	51
4.4.5	Tests unitaires :	55
	Conclusion	56
5	Sprint 3 : Conteneurisation	57
	Introduction	58
5.1	Backlog du sprint 3	58
5.2	Analyse et conception	58
5.2.1	Diagramme d'activité	59
5.2.2	Architecture physique	59
5.3	Réalisation	60
5.3.1	Configuration service	60
5.3.2	Make file	64
5.3.3	Visualisation	66
	Conclusion	68
	Conclusion générale	69
	Webographie	70

Table des figures

1.1	Logo The Coding Machine.	3
1.2	Organigramme The Coding Machine.	4
2.1	Diagramme de cas d'utilisation global	11
2.2	Architecture client-serveur	14
2.3	Architecture logicielle MVC	15
2.4	Phpstorm.	16
2.5	GitLab.	17
2.6	Symfony.	18
2.7	Vue.js.	18
3.1	Diagramme de cas d'utilisation du sprint 1	21
3.2	Cas d'utilisation Gérer collecte	21
3.3	Diagramme de Cas d'utilisation Gérer article.	22
3.4	Diagramme de Cas d'utilisation Gérer tournée.	23
3.5	Diagramme de séquence Authentification.	23
3.6	Diagramme de séquence Ajouter un article.	25
3.7	Diagramme de séquence Consulter Article.	27
3.8	Diagramme de classe de sprint 1	28
3.9	Interface d'authentification	30
3.10	Interface Ajouter un article	31
3.11	Interface consulter liste des articles	32
3.12	Interface modifier article	33
3.13	Interface création demande de collecte	34

3.14 Interface consulter collecte	34
3.15 Interface lister les collectes	35
3.16 Interface planifier tournée	36
3.17 Interface modifier Tournée	36
3.18 Interface consulter carte des collectes	37
4.1 Diagramme de cas d'utilisation du sprint 2	41
4.2 Cas d'utilisation gérer rôle	41
4.3 Diagramme de Cas d'utilisation gérer agence.	42
4.4 Diagramme de séquence supprimer un rôle.	43
4.5 Diagramme de séquence modifier rôle.	45
4.6 Diagramme de séquence Consulter agence.	47
4.7 Diagramme de classe de l'application	48
4.8 Interface liste des rôles	49
4.9 Interface modifier rôle	50
4.10 Interface consulter agence	51
4.11 Tableau de board	52
4.12 statistiques des collectes par agence	52
4.13 statistiques des tournées par agent	53
4.14 Accès rapide	54
4.15 Prochaine tournée	55
4.16 Tests unitaires	56
5.1 Diagramme d'activité du sprint	59
5.2 Diagramme de Déploiement	60
5.3 Fichier Docker compose	61
5.4 docker file backend	61
5.5 docker file frontend	62
5.6 Make file	64
5.7 La plateforme conteneurisée	66
5.8 Base de données conteneurisée	67
5.9 La plateforme conteneurisée	67

INTRODUCTION GENERALE

Dans une ère marquée par la technologie et l'innovation, la digitalisation est aujourd'hui considérée comme une réelle révolution. Le monde est actuellement en pleine migration vers le numérique, éradiquant ainsi les systèmes traditionnels habituels.

Afin de rester compétitives et de suivre l'évolution des marchés mondiaux, les entreprises contemporaines doivent se conformer à cette révolution en optant pour la digitalisation de leurs activités. La transition numérique touche aujourd'hui la majorité des secteurs, de l'industrie à la médecine, et les entreprises ont grandement besoin de l'assistance de professionnels dans ces domaines. C'est dans ce contexte que s'inscrit notre stage de fin d'études, qui vise à contribuer à la transformation digitale de notre client en développant une plateforme de gestion de collecte, de reconditionnement et de remise en bon état d'usage du matériel médicaux.

Grâce à cette plateforme, les établissements de santé peuvent bénéficier d'un processus efficace et sécurisé pour récupérer les équipements médicaux usagés, les remettre en bon état d'utilisation et les réintégrer dans leur parc de matériel. Cela permet de maximiser l'utilisation des ressources, de réduire les coûts liés à l'achat de nouveaux équipements et de contribuer à une meilleure gestion durable des ressources médicales.

Ce rapport se compose de cinq chapitres. Dans le premier chapitre, nous commençons par une présentation du cadre du projet, dans lequel nous exposons la société et l'étude de l'existant. Le deuxième chapitre est consacré à l'analyse et à la spécification des besoins. Nous identifions les besoins, établissons le backlog du produit, décrivons l'environnement de développement et les choix techniques, détaillons l'architecture du système et planifions les sprints. Les troisième et quatrième chapitres abordent le contenu de premier et deuxième sprints, qui concernent la conception et le développement des besoins de notre client. Dans le cinquième chapitre, nous traitons le troisième sprint qui sera consacré à la partie de virtualisation. Enfin, nous concluons avec une conclusion.

Chapitre 1

Cadre général du projet

“*Prévoir consiste à projeter dans l'avenir ce qu'on a perçu dans le passé.*”

Henri Bergson

Introduction

Dans ce chapitre, il s'agit de placer le projet dans son cadre général. Nous commencerons en premier lieu par la présentation de l'organisation d'accueil. Puis, en second lieu, nous présenterons le projet, allant de son contexte, de la problématique à laquelle il répond, jusqu'à la solution proposée.

1.1 Présentation de l'organisme d'accueil

1.1.1 Présentation de The coding Machine [1]

The Coding Machine (TCM) est une société française par Actions Simplifiée (S.A.S) fondée en 2005 par Jean Guillaume Dajurdin et David Négrier. Elle est spécialisée dans le secteur d'activité de la programmation informatique, telle que le développement sur mesure de plateformes web (extranet, sites internet, web services, applications mobiles). TCM compte principalement trois grands sièges, chacun employant en moyenne 50 salariés : PARIS, où j'ai effectué mon stage, LYON et HONG-KONG.



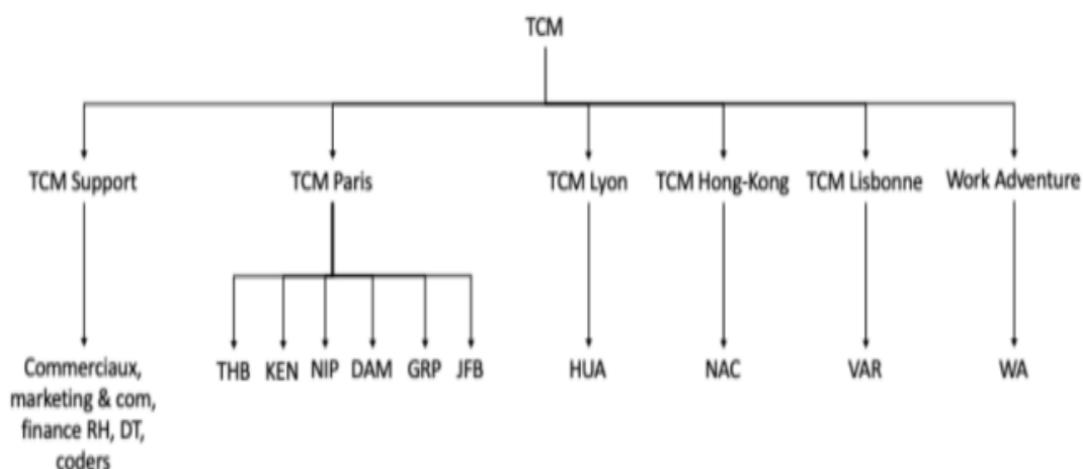
FIGURE 1.1 – Logo The Coding Machine.

1.1.2 Structure de l'organisme

TCM compte plus de 70 salariés répartis entre les différentes filiales. Structuralement, l'entreprise compte plusieurs entités :

- Le Holding permettant de regrouper toutes les filiales sous une même entreprise.
- Le Support (ou Hub), composé des services commerciaux, RH, marketing et communication, financier et d'une direction technique. Les « coders » sont compris au sein de ce département.
- Les filiales (Paris, Lyon, Hong-Kong et Lisbonne).

L'organigramme de la société est structuré comme suit :



* Chaque équipe est nommée à partir du trigramme de son DP

FIGURE 1.2 – Organigramme The Coding Machine.

1.2 Cadre du Projet

Dans le cadre de notre projet de fin d'études chez The Coding Machine, nous sommes chargés de concevoir et développer une plateforme spécialisée dans le domaine de la santé appelée "LEA"

1.2.1 Présentation du projet

LEA vise à faciliter la collecte, le reconditionnement et la remise en bon état des matériels médicaux. En offrant aux établissements de santé un processus fluide et sécurisé, notre plateforme permettra de récupérer les équipements médicaux usagés, de les restaurer à un état fonctionnel optimal, et de les réintégrer dans leur inventaire.

1.2.2 Etude de l'existant

Afin de mener à bien notre mission, il est nécessaire d'étudier la solution existante pour bien comprendre l'état actuel.

Description de l'existant

Actuellement, l'application existante ne permet que d'exposer les articles après leur reconditionnement. Cependant, avant d'arriver à cette étape d'exposition, les articles passent par plusieurs autres processus. Tout d'abord, les demandes sont reçues par téléphone, puis elles sont enregistrées sur papier. Ensuite, le client contacte les agents de collecte pour vérifier leur disponibilité. Une fois qu'un agent de collecte est disponible, il se rend sur site pour récupérer l'article et le transporte vers le dépôt le plus proche de son agence. Au dépôt, un agent doit saisir manuellement sur un support papier que l'article est prêt pour le reconditionnement. Par la suite, l'article est déplacé vers l'usine où il sera reconditionné. Lorsque l'article est prêt à être exposé, les informations sont entrées dans l'application.

Problématique

L'application actuelle présente diverses lacunes qui affectent son efficacité et sa fonctionnalité. Tout d'abord, il n'y a pas de suivi en temps réel de l'état et de l'emplacement des articles, ce qui entraîne des retards et des difficultés dans la gestion de la chaîne logistique. De plus, la saisie manuelle des informations sur support papier expose l'application à des risques d'erreurs et de perte de données. En outre, les limitations de communication empêchent une coordination fluide entre les agents de collecte, les dépôts et l'usine de reconditionnement. L'absence de fonctionnalités avancées de gestion des stocks crée des difficultés dans le suivi des niveaux d'inventaire, le réapprovisionnement et la planification des ressources. Enfin, le manque de statistiques disponibles limite la capacité de l'application à effectuer des analyses approfondies sur les performances de l'entreprise. Par conséquent, une amélioration de l'application est nécessaire pour résoudre ces problèmes et optimiser les opérations de l'entreprise.

Solution proposée

Afin de remédier à ces problèmes, nous proposons la mise en place d'une plateforme numérique complète et intégrée. Cette plateforme vise à faciliter et optimiser le processus de collecte, de reconditionnement et de gestion des matériels médicaux. En adoptant cette solution, notre client pourra bénéficier d'un processus plus fluide, efficace et traçable. La numérisation de ce processus permettra une meilleure gestion des données, une coordination simplifiée et une optimisation des ressources. En définitive, cette solution contribuera à améliorer la qualité des services, à réduire les coûts et à assurer une gestion plus responsable des ressources médicales.

1.3 Méthodologie de travail

La sélection de la méthodologie de travail est une étape cruciale dans le développement du projet. Dans notre cas, nous avons opté pour la méthode agile, plus précisément Scrum, tout au long de la réalisation de notre projet.

Cette approche itérative et incrémentale nous permet d'assurer un niveau de qualité optimal, d'améliorer la performance de notre solution et de répondre aux exigences du projet. De plus, elle nous permet de mieux gérer les délais et d'éviter tout dépassement.

La méthodologie Scrum favorise la collaboration et la communication entre les membres de l'équipe. Elle encourage également la transparence et l'adaptabilité face aux changements. Grâce à des cycles courts appelés "sprints", nous sommes en mesure de livrer des résultats fonctionnels à intervalles réguliers, ce qui facilite les ajustements et les retours d'expérience.

En résumé, l'utilisation de la méthode agile, en particulier Scrum, nous permet de travailler de manière efficace et collaborative, d'assurer la qualité de notre solution et de répondre aux attentes du projet tout en respectant les délais fixés.

Les rôles SCRUM

L'organisation de l'équipe Scrum est un des piliers qui font la réussite de la méthode. L'équipe se compose comme suit :

- Le Scrum Master est responsable de faciliter le processus Scrum et de s'assurer que l'équipe suit les principes et les pratiques de Scrum. Il aide l'équipe à s'auto-organiser et à résoudre les problèmes.
- Le Product Owner est chargé de définir les exigences du projet et de les prioriser dans le backlog du produit. Il travaille en étroite collaboration avec l'équipe de

développement pour s'assurer que les besoins du client sont bien compris et pris en compte.

- L'équipe de développement est responsable de réaliser les tâches et de développer les fonctionnalités du projet. Elle est autonome et s'auto-organise pour atteindre les objectifs du sprint.

1.4 Langage de modélisation utilisé

Nous avons utilisé UML comme langage de modélisation pour créer nos diagrammes. UML nous permet de représenter visuellement la structure, le comportement et les relations des éléments de notre système. Les diagrammes UML, tels que les diagrammes de cas d'utilisation et de classe, nous ont aidés à communiquer et à documenter nos conceptions logicielles de manière claire et standardisée.

Conclusion

Ce chapitre a permis de situer le projet dans son cadre général en présentant l'organisme d'accueil, le contexte, la problématique, l'étude et le critique de l'existant ainsi que la solution proposée en précisant la méthodologie adoptée. Dans le chapitre suivant, nous nous intéresserons à l'identification et à la planification des besoins du projet et nous présenterons notre environnement de développement ainsi que nos choix techniques.

Chapitre 2

Planification du Projet

“ Toutes les choses sont créées deux fois : une première fois mentalement, puis une seconde fois physiquement. La clé de la créativité, c'est de commencer avec la fin à l'esprit, avec une vision et un modèle du résultat désiré ”

Covey

Introduction

La planification du projet est l'une des phases les plus cruciales. D'où l'objectif de ce chapitre et la compréhension du contexte du système. Cela comprend la détermination des besoins, du backlog de produit, de l'environnement de développement et les choix techniques ainsi que la planification des releases.

2.1 Spécifications des Besoins

Dans cette section, notre objectif est d'identifier les personnes impliquées dans notre application et d'analyser les besoins fonctionnels et non-fonctionnels .

2.1.1 Spécifications des Besoins Fonctionnels

Dans notre système informatique, nous avons deux principaux acteurs un utilisateur et un administrateur

Un utilisateur peut :

- S'authentifier.
- Gérer les articles.
- Gérer les collectes.
- Gérer les tournées.

L'administrateur possède toutes les fonctionnalités d'un utilisateur standard, avec en plus la capacité de :

- Paramétrier et gérer les rôles.
- Gérer les agences.
- consulter les statistiques.

Mis à part les besoins fonctionnels de notre client, notre société exige également la mise en place d'un environnement favorable pour les développeurs.

Parmi le besoins spécifiques à cet environnement :

- Simplifier l'installation du projet pour les développeurs en réduisant le nombre d'étapes nécessaires et en automatisant autant que possible le processus d'installation.
- Remédier les problèmes de configuration d'environnement.
- Isoler les dépendances afin de permettre aux développeurs de travailler et d'effectuer des évolutions sur chaque partie séparément, sans interférer avec les autres composants du projet.

2.1.2 Spécifications des Besoins non fonctionnels

Les besoins non fonctionnels représentent les exigences implicites auxquelles le système doit répondre.

Le tableau 2.1 représente les trois aspects importants :

TABLE 2.1 – Besoins non fonctionnels

Besoin	Description
La Portabilité	il s'agit de la capacité d'une application à fonctionner sur différentes plateformes de manière transparente et sans nécessiter de modifications majeures.
L'ergonomie	Cela fait référence à la conception des interfaces utilisateur conviviale et efficace
La sécurité	Il est essentiel que l'application garantisse la confidentialité, l'intégrité et la disponibilité des informations, en mettant en place des mesures de protection contre les menaces et les accès non autorisés.

2.1.3 Diagramme de cas d'utilisation global

Les besoins et les acteurs qu'on a extrait se traduisent dans ce diagramme qui sera notre repère tout au long du développement. Ce diagramme nous donne une vision globale sur l'application web. Au cours de chaque sprint, on construira une conception plus détaillée d'une façon incrémentale.

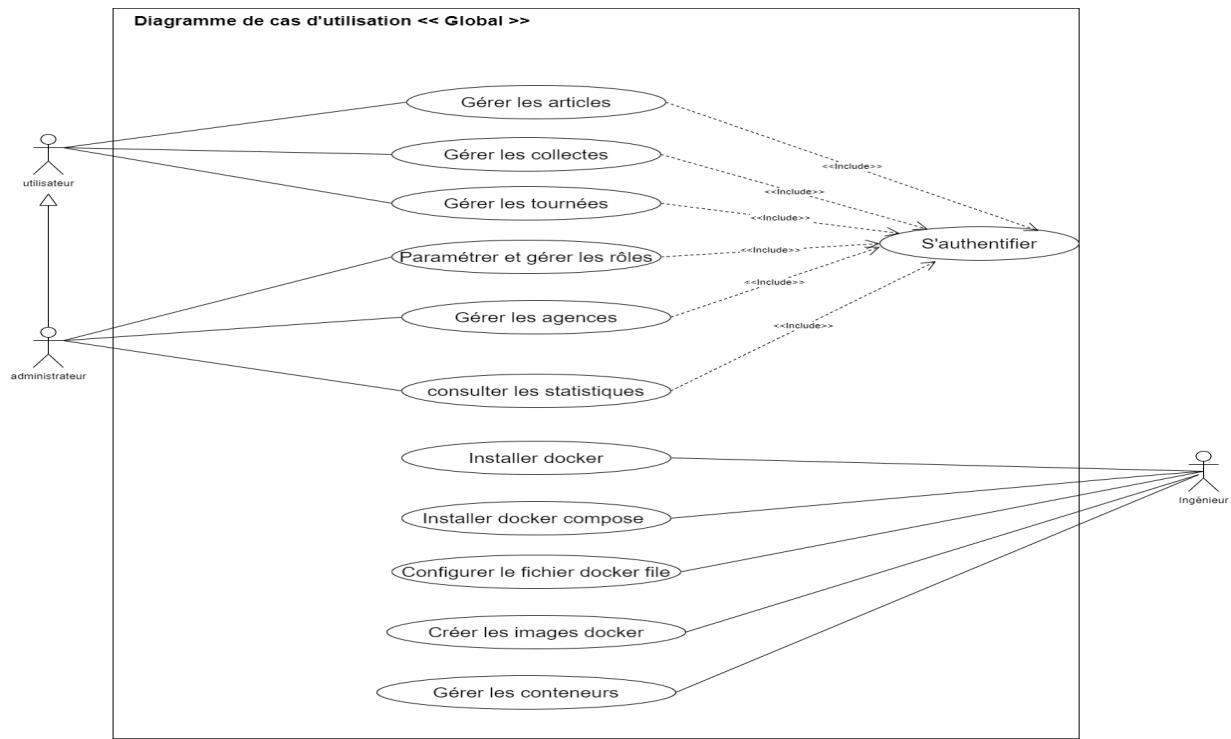


FIGURE 2.1 – Diagramme de cas d'utilisation global

2.2 Pilotage du Projet avec Scrum

2.2.1 Équipe et rôle

Comme on a mentionné dans la section méthodologie de travail du premier chapitre , nous avons choisi la méthodologie agile car elle offre la facilité d'un développement rapide et permettra de réutiliser les fonctionnalités développées séparément de la plateforme. Notre méthodologie nécessite la collaboration de plusieurs intervenants.Pour notre projet :

- Product Owner (PO) : M. Louis caillé est notre encadrant professionnel, il a une forte connaissance sur le projet et son cahier de charge.
- Scrum Master (SM) : Mme Yama NIDONE, qui est responsable de la gestion et de la coordination du projet.
- Équipe de développement (Development Team) : développement des objets techniques permettant de répondre au backlog convenu (Technique) est constitué de Milan Lima et Amal Grami.

2.2.2 Backlog produit

En se basant sur les besoins pré-cités, voici la description du backlog produit, comme illustré par la table 2.2, qui représente les tâches (User Stories). Chaque User Story est caractérisé par un ID et une priorité qui a été fixés en commun accord avec le product owner.

TABLE 2.2 – Backlog du produit

ID	User Stories	Priorité
1	En tant qu'utilisateur , je veux m'authentifier	1
2	En tant qu'utilisateur , je veux créer,consulter, modifier, supprimer les collectes et consulter la carte des collectes	1
3	En tant qu'utilisateur , je veux créer, exporter, modifier, supprimer et consulter les articles	1
4	En tant qu'utilisateur , je veux ajouter, modifier, consulter et annuler les tournées de collecte, Affecter ou supprimer une collecte à une tournée	1
5	En tant qu'administrateur ,je veux gérer les rôles des utilisateurs dans la plateforme	2
6	En tant qu'administrateur ,je veux créer, enregistrer, modifier, supprimer et consulter les agences	2
7	En tant qu'administrateur , je veux consulter les statistiques	2
8	En tant qu'ingénieur devOps, je veux installer docker	3
9	En tant qu'ingénieur devOps, je veux installer docker compose	3
10	En tant qu'ingénieur devOps, je veux configurer le fichier docker file	3
11	En tant qu'ingénieur devOps, je veux créer les images docker	3
12	En tant qu'ingénieur devOps, je veux configurer le fichier docker-compose.	3
13	En tant qu'ingénieur devOps, je veux lancer les services dans des conteneurs	3
14	En tant qu'ingénieur devOps, je veux gérer les conteneurs	3

2.2.3 Planification des sprints

Un plan de haut niveau est convenu pour les sprints durant la réunion de planification . Cette planification permet de tracer une ligne directrice qui reflète les attentes en termes de fonctionnalités attendues, la charge, les délais et les responsabilités (Qui fait quoi, quand et comment). Ceci sert de base pour mesurer l'avancement des travaux au sein du projet. Pour créer un plan de diffusion, les éléments suivants doivent être disponibles :

- Un Backlog de produits Scrum hiérarchisé et estimé
- La vitesse de l'équipe Scrum
- Conditions de satisfaction (objectifs pour l'emploi du temps, portée, ressources).

En se basant sur notre backlog du produit, nous avons divisé notre projet comme le montre la table 2.3.

TABLE 2.3 – Planification des sprints

Sprint	fonctionnalités	User Story ID
Sprint 1	Authentification, Gestion des collectes , des articles et des tournées	1,2,3,4
Sprint 2	Gestion des rôles ,agences et suivi des statistiques	5,6,7
Sprint 3	Conteneurisation	8,9,10,11,12,13,14

2.3 Architecture de l'application

2.3.1 Architecture client-serveur

Pour notre application nous avons adopté une architecture à deux niveaux, également connue sous le nom d'architecture client-serveur.c'est un modèle d'architecture logicielle dans lequel une application est divisée en deux parties distinctes : le backend et le frontend.

Le backend, également appelé serveur, est responsable de la gestion des données, de la logique métier et de l'exécution des opérations. Il fournit des services et des fonctionnalités à travers des API ou des services web, répondant ainsi aux demandes des clients.

Le frontend, également appelé client, est la partie de l'application avec laquelle l'utilisateur interagit directement. Il est responsable de la présentation des données, de l'interface utilisateur et de la gestion des interactions utilisateur. Le frontend communique avec le backend en utilisant les API ou les services fournis par celui-ci. Voici quelques avantages cette architecture :

- Séparation claire des responsabilités entre le backend et le frontend.
- Facilité de maintenance et de gestion de l'application grâce à la division des tâches.
- Possibilité de développer et de déployer indépendamment le backend et le frontend, ce qui permet des cycles de développement plus rapides.
- Scalabilité qui permet d'ajuster et de mettre à l'échelle individuellement les ressources du backend et du frontend.
- Réutilisabilité du code grâce à la modularité de l'architecture, ce qui facilite le développement de nouvelles fonctionnalités et la gestion des évolutions.

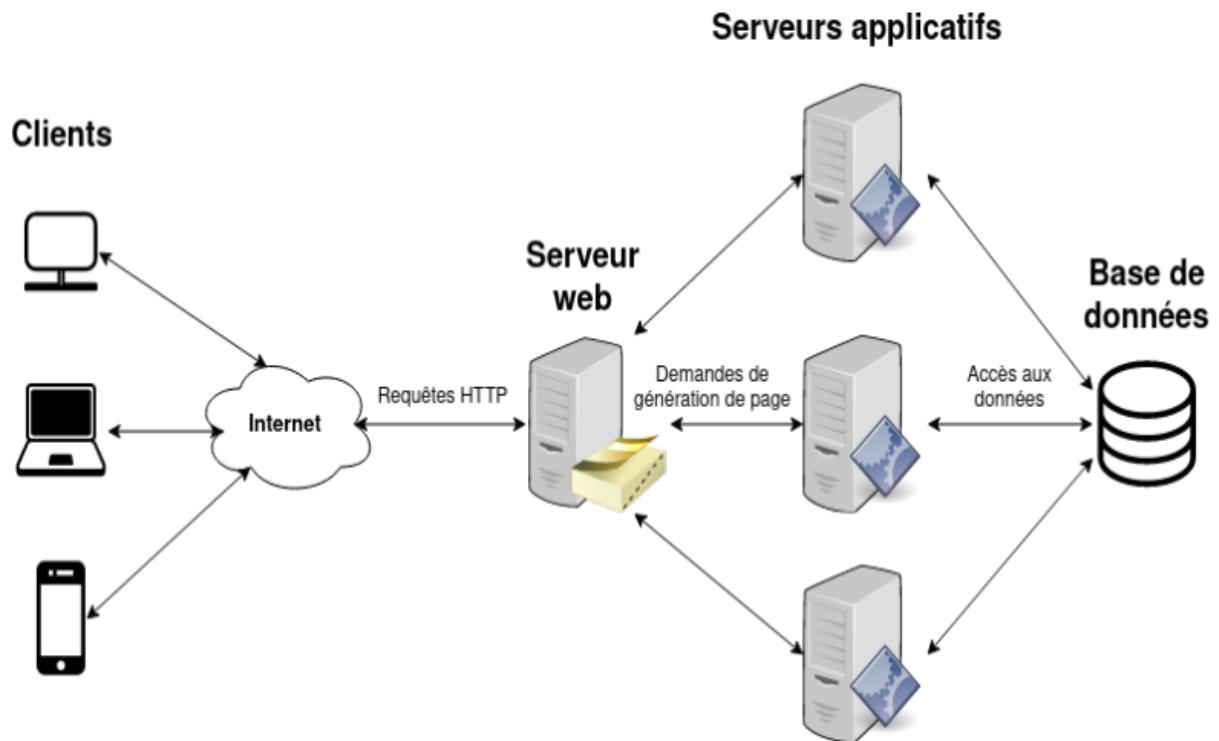


FIGURE 2.2 – Architecture client-serveur

2.3.2 Architecture Logicielle

Architecture MVC[2]

Dans cette partie nous allons adopter le patron de conception architectural MVC pour bien synchroniser avec une architecture physique cohérente. Ce pattern MVC permet de simplifier l'écriture des interfaces graphiques grâce à ces éléments.

MVC (Model-View-Controller ou Modèle-Vue-Contrôleur) est un modèle dans la conception de logiciels. Il met l'accent sur la séparation entre la logique métier et l'affichage du logiciel. Cette «séparation des préoccupations» permet une meilleure répartition du travail et une maintenance améliorée. Certains autres modèles de conception sont basés sur MVC,

tels que MVVM (Model-View-Viewmodel), MTP (Model-View-Presenter) MVW (Model-View-Whatever).

Les 3 parties du modèle de conception de logiciel MVC peuvent être décrites comme suit :

- **Model (modèle)** : gère les données et la logique métier.
- **View (vue)** : gère la disposition et l'affichage.
- **Controller (contrôleur)** : achemine les commandes des parties "model" et "view".

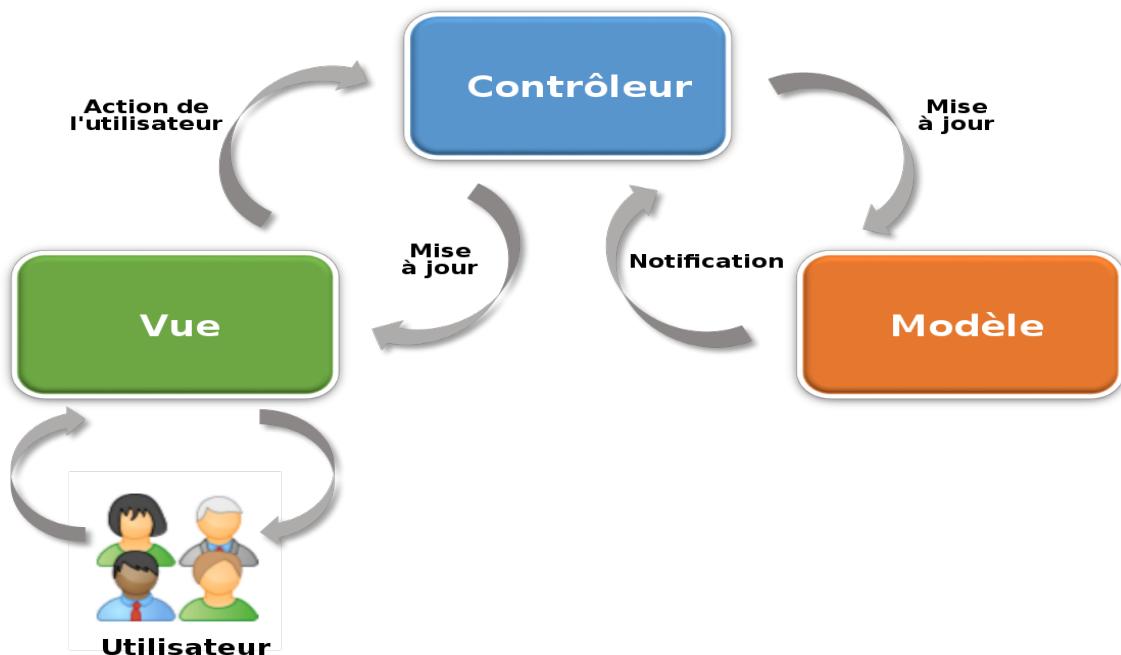


FIGURE 2.3 – Architecture logicielle MVC

2.4 Environnement de travail

L'analyse des besoins techniques couvre la spécification des technologies à utiliser ainsi que la structure applicative pour donner une vision globale sur l'architecture de notre projet. Avant de se lancer dans la conception et l'implémentation de notre projet nous allons décrire l'environnement et les outils du travail.

nous allons commencer par définir l'environnement matériel, ainsi que l'environnement de développement.

2.4.1 Environnement matériel

L'environnement matériel utilisé durant la réalisation de notre projet est illustré par la table 2.4.

TABLE 2.4 – Environnement Matériel

Fabricant	Processeur	Mémoire	Système d'exploitation
Dell	Core i7	16 GO de RAM	Ubuntu

2.4.2 Environnement Logiciel

La détermination de l'environnement logiciel est une étape importante qui influence directement la réalisation et le succès du projet. Dans la partie suivante, nous présenterons les environnements logiciels utilisés.

PhpStorm

PhpStorm est un environnement de développement intégré (IDE) spécifiquement conçu pour le développement d'applications en PHP. Il est développé par JetBrains L'IDE offre une intégration étroite avec les outils de gestion de versions tels que Git, Mercurial et SVN, ce qui facilite la collaboration sur les projets. Il prend en charge les frameworks populaires tels que Symfony, Laravel, WordPress, Drupal, etc., et offre des fonctionnalités spécifiques à ces frameworks. [3]



FIGURE 2.4 – Phpstorn.

GitLab

GitLab est une plateforme de gestion de code source très complète qui offre une multitude de fonctionnalités pour le développement collaboratif. Son principal avantage réside

dans la combinaison du contrôle de version décentralisé avec Git, permettant aux développeurs de travailler ensemble sur un même projet, de suivre l'historique des modifications et de fusionner facilement les branches de code. GitLab intègre également des fonctionnalités avancées d'intégration continue et de déploiement continu (CI/CD), automatisant ainsi les tests, les builds et les déploiements d'applications. [4]



FIGURE 2.5 – GitLab.

2.4.3 Langages de programmation

Notre choix de langage de programmation est basé sur une évaluation approfondie des avantages et des inconvénients de chaque langage. Il est essentiel de sélectionner un langage en tenant compte des besoins spécifiques de l'entreprise afin d'éviter les pertes de temps liées à un changement de langage en cours de projet. Pour la réalisation de notre solution, nous avons opté pour Symfony, un framework PHP, pour la partie backend, et Vue.js, un framework JavaScript, pour le frontend. Cette combinaison de technologies nous permet de bénéficier des fonctionnalités avancées offertes par Symfony pour la gestion de la logique métier et du traitement des requêtes, ainsi que de la flexibilité et de la réactivité de Vue.js pour la création d'interfaces utilisateur dynamiques et interactives.

Symfony

Symfony est un framework PHP open-source reconnu pour sa stabilité, sa modularité et son efficacité dans le développement web. Grâce à sa structure organisée basée sur le modèle MVC (Modèle-Vue-Contrôleur), Symfony permet une séparation claire des responsabilités et facilite la maintenabilité du code. Son écosystème riche et dynamique comprend une vaste collection de composants réutilisables qui couvrent une variété de fonctionnalités couramment utilisées, allant de la gestion des formulaires à la sécurité en passant par la manipulation des fichiers. Cela accélère le processus de développement en offrant des solutions préconstruites et en permettant aux développeurs de se concentrer sur les aspects spécifiques de leur application.[5]



FIGURE 2.6 – Symfony.

Vue.js

Vue.js est un framework JavaScript progressif et réactif, conçu pour créer des interfaces utilisateur dynamiques. Sa syntaxe simple et intuitive facilite l'apprentissage et l'utilisation du framework. Grâce à son approche progressive, Vue.js peut être intégré progressivement dans un projet existant. Il offre une réactivité optimale en suivant les changements de données et mettant à jour automatiquement l'interface utilisateur. La modularité est encouragée grâce à la création de composants réutilisables, ce qui favorise la structuration du code. L'écosystème de Vue.js est riche en plugins, bibliothèques et outils complémentaires qui étendent ses fonctionnalités. Vue.js est apprécié pour sa flexibilité, sa performance et son adoption rapide par la communauté des développeurs.[6]



FIGURE 2.7 – Vue.js.

Conclusion

Dans ce chapitre, nous avons identifié et clarifié les besoins du projet, ce qui nous a permis d'avoir une vision plus précise de ce qui est attendu. Nous avons également présenté l'environnement de travail et l'architecture logicielle choisie. Le prochain chapitre sera consacré à la présentation du travail réalisé lors du premier sprint

Chapitre 3

Sprint 1 : Authentification et gestion des articles, collectes et des tournées

“ Celui qui déplace la montagne, c'est celui qui commence par enlever les petites pierres.”

Confucius

Introduction

Dans le chapitre précédent nous avons eu l'occasion de spécifier les besoins fonctionnels, comprendre l'architecture du système et l'environnement dans lequel le projet sera mis en ouvre. Par conséquent, cette partie du projet sera consacrée à la conception et la développement de notre application.

3.1 Backlog du sprint 1

Dans cette partie, nous décrivons le backlog du produit pour le premier sprint qui consiste à une liste de fonctionnalités à réaliser. Elles sont exprimées sous forme des besoins est priorisées par le Product Owner ce qui permet d'établir un ordre à respecter lors de la réalisation de l'application. Notre sprint backlog se présente comme suit :

TABLE 3.1 – Backlog du produit - Sprint 1

ID	User Stories	Priorité
1	En tant qu'utilisateur , je veux m'authentifier	1
2	En tant qu'utilisateur , je veux créer, consulter, modifier, supprimer les collectes et consulter la carte des collectes	2
3	En tant qu'utilisateur , je veux créer, exporter, modifier, supprimer et consulter les articles	2
4	En tant qu'utilisateur , je veux ajouter, modifier, consulter et annuler les tournées de collecte, Affecter ou supprimer une collecte à une tournée	3

3.2 Analyse et conception

3.2.1 Diagramme de cas d'utilisation

Dans la figure 3.1, nous illustrons le diagramme des cas d'utilisations pour ce premier sprint.

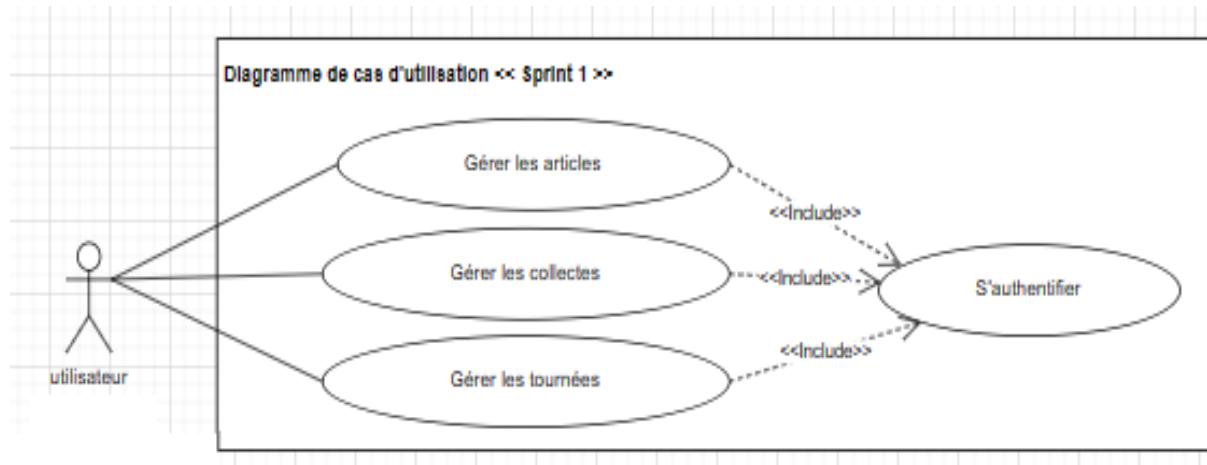


FIGURE 3.1 – Diagramme de cas d'utilisation du sprint 1

3.2.2 Raffinement de cas d'utilisation Gérer collecte

Dans la figure 3.2, nous illustrons le diagramme de raffinement de cas d'utilisation "Gérer collecte", l'utilisateur dispose de plusieurs fonctionnalités. Pour ajouter une demande de collecte, il est obligatoire d'associer un donneur à cette collecte. De plus, il peut visualiser la liste complète des collectes disponibles et avoir la flexibilité de consulter, modifier ou supprimer une collecte spécifique à partir de cette liste. En outre, l'utilisateur a également la possibilité de consulter la carte des collectes pour une visualisation géographique.

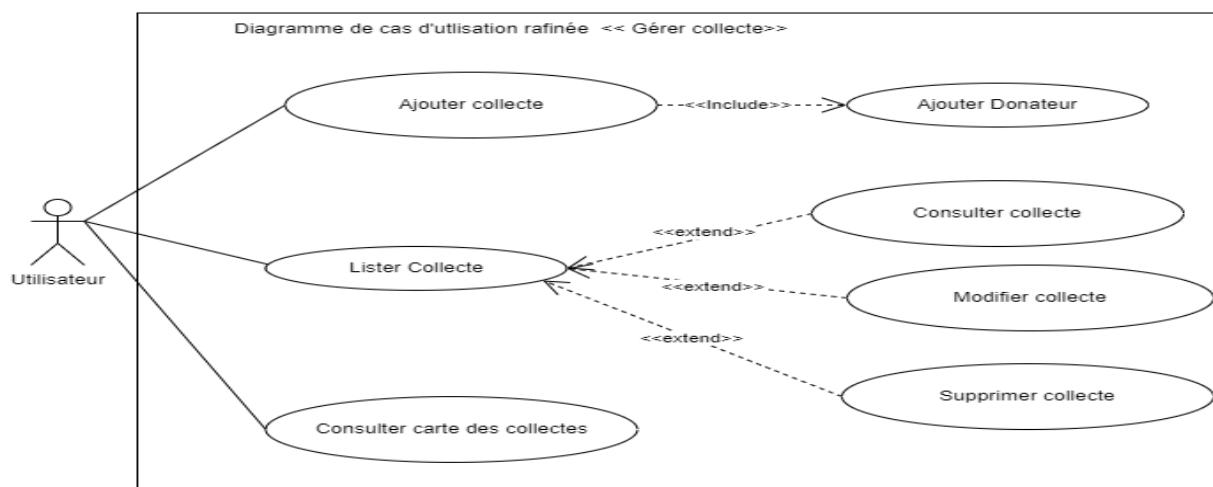


FIGURE 3.2 – Cas d'utilisation Gérer collecte

3.2.3 Raffinement de cas d'utilisation Gérer article

Durant cette section nous allons raffiner le cas d'utilisation Gérer article en donnant le diagramme des cas d'utilisation

Dans le diagramme représenté dans la figure 3.3, l'utilisateur a la possibilité d'ajouter un article. Cependant, pour pouvoir ajouter cet article, il doit préalablement lister les catégories disponibles afin de pouvoir lui attribuer une catégorie spécifique. L'utilisateur peut également lister les articles existants. À travers cette fonctionnalité, il a la flexibilité de consulter, modifier, supprimer ou exporter un article selon ses besoins

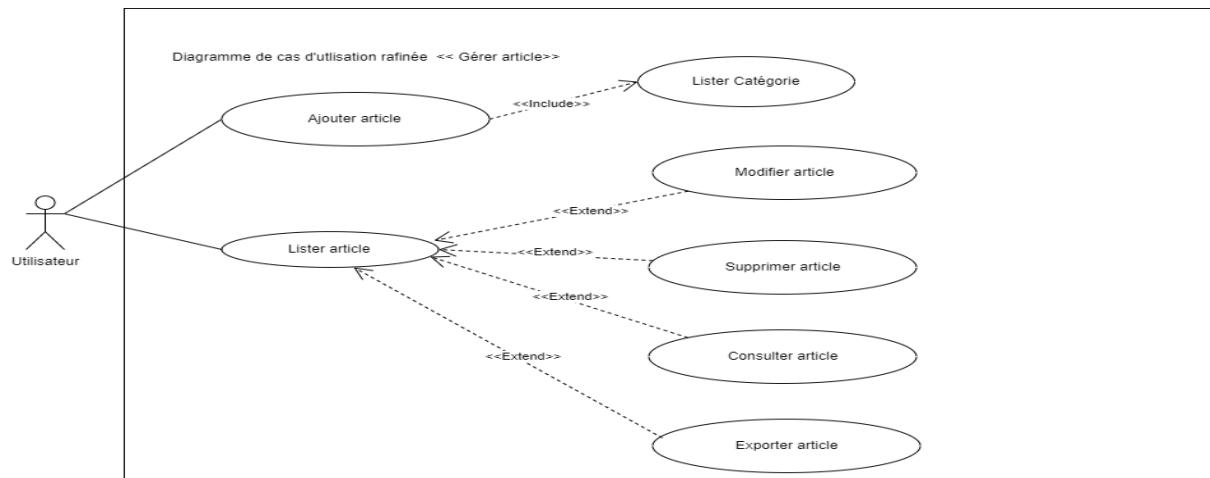


FIGURE 3.3 – Diagramme de Cas d'utilisation Gérer article.

3.2.4 Raffinement de cas d'utilisation Gérer tournée

Durant cette section, nous allons raffiner le cas d'utilisation "Gérer tournées" en fournissant le diagramme de cas d'utilisation.

Dans le diagramme représenté dans la figure 3.4 , l'utilisateur dispose de plusieurs fonctionnalités liées aux tournées. Il peut ajouter, supprimer, consulter et modifier une tournée spécifique. Lors de la modification d'une tournée, l'utilisateur a la possibilité d'affecter ou de révoquer une collecte pour cette tournée. Cela signifie qu'il peut associer une collecte à la tournée ou la retirer de la tournée en fonction de ses besoins ou des changements de planification.

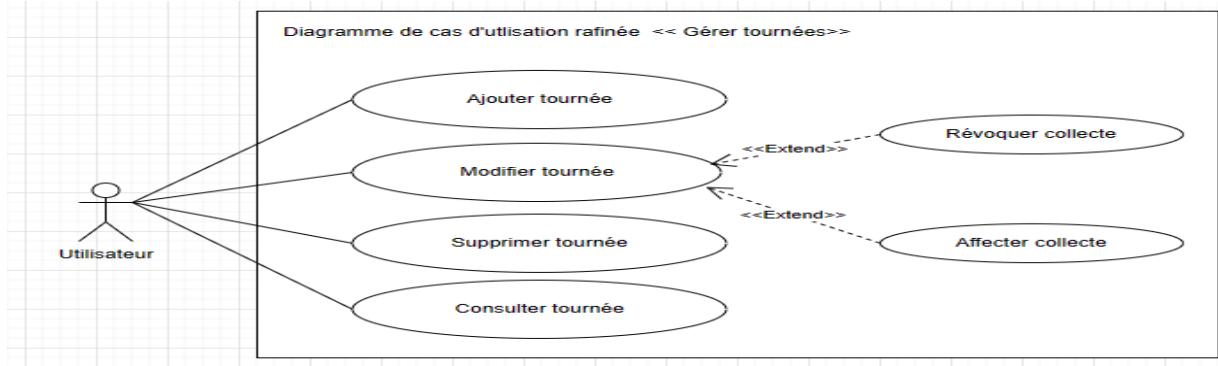


FIGURE 3.4 – Diagramme de Cas d'utilisation Gérer tournée.

3.2.5 Diagrammes de séquence

Dans cette partie, nous avons choisi d'utiliser les diagrammes de séquence et la description textuelle des cas d'utilisations pour expliquer le fonctionnement de notre application de manière claire et détaillée. Ces deux outils permettent de représenter visuellement les interactions entre les différents acteurs et composants de notre système.

Diagramme de séquence Authentification

Le diagramme de séquence "Authentification" dans la figure 3.5 représente les interactions entre l'utilisateur et le système lors du processus d'authentification dans l'application.

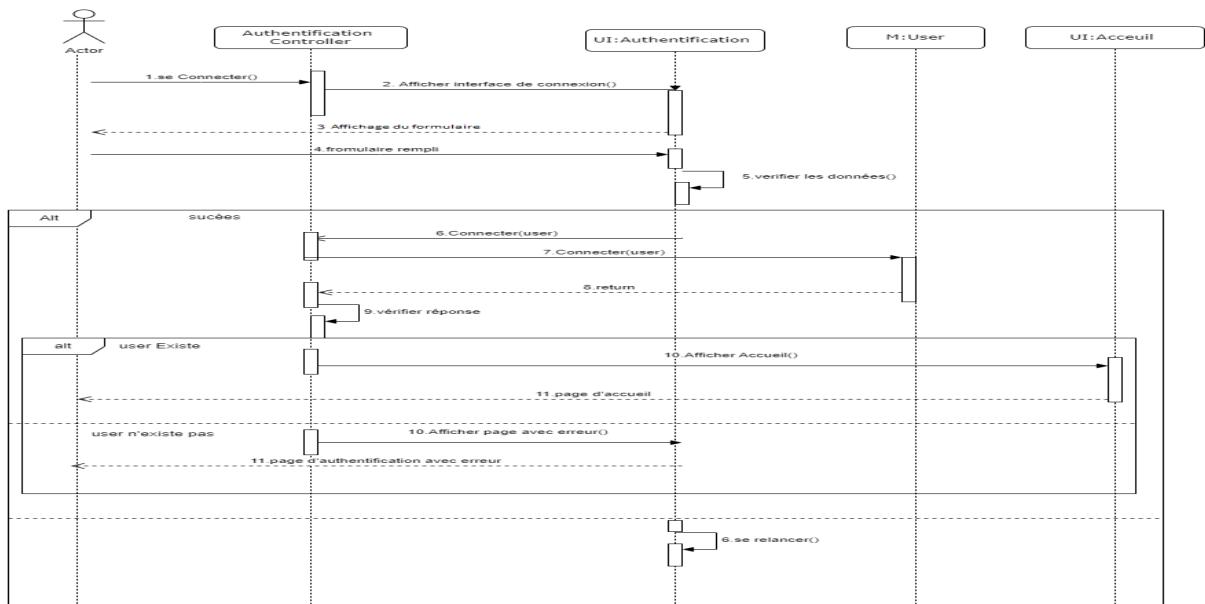


FIGURE 3.5 – Diagramme de séquence Authentification.

Description textuelle d'authentification

La table 3.2 représente la description textuelle d'authentification avec les différentes étapes et scénarios liés à l'authentification d'un utilisateur.

TABLE 3.2 – Description textuelle d'authentification

Cas d'utilisation	Authentification d'un utilisateur
Acteur	Utilisateur
Précondition	Utilisateur non authentifié
Post-condition	Utilisateur authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page de connexion ou d'inscription du système. 2. Le système affiche le formulaire d'authentification avec des champs pour saisir les identifiants (nom d'utilisateur et mot de passe). 3. L'utilisateur saisit ses identifiants dans les champs appropriés. 4. Le système vérifie les informations saisies par l'utilisateur en les comparant avec celles enregistrées dans la base de données des utilisateurs. 5. Si les informations d'identification sont correctes, le système authentifie l'utilisateur et lui accorde l'accès aux fonctionnalités réservées aux utilisateurs authentifiés. 6. Le système redirige l'utilisateur vers la page d'accueil ou la page demandée après l'authentification réussie.
Scénarios alternatifs et exceptions	<p>3.1 L'utilisateur a oublié ses identifiants. Le système fournit une option de récupération de mot de passe où l'utilisateur peut demander un lien de réinitialisation de mot de passe via l'e-mail enregistré.</p> <p>4.1 Les informations d'identification fournies par l'utilisateur sont incorrectes. Le système affiche un message d'erreur indiquant que les identifiants sont invalides et demande à l'utilisateur de les saisir de nouveau.</p>

Diagramme de séquence Ajouter article.

La figure 3.6 illustre la séquence des interactions entre l'utilisateur et les différentes couches de l'application lors de la création d'un article.

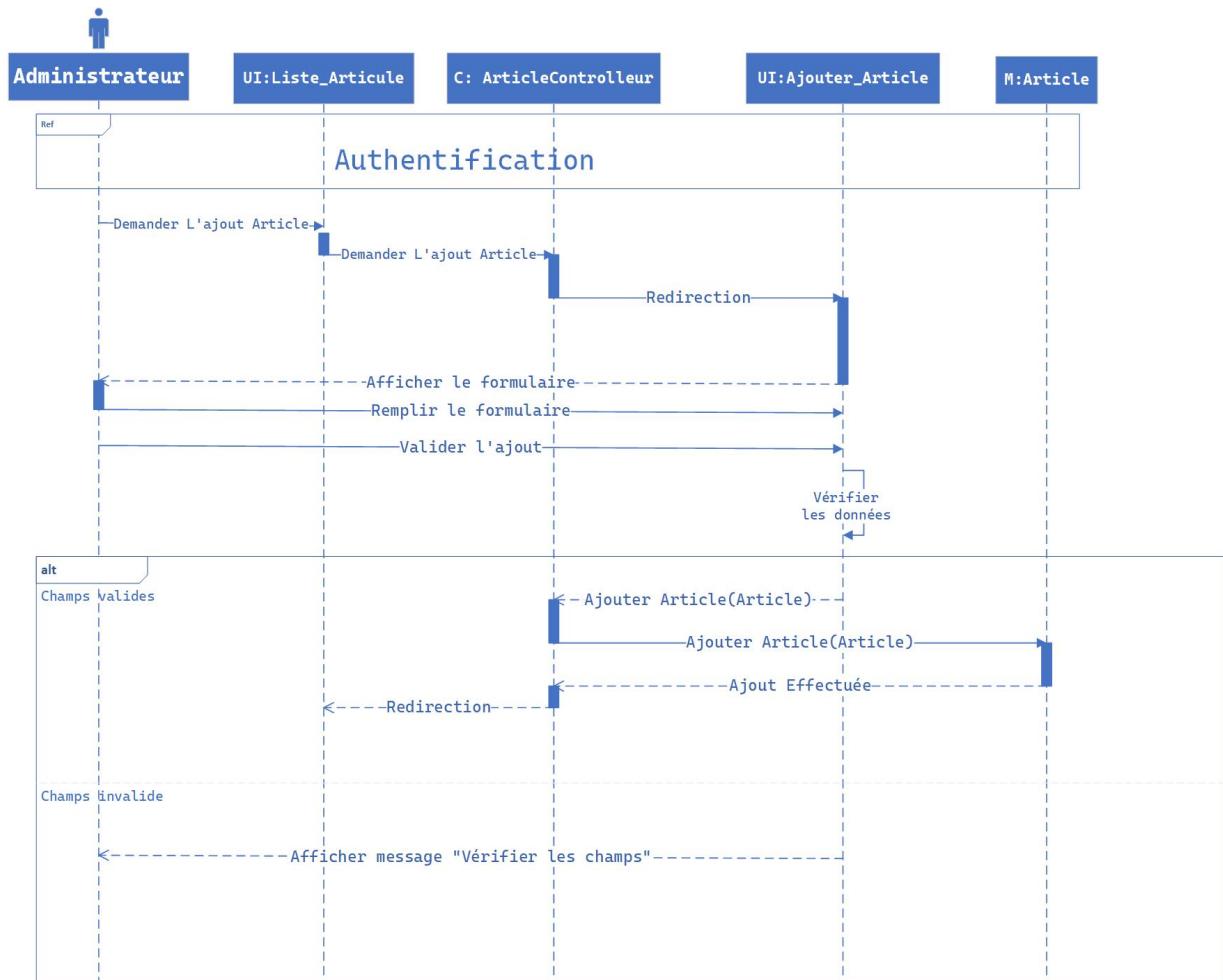


FIGURE 3.6 – Diagramme de séquence Ajouter un article.

Description textuelle Ajouter un article

La table 3.3 représente la description textuelle de cas d'utilisation "Ajouter article" avec les étapes et les actions nécessaires pour créer un nouvel article.

TABLE 3.3 – Description textuelle Ajouter un article

Cas d'utilisation	Ajouter un article
Acteur	Utilisateur
Précondition	Utilisateur authentifié
Post-condition	Article créé
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page de création d'articles. 2. Le système affiche le formulaire de création. 3. L'utilisateur remplit le formulaire en fournit les informations requises. 4. Le système vérifie les informations saisies par l'utilisateur. 5. Le système enregistre l'article dans la base de données. 6. Le système affiche un message indiquant que l'article a été créé avec succès.
Scénarios alternatifs et exceptions	<ol style="list-style-type: none"> 3.1. L'utilisateur peut annuler la création de l'article à tout moment. 4.1. Les données saisies par l'utilisateur sont incorrectes ou manquantes : <ol style="list-style-type: none"> 4.1.1 Le système affiche un message d'erreur spécifiant les problèmes rencontrés et demande à l'utilisateur de corriger les informations saisies.

Diagramme de séquence Consulter article.

Le diagramme de séquence "Consulter article" dans la figure 3.7 illustre la séquence d'actions entre l'utilisateur et le système lorsqu'il souhaite consulter un article spécifique.

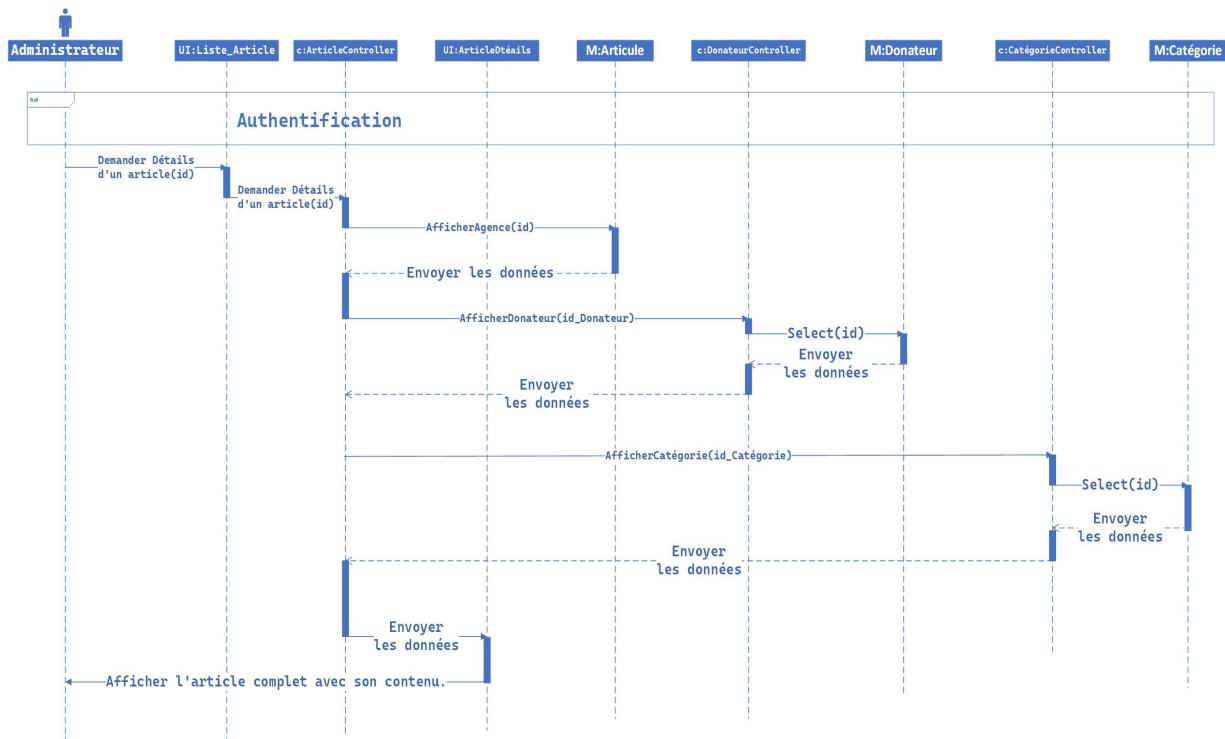


FIGURE 3.7 – Diagramme de séquence Consulter Article.

Description textuelle Consulter article

Le tableau 3.4 détaille les différentes étapes et actions requises pour permettre à l'utilisateur de consulter un article dans le système.

TABLE 3.4 – Description textuelle Consulter article

Cas d'utilisation	Consulter un article
Acteur	Utilisateur
Précondition	Utilisateur authentifié
Post-condition	Article affiché
Scénario nominal	<ol style="list-style-type: none"> L'utilisateur accède au site web. Le système affiche la liste d'articles disponibles. L'utilisateur sélectionne l'article qu'il souhaite consulter en cliquant dessus. Le système affiche l'article complet avec son contenu.

3.3 Conception

La conception vise à transformer le modèle d'analyse en un modèle concret à partir duquel le développeur peut directement implémenter le système. Nous commençons alors par présenter le diagramme de classes.

Diagramme de classes

Ce diagramme est considéré comme la partie la plus importante de la modélisation orientée Objet. Le diagramme de classes montre la structure interne de l'application, il permet de fournir une représentation abstraite des objets système qui vont interagir pour la réalisation des cas d'utilisation.

La figure 3.8 illustre le diagramme de classe de notre système durant notre premier sprint.

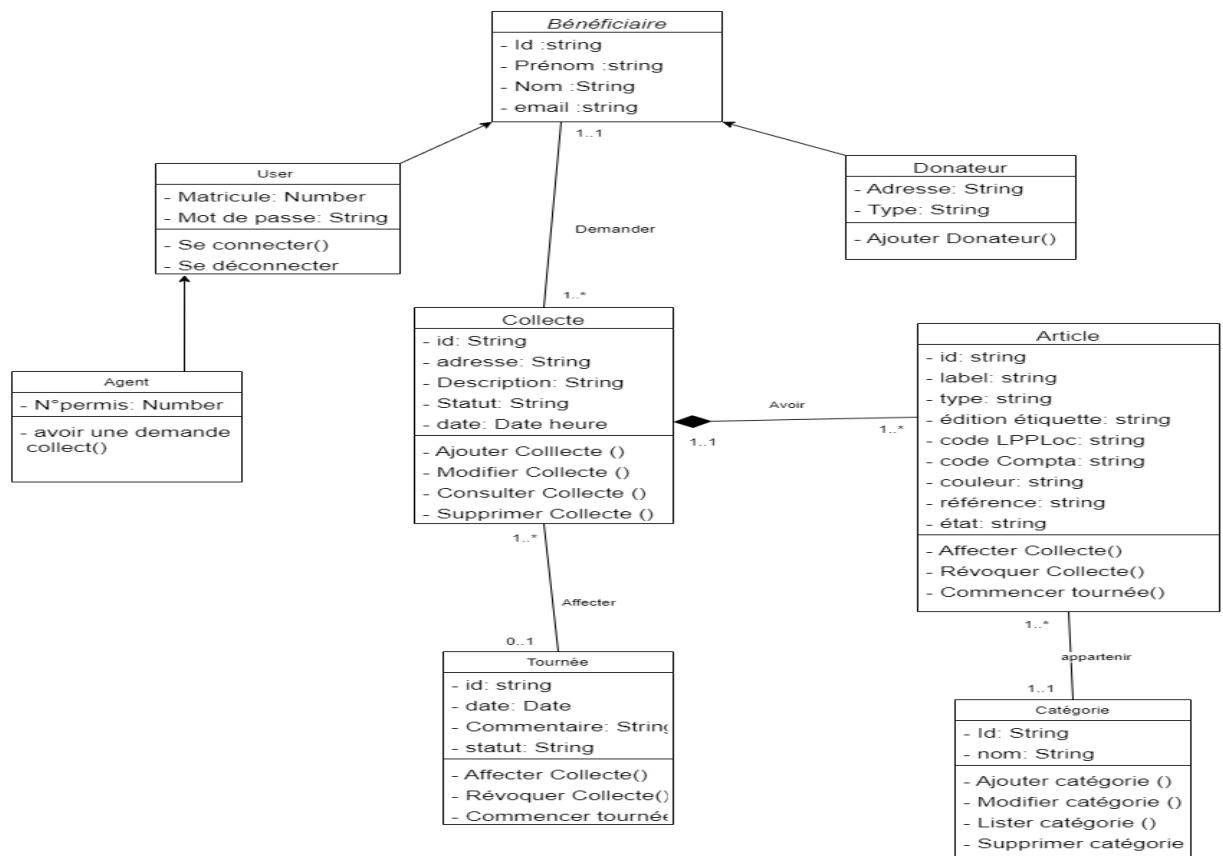


FIGURE 3.8 – Diagramme de classe de sprint 1

Description des classes

- Classe article : cette classe contienne les attributs et les méthodes relatifs à un article.
- Classe catégorie : cette classe contienne les attributs et les méthodes relatifs à une catégorie. Une catégorie peut appartenir à un ou plusieurs article(s).
- Classe collecte : cette classe contienne les attributs et les méthodes relatifs à une demande de collecte. Une demande de collecte doit avoir minimum un article à collecter.
- Classe tournée : cette classe contienne les attributs et les méthodes relatifs à une tournée. Une tournée peut contenir un ou plusieurs collecte(s).
- Classe Bénéficiaire : cette classe contienne tous les attributs communs de tous types de personnes qui sont en relation avec note plateforme.
- Classe donateur : cette classe représente les attributs et les méthodes communs d'une personne qu'elle va donner un article.
- Classe agent : cette classe représente les attributs et les méthodes communs d'une personne chargée à récupérer les collectes.
- Classe utilisateur : cette classe représente les attributs et les méthodes communs de la personne qui lance les demandes de collecte.

3.4 Réalisation

Dans cette partie nous allons présenter des captures d'écrans des parties fonctionnelles de notre application développée pendant le sprint 1.

3.4.1 Interface d'authentification

La figure 3.9 ci-dessus représente l'interface d'authentification,



FIGURE 3.9 – Interface d'authentification

Cette interface permet à chaque utilisateur et administrateur d'accéder à l'application en fonction de leurs rôles et de leurs privilèges spécifiques. Chacun peut s'authentifier en utilisant son adresse e-mail et son mot de passe.

3.4.2 Gestion des articles

Créer un nouvel article

La figure 3.10 représente l'interface "Ajouter article"

The screenshot shows the 'Ajouter un nouveau article' (Add new item) screen. On the left, there's a sidebar with 'Articles' selected. The main area has a search bar and a user profile. The form is divided into six sections: 1. Général (General), 2. Identification de l'article (Article identification), 3. Attributs (Attributes), 4. Prix et type de transaction (Price and transaction type), 5. Poids (Weight), and 6. Informations complémentaires (Additional information). The 'Nom de l'article' field is currently active, showing a tooltip about entering a technical name. Each section has an 'Enregistrer' (Save) button at the bottom.

FIGURE 3.10 – Interface Ajouter un article

L'interface "Ajouter article" permet aux utilisateurs d'ajouter un article. Pour faciliter le processus d'ajout, nous avons regroupé les champs à remplir en fonction de leur nature. Par exemple, les informations générales sont regroupées ensemble, les identifications sont regroupées ensemble, et les prix sont également regroupés ensemble. Cette organisation vise à simplifier et à rationaliser le processus d'ajout en permettant aux utilisateurs de saisir les informations de manière structurée et intuitive.

Consulter la liste des articles :

La figure 3.11 montre l'interface " consulter la liste des articles"

FIGURE 3.11 – Interface consulter liste des articles

L'utilisateur peut visualiser la liste des articles avec leurs catégories. À partir de cette liste, il a plusieurs options. Il peut exporter les articles. Il peut également modifier, supprimer ou consulter un article spécifique en sélectionnant l'option correspondante pour l'article souhaité. De plus, l'utilisateur a la possibilité d'effectuer des recherches d'articles en utilisant des mots-clés ou des critères spécifiques. Il peut également trier les résultats en fonction de différents attributs tels que le nom ou la catégorie. Enfin, l'utilisateur peut filtrer les résultats en utilisant des filtres prédéfinis ou en définissant ses propres critères de filtrage pour affiner sa recherche

Modifier un article :

La figure 3.12 illustre L'interface de modification d'un article existant

DONNÉES	VALEURS
Type de transaction	Vente + Ajouter
Fabricant	Nom du fabricant
Catégorie	Nom de la catégorie
Attributs	Couleur + Ajouter Dimension + Ajouter
Code LPPs	CodeLPP1 + Ajouter CodeLPP2 + Ajouter CodeLPP3 + Ajouter
Prix Public Fabricant	Valeur
Prix de vente particulier	Valeur
Prix de vente professionnel	Valeur
Prix de vente interagence	Valeur
Code UDI	Valeur
Code ACL	Valeur
Poids	Valeur en kg
Poids Maximum	Valeur en kg
Fiche de Mesure	URL
Numéro de série requis	Oui
Marquage CE requis	Non

FIGURE 3.12 – Interface modifier article

L'interface de modification d'un article existant permet à l'utilisateur de mettre à jour les informations d'un article déjà enregistré dans le système. elle présente les champs pertinents de l'article, tels que le nom, la description, la catégorie, le prix, etc., pré-remplis avec les données existantes de l'article. L'utilisateur peut modifier les champs nécessaires.

3.4.3 Gestion des collectes

Créer une demande de collecte

La figure 3.13 montre l'interface de création d'une demande de collecte :

3.4. Réalisation

34

FIGURE 3.13 – Interface création demande de collecte

Cette interface propose un processus guidé par étapes. L'utilisateur remplit les informations du donneur, la disponibilité pour la collecte et le point de collecte. Une fois toutes les étapes sont complétées, l'utilisateur peut vérifier par le récapitulatif des informations avant d'enregistrer la demande.

Consulter collecte

La figure 3.14 montre l'interface "consulter collecte" :

FIGURE 3.14 – Interface consulter collecte

L'interface "Consulter collecte" présente les détails d'une collecte spécifique. Elle affiche

des informations clés telles que la date de la demande, la description, la date, le lieu, le responsable et le statut de la collecte.

Lister les demandes de collectes

La figure 3.15 permet à l'utilisateur de Lister les demandes de collectes :

DATE DE SAISIE	DÉPARTEMENT	ÉTAT	STATUT	TYPE DE DONNEUR	ADRESSE	ACTIONS
12 avril 2022	Département	Urgent	Nouvelle demande	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Collecte à venir	Particulier	Adresse de la collecte	
12 avril 2022	Département	Normal	Collecte terminée	Particulier	Adresse de la collecte	
12 avril 2022	Département	Normal	Collecte à prévoir	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Nouvelle demande	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Nouvelle demande	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Collecte à venir	Particulier	Adresse de la collecte	
12 avril 2022	Département	Normal	Collecte terminée	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Nouvelle demande	Particulier	Adresse de la collecte	
12 avril 2022	Département	Normal	Collecte à prévoir	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Nouvelle demande	Particulier	Adresse de la collecte	
12 avril 2022	Département	Urgent	Nouvelle demande	Particulier	Adresse de la collecte	

FIGURE 3.15 – Interface liser les collectes

L'utilisateur a accès à la liste des collectes disponibles. À partir de cette liste, il peut consulter les détails de chaque collecte. L'utilisateur a également la possibilité de réaliser des actions supplémentaires, telles que l'exportation des collectes, la modification des informations ou la suppression d'une collecte spécifique. De plus, il peut effectuer des recherches ciblées en utilisant des mots-clés ou des filtres spécifiques, afin de trouver rapidement les collectes qui l'intéressent.

3.4.4 Gestion Tournées

Planifier une Tournée

La figure 3.16 montre l'interface de planification d'une tournée :

3.4. Réalisation

36

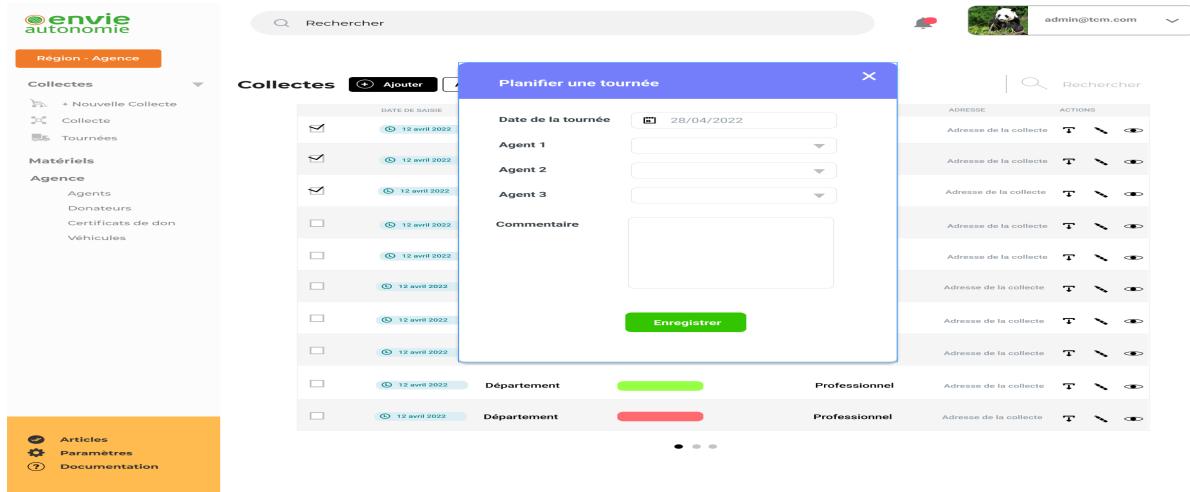


FIGURE 3.16 – Interface planifier tournée

Dans cette interface, l'utilisateur a la possibilité de planifier une tournée. Il peut sélectionner la date de la tournée et affecter cette tournée à un ou plusieurs agents. De plus, l'utilisateur peut ajouter des commentaires pertinents concernant la tournée, tels que des instructions spéciales ou des informations supplémentaires.

Modifier une tournée

La figure 3.17 représente l'interface qui permet à l'utilisateur de modifier une tournée

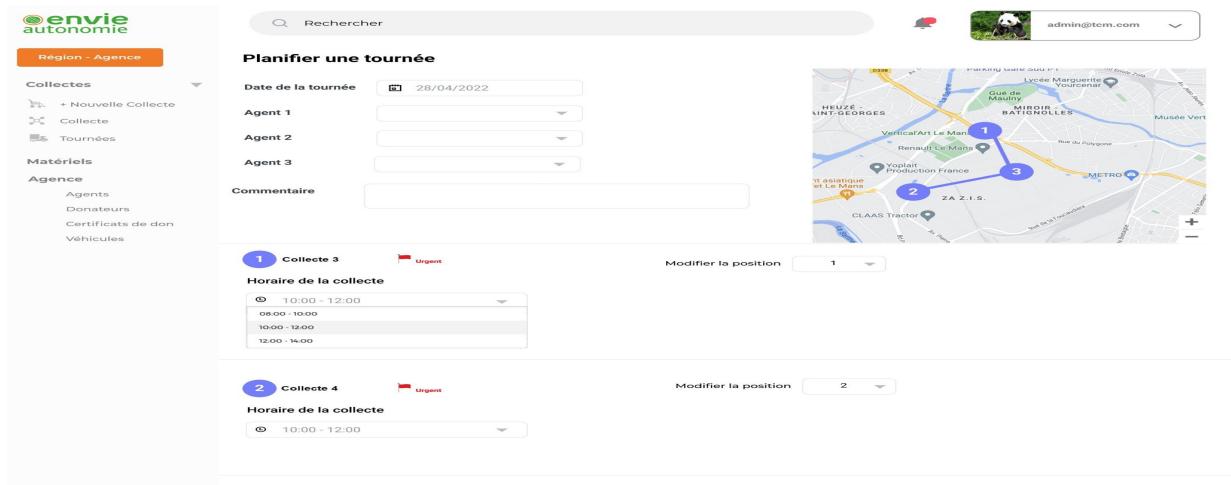


FIGURE 3.17 – Interface modifier Tournée

Dans cette interface, l'utilisateur a la possibilité de modifier une tournée existante. Il peut effectuer des modifications sur les informations de base de la tournée, telles que la date,

les agents assignés et les commentaires. De plus, l'utilisateur peut également affecter de nouvelles collectes à la tournée ou révoquer des collectes déjà assignées. Cette fonctionnalité offre la flexibilité nécessaire pour ajuster et mettre à jour les détails de la tournée en fonction des besoins et des changements survenus.

Consulter carte des collectes

La figure 3.18 montre l'interface "consulter carte des collectes" :

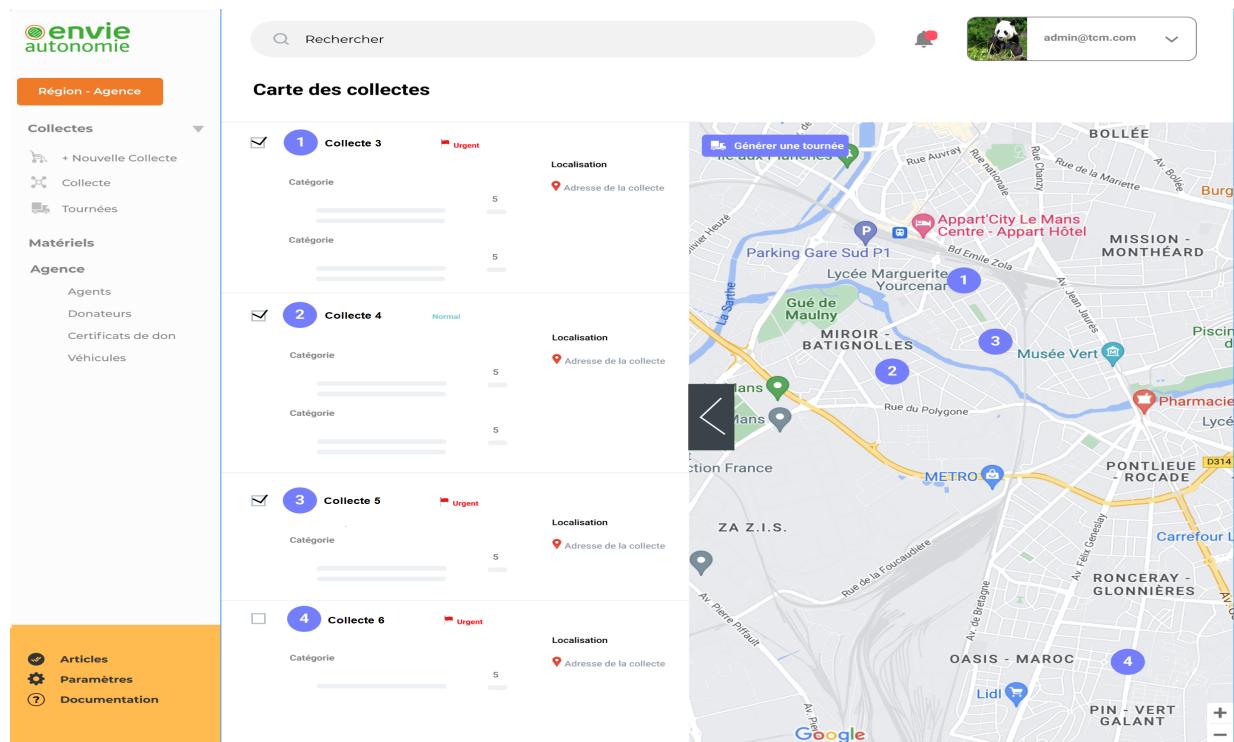


FIGURE 3.18 – Interface consulter carte des collectes

Cette interface permet à l'utilisateur de consulter les différents points de collecte sur une carte en fonction de son adresse. L'utilisateur peut voir les emplacements des points de collecte marqués sur la carte, ainsi que les informations associées à chaque point de collecte, telles que la date, l'heure et la catégorie d'article. Cela permet à l'utilisateur de visualiser facilement les points de collecte disponibles à proximité de son emplacement et d'obtenir les détails pertinents pour planifier les tournées.

Conclusion

Dans ce chapitre nous avons traité les besoins spécifiques du premier sprint , nous avons conçu les différents diagrammes et nous avons réalisé un incrément potentiellement livrable.

Chapitre 4

Sprint 2 : Gestion des rôles ,des agences et suivi des statistiques

“ Il faut être enthousiaste de son métier pour y exceller.”

Denis Diderot

Introduction

Dans le chapitre précédent nous avons eu l'occasion de traiter la première partie de notre application. Par conséquent, cette partie du projet sera consacrée à l'implémentation de Deuxième sprint qui consiste à la conception et le développement de la partie de gestion des rôles, des agences et le suivi des statistiques.

4.1 Backlog du sprint 2

Notre sprint backlog se présente comme suit :

TABLE 4.1 – Backlog du produit - Sprint 2

ID	User Stories	Priorité
5	En tant qu'administrateur ,je veux gérer les rôles des utilisateurs dans la plateforme	1
6	En tant qu'administrateur ,je veux créer, enregistrer, modifier, supprimer et consulter les agences	2
7	En tant qu'administrateur , je veux consulter les statistiques	3

4.2 Analyse et conception

4.2.1 Diagramme de cas d'utilisation

Dans la figure 4.1, nous illustrons le diagramme des cas d'utilisations pour ce deuxième sprint.

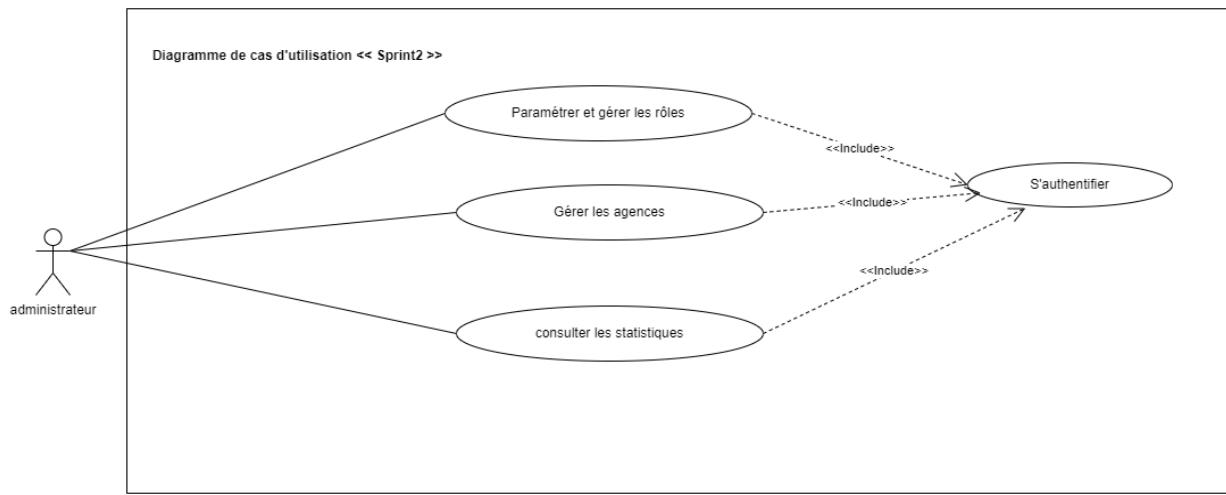


FIGURE 4.1 – Diagramme de cas d'utilisation du sprint 2

4.2.2 Raffinement de cas d'utilisation Gérer rôle

Dans cette section, nous détaillons les actions et les fonctionnalités spécifiques liées à la gestion des rôles.

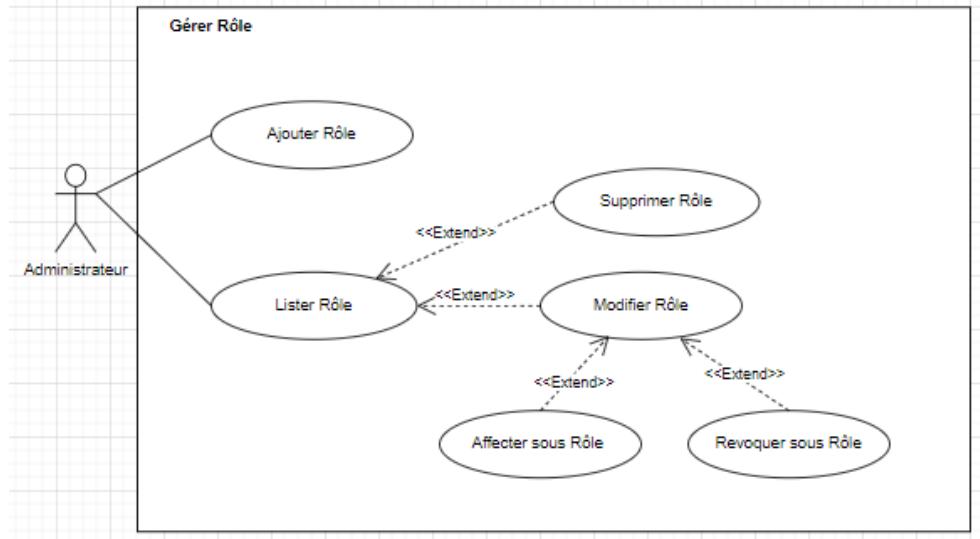


FIGURE 4.2 – Cas d'utilisation gérer rôle

Dans le diagramme ci-dessus, l'administrateur dispose de plusieurs fonctionnalités. Il peut ajouter un rôle. De plus, il peut visualiser la liste complète des rôles disponibles et avoir la flexibilité de supprimer ou modifier un rôle. Lors de la modification d'un rôle, l'administrateur a la possibilité d'affecter ou de révoquer un sous-rôle.

4.2.3 Raffinement de cas d'utilisation Gérer agence

Durant cette section nous allons raffiner le cas d'utilisation Gérer agence.

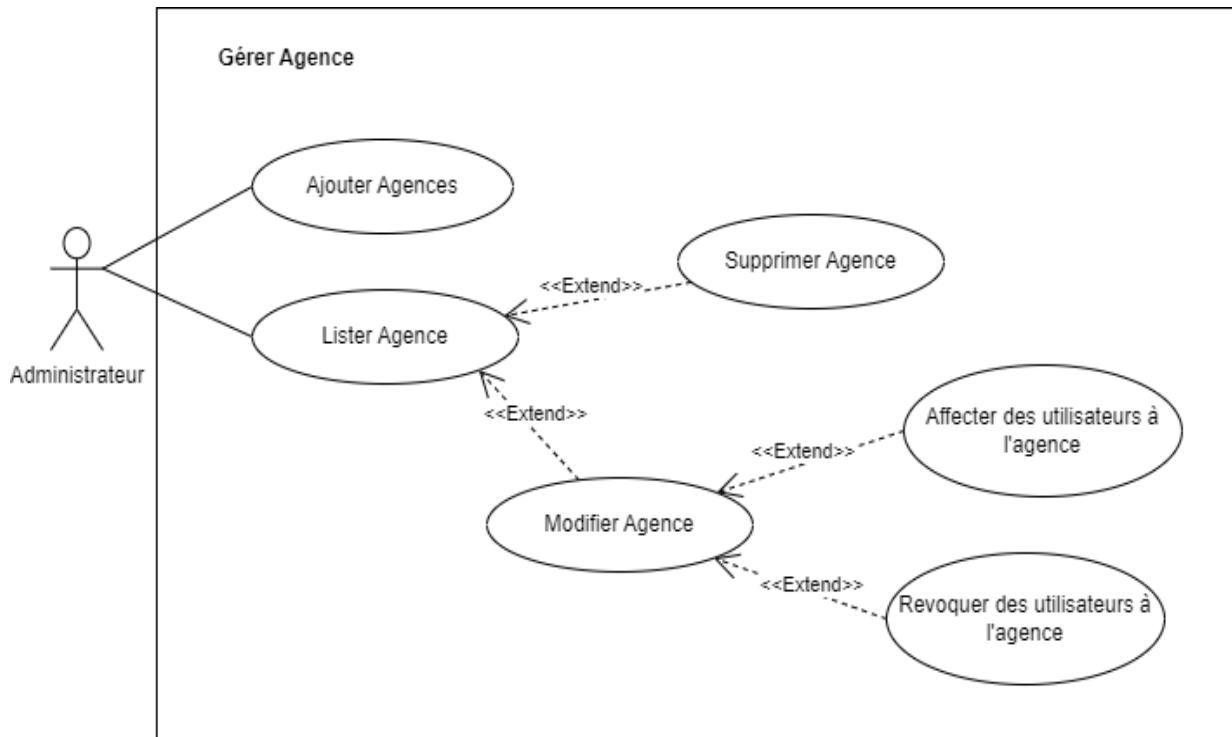


FIGURE 4.3 – Diagramme de Cas d'utilisation gérer agence.

L'administrateur a la possibilité d'ajouter une agence. Il peut également lister les agences existantes. Il peut de supprimer ou modifier une agence. Lors de la modification d'une agence, l'administrateur a la possibilité d'affecter ou de révoquer des utilisateurs à cette agence.

4.2.4 Diagrammes de séquence

Dans cette partie, nous avons choisi d'utiliser les diagrammes de séquence et la description textuelle des cas d'utilisations pour expliquer le fonctionnement de notre application de manière claire et détaillée. Ces deux outils permettent de représenter visuellement les interactions entre les différents acteurs et composants de notre système.

Diagramme de séquence supprimer rôle

Le diagramme de séquence "supprimer rôle" dans la figure 4.4 représente les interactions entre l'administrateur et le système lors du processus de suppression d'un rôles.

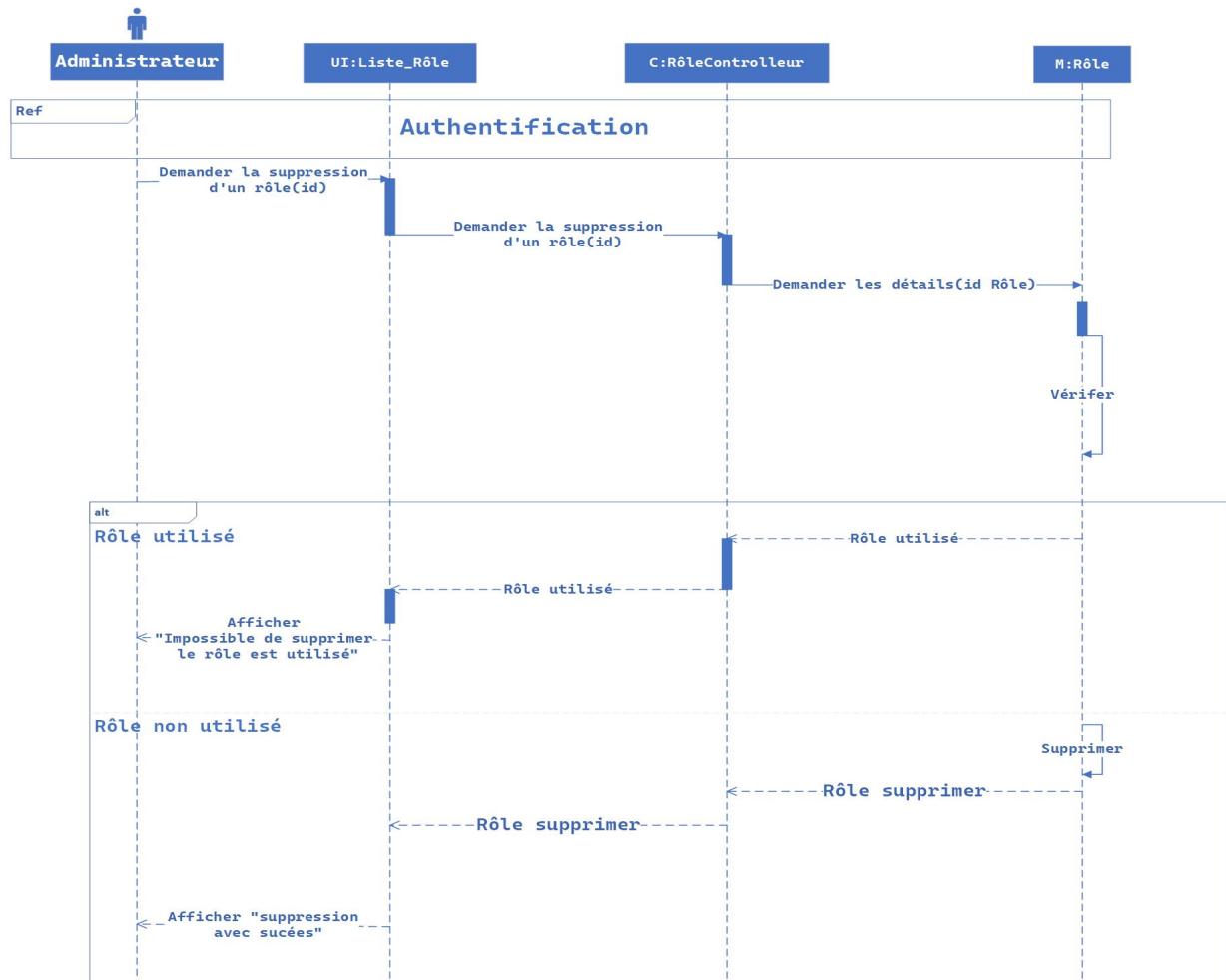


FIGURE 4.4 – Diagramme de séquence supprimer un rôle.

Description textuelle supprimer un rôle

Le tableau 4.2 représente la description textuelle de la suppression d'un rôle avec les différentes étapes et scénarios.

TABLE 4.2 – Description textuelle supprimer un rôle

Cas d'utilisation	supprimer un rôle
Acteur	Adminstrateur
Précondition	adminstrateur authentifié
Post-condition	rôle supprimé
Scénario nominal	<p>1.L'utilisateur accède à la page de gestion des rôles.</p> <p>2. L'utilisateur sélectionne le rôle qu'il souhaite supprimer.</p> <p>3.Le système affiche les détails du rôle sélectionné, y compris ses informations et ses permissions associées.</p> <p>4.L'utilisateur confirme sa volonté de supprimer le rôle en cliquant sur le bouton de suppression.</p> <p>5. Le système vérifie si le rôle peut être supprimé en vérifiant s'il est utilisé par des utilisateurs ou s'il a des dépendances avec d'autres éléments du système.</p> <p>6. Si le rôle ne présente aucune dépendance, le système supprime le rôle de la base de données et de tout lien associé.</p> <p>7.Le système affiche un message de confirmation indiquant que le rôle a été supprimé avec succès.</p>
Scénarios alternatifs et exceptions	<p>4.1. L'utilisateur peut annuler la suppression à tout moment.</p> <p>6.1. Si le rôle présente des dépendances, le système affiche un message d'erreur expliquant pourquoi le rôle ne peut pas être supprimé.</p>

Diagramme de séquence modifier rôle.

La figure 4.5 illustre la séquence des interactions entre l'utilisateur et le système lors de la modification d'un rôle

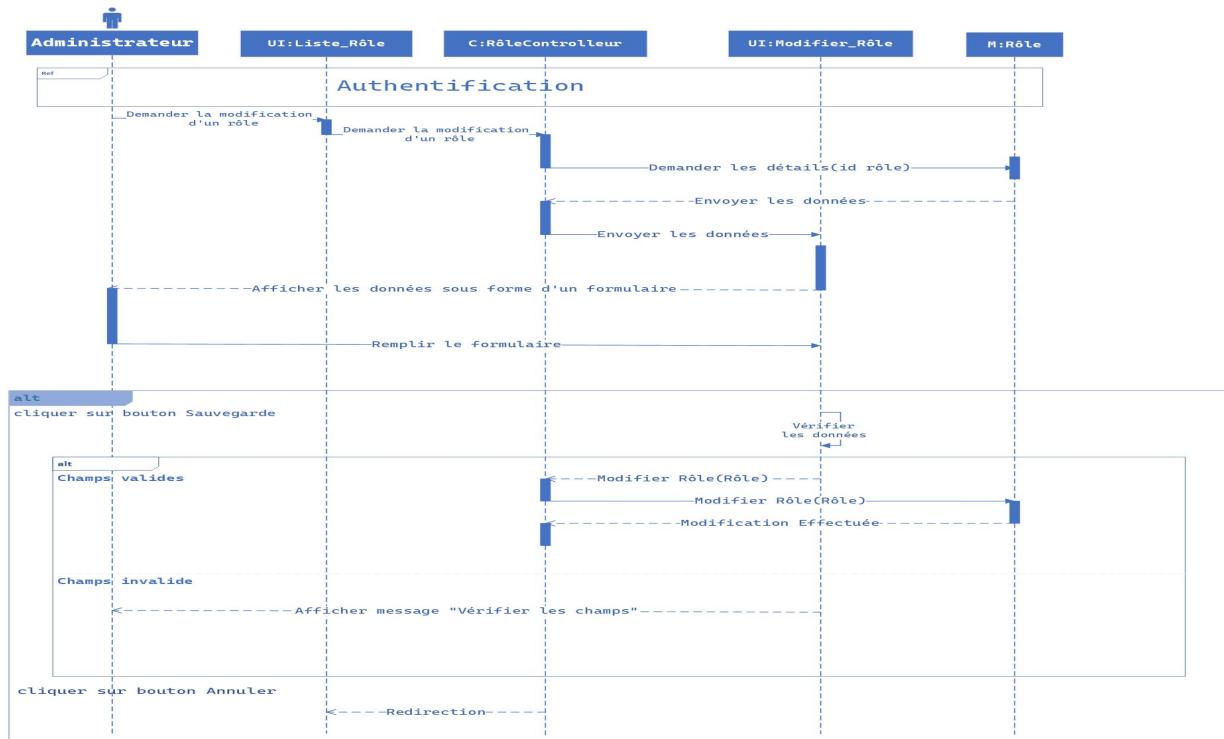


FIGURE 4.5 – Diagramme de séquence modifier rôle.

Description textuelle modifier rôle

Le tableau 4.3 représente la description textuelle de modifier rôle avec les étapes et les actions nécessaires.

TABLE 4.3 – Description textuelle modifier rôle

Cas d'utilisation	modifier rôle
Acteur	Utilisateur
Précondition	Utilisateur authentifié
Post-condition	rôle modifié
Scénario nominal	<p>1.L'utilisateur accède à la page de gestion des rôles.</p> <p>2.L'utilisateur sélectionne le rôle qu'il souhaite modifier.</p> <p>3.Le système affiche les détails du rôle sélectionné, y compris son nom et ses permissions associées..</p> <p>4.L'utilisateur effectue les modifications souhaitées, telles que le changement du nom du rôle ou la modification des permissions.</p> <p>5. Une fois les modifications terminées, l'utilisateur valide en appuyant sur le bouton de sauvegarde.</p> <p>6. Si les données sont valides, le système met à jour les informations du rôle dans la base de données en remplaçant les anciennes valeurs par les nouvelles.</p> <p>7.Le système affiche un message de confirmation indiquant que la modification du rôle a été effectuée avec succès.</p>
Scénarios alternatifs et exceptions	<p>4.1. L'utilisateur peut annuler la modification à tout moment.</p> <p>6.1. Si les données sont invalides le système affiche un message d'erreur.</p>

Diagramme de séquence Consulter agence.

Le diagramme de séquence "Consulter agence" dans la figure 4.6 illustre la séquence d'actions entre l'utilisateur et le système lorsqu'il souhaite consulter une agence spécifique.

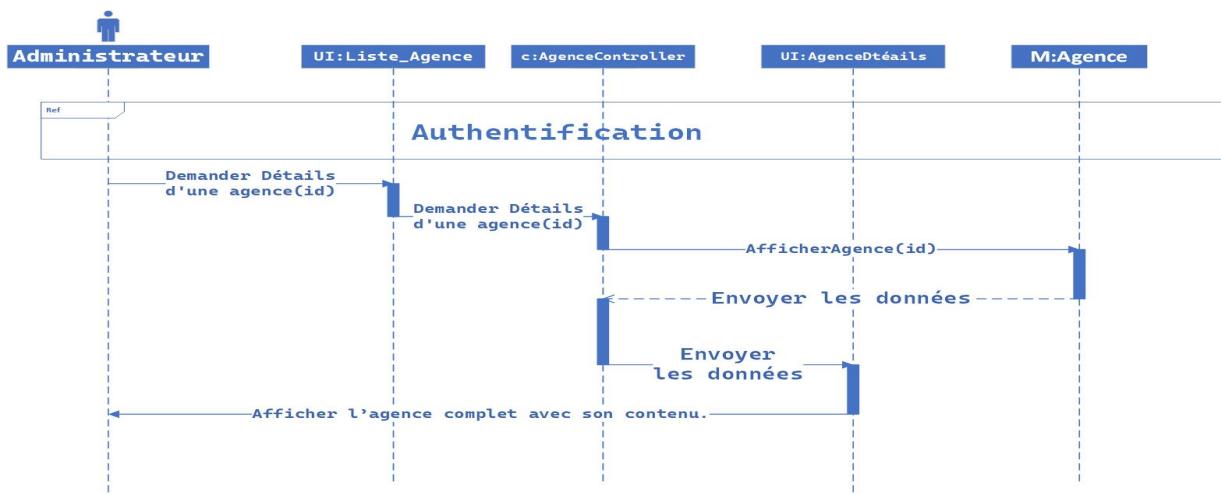


FIGURE 4.6 – Diagramme de séquence Consulter agence.

Description textuelle Consulter agence

La table 4.4 détaille les différentes étapes et actions requises pour permettre à l'utilisateur de consulter une agence.

TABLE 4.4 – Description textuelle Consulter agence

Cas d'utilisation	Consulter agence
Acteur	Utilisateur
Précondition	Utilisateur authentifié
Post-condition	agence affiché
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède au site web. 2. Le système affiche la liste des agences disponibles. 3. L'utilisateur sélectionne l'agence qu'il souhaite consulter en cliquant dessus. 4. Le système affiche l'agence complète avec son contenu.

4.3 Conception

L'objectif de la conception est de prendre le modèle d'analyse et de le transformer en un modèle concret qui peut être directement utilisé par les développeurs pour l'implémentation du système. Une des premières étapes de cette transformation consiste à présenter le diagramme de classes.

Diagramme de classes

La figure 4.7 illustre le diagramme de classe global de notre système durant le deuxième sprint. Les classes traitées dans ce deuxième sprint sont mises en évidence en couleur rose.

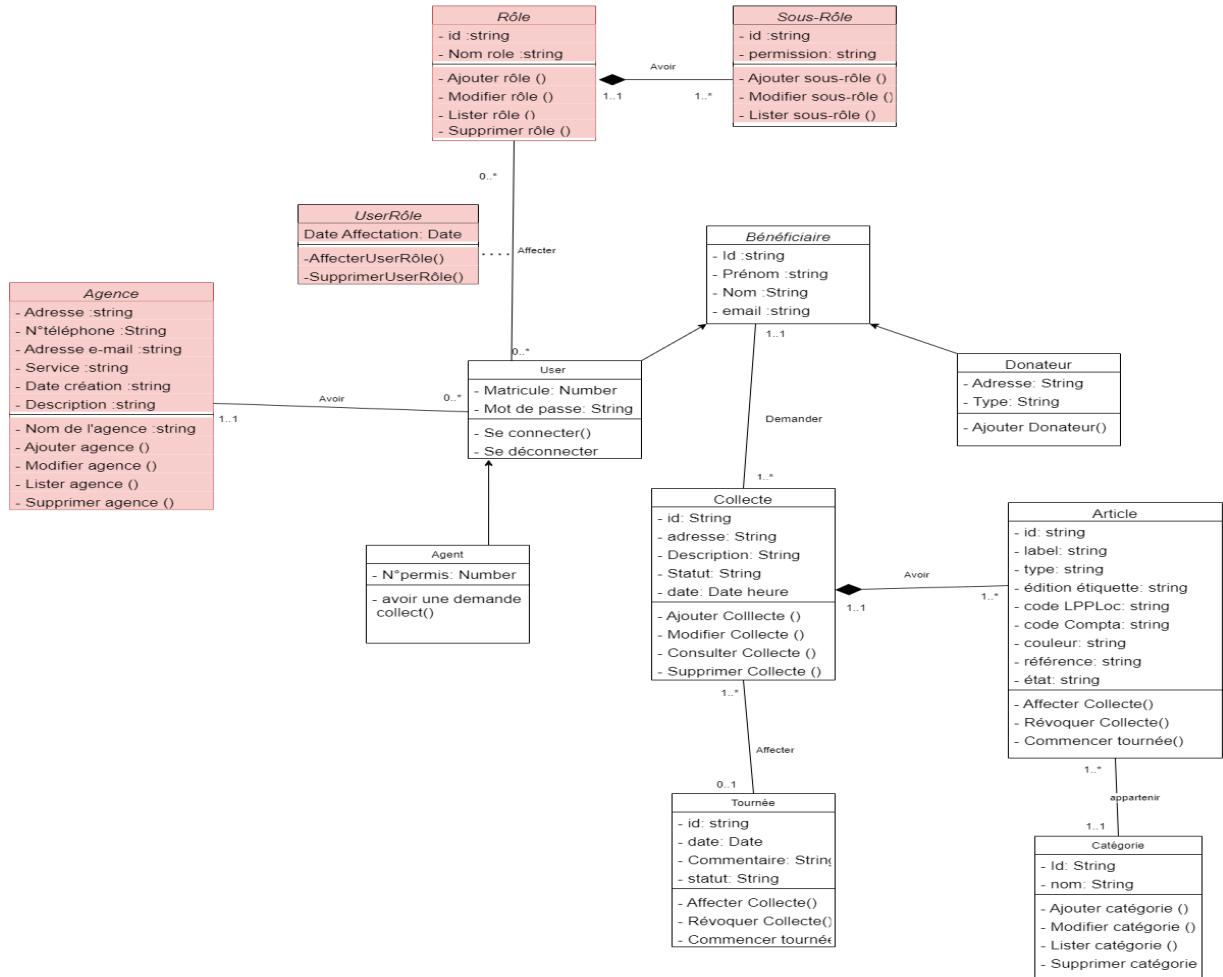


FIGURE 4.7 – Diagramme de classe de l'application

Description des classes

- Classe Agence : Cette classe contient les attributs et les méthodes relatifs à une agence. Dans une agence, on peut trouver plusieurs utilisateurs. Un utilisateur est affecté à une seule agence.
- Classe Rôle : Cette classe contient les attributs et les méthodes relatifs à un rôle. Un rôle peut être affecté à un ou plusieurs utilisateurs, tandis qu'un utilisateur peut avoir un ou plusieurs rôles.
- Classe Sous-Rôle : Cette classe contient les attributs et les méthodes relatifs à un sous-rôle. Un rôle est composé des plusieurs sous-rôle.

4.4 Réalisation

Dans cette partie nous allons présenter des captures d'écrans des parties fonctionnelles de notre application développée pendant le sprint 2.

4.4.1 Liste des rôles

La figure 4.8 montre l'interface de la liste des rôles. :

Nom	Description	ACTIONS
COLLECTOR_MANAGER	Description of collector manager	
ADMINISTRATOR	Description of administrator	

FIGURE 4.8 – Interface liste des rôles

Cette interface est destinée aux administrateurs et leur permet de consulter les différents rôles disponibles. Les administrateurs peuvent parcourir la liste des rôles et visualiser les détails de chaque rôle en cliquant sur celui qui les intéresse. Cela leur permet d'obtenir plus d'informations sur le rôle sélectionné, y compris les permissions spécifiques qui lui sont attribuées. À travers cette interface l'administrateur a l'accès de modifier ou supprimer un rôle.

4.4.2 Modifier un rôle

La figure 4.9 montre l'interface "modifier un rôle" :

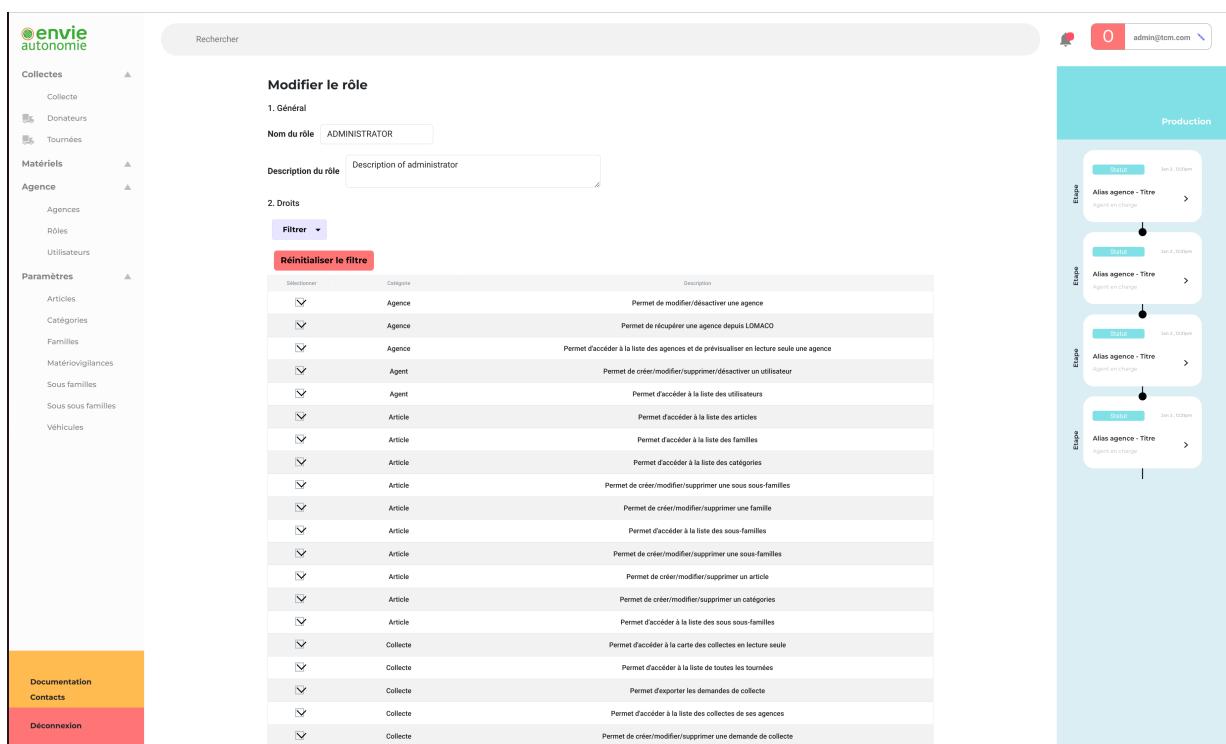


FIGURE 4.9 – Interface modifier rôle

L'interface "Modifier rôle" permet à l'administrateur de modifier un rôle de deux façons. Tout d'abord, il peut modifier le nom du rôle, en le remplaçant par un nouveau nom. L'administrateur peut également affecter ou révoquer des permissions spécifiques au rôle, lui permettant de définir les autorisations appropriées pour ce rôle. Ainsi, l'interface "Modifier rôle" offre à l'administrateur la possibilité d'ajuster et personnaliser les attributs et les permissions d'un rôle.

4.4.3 Consulter agence :

La figure 4.10 montre l'interface "consulter agence" :

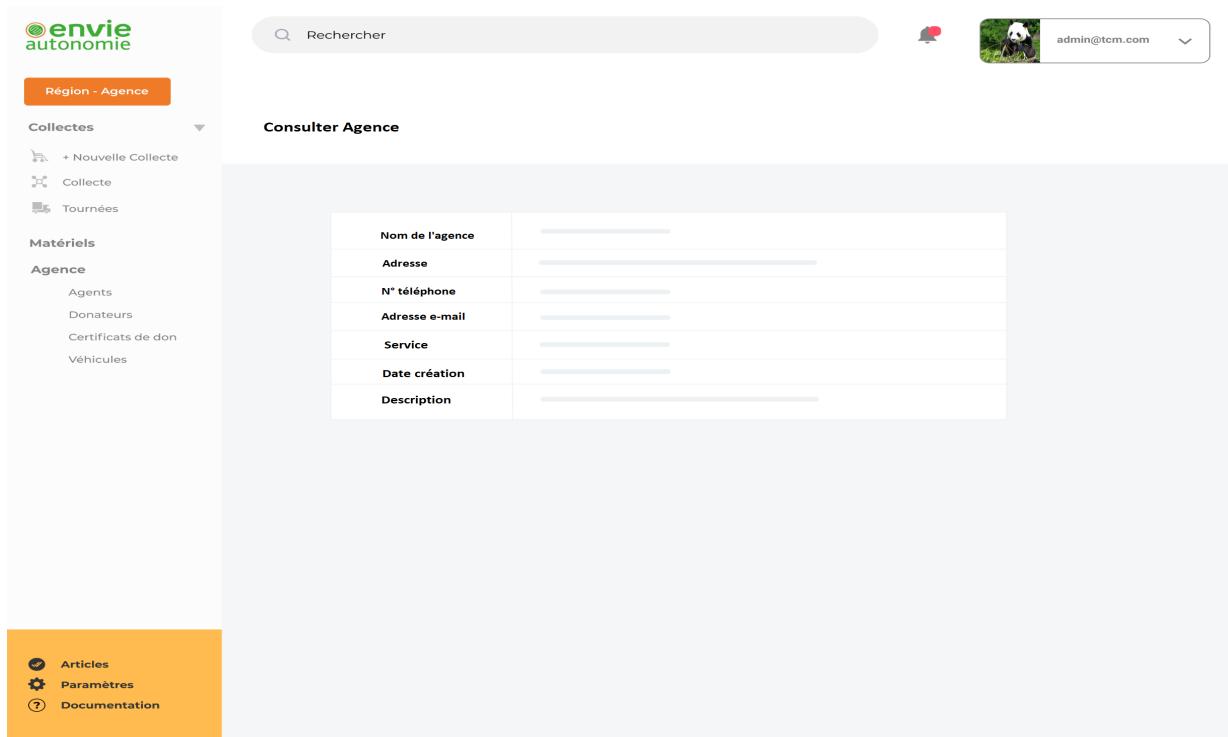


FIGURE 4.10 – Interface consulter agence

l'administrateur peut sélectionner une agence dans la liste des agences disponibles. Une fois l'agence est sélectionnée, l'interface affiche les informations associées à cette agence, telles que son nom, son adresse, son numéro de téléphone, etc. L'administrateur peut consulter ces informations pour obtenir une vue d'ensemble complète de l'agence.

Cette fonctionnalité de consultation permet à l'administrateur d'accéder rapidement et facilement aux détails pertinents d'une agence.

4.4.4 Tableau de bord :

La figure 4.11 présente le tableau de bord de notre application :

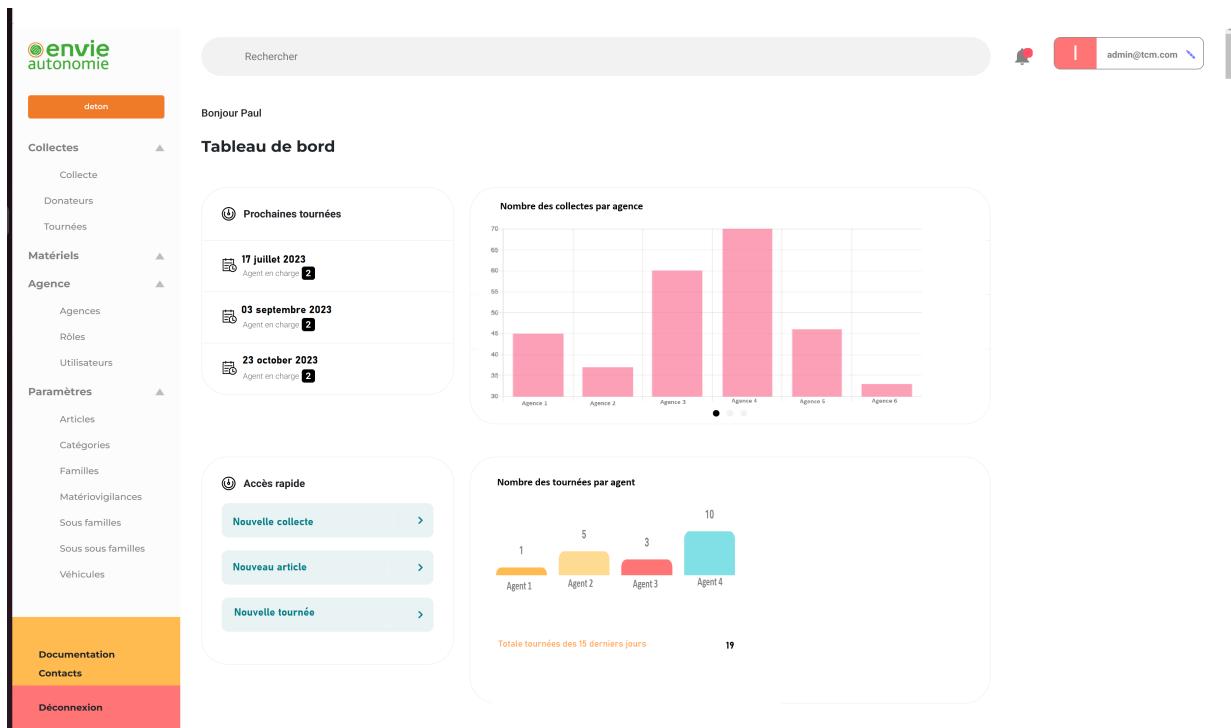


FIGURE 4.11 – Tableau de board

Cette interface nous permet d'avoir une vision globale de notre application. Elle affiche une section pour les prochaines tournées à venir, une section d'accès rapide et deux sections dédiées aux statistiques.

Statistiques :

Les figures 4.12 , 4.13 présentent les graphiques de statistiques de notre application.

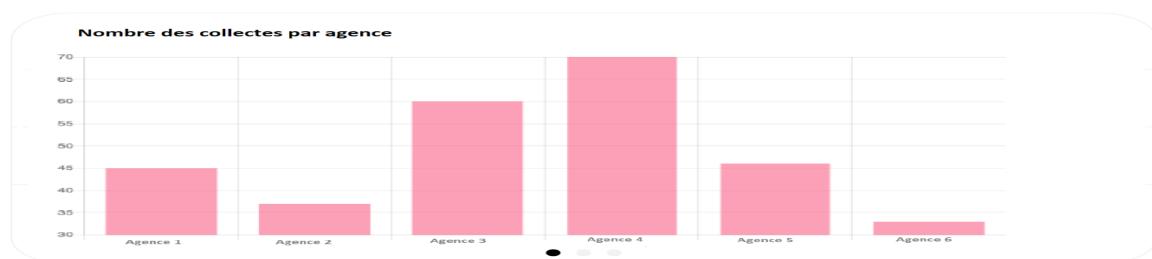


FIGURE 4.12 – statistiques des collectes par agence

Cette interface nous permet de surveiller le rendement de chaque agence en ayant une visibilité sur le nombre de demandes de collecte traitées par chaque agence. L'administrateur peut visualiser les statistiques des collectes par agence pour l'année , le mois ou la semaine en cours. Il suffit de cliquer sur les points de pagination situés en bas du graphique pour mettre à jour le graphique en conséquence.

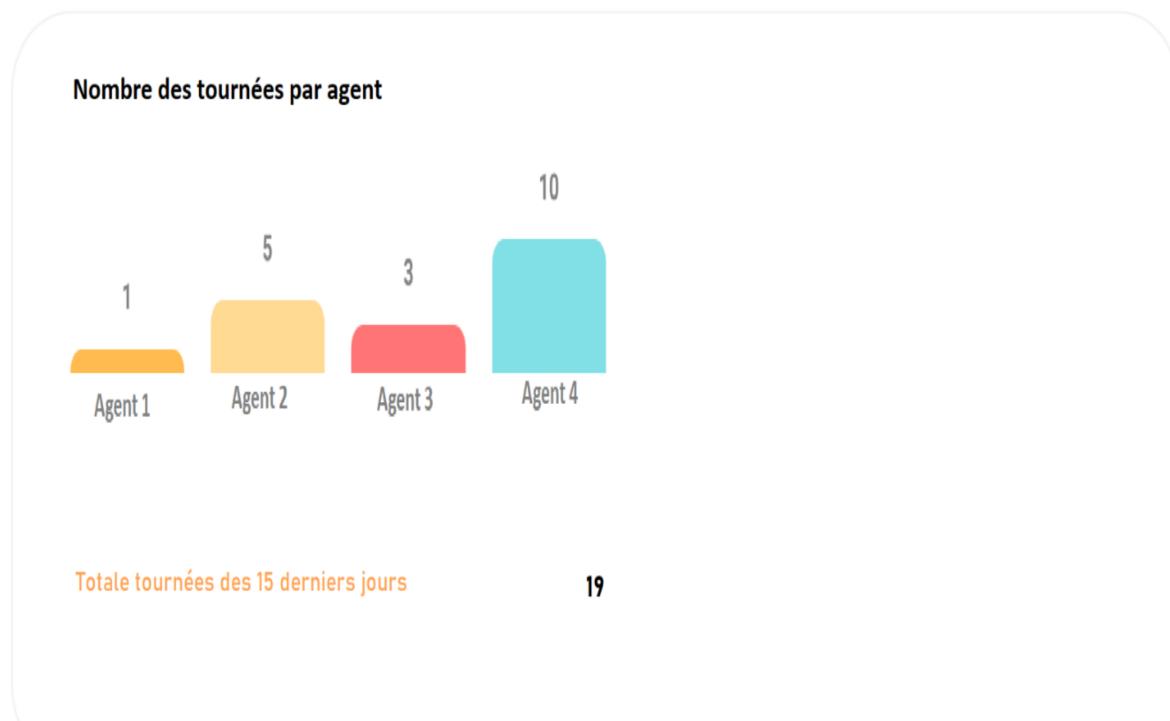


FIGURE 4.13 – statistiques des tournées par agent

Cette interface est conçue pour permettre la surveillance et le suivi du rendement de chaque agent. Elle offre une visibilité claire sur le nombre de tournées réalisées par chaque agent chaque deux semaines.

Cette fonctionnalité peut être utile pour identifier les agents les plus performants, ceux qui pourraient avoir besoin d'un soutien supplémentaire ou ceux qui pourraient bénéficier de formations spécifiques.

Accès rapide :

La figure 4.14 montre la partie d'accès rapide.



FIGURE 4.14 – Accès rapide

Cette section est conçue pour offrir à l'utilisateur une expérience plus fluide en lui permettant d'accéder rapidement à ses trois dernières actions effectuées sur l'application. Cette fonctionnalité permet à l'utilisateur de retrouver facilement les tâches ou les fonctionnalités auxquelles il a récemment travaillé, sans avoir à naviguer à travers plusieurs menus ou à effectuer des recherches.

Prochaine tournée :

La figure 4.15 présente les dates des prochaines tournées.

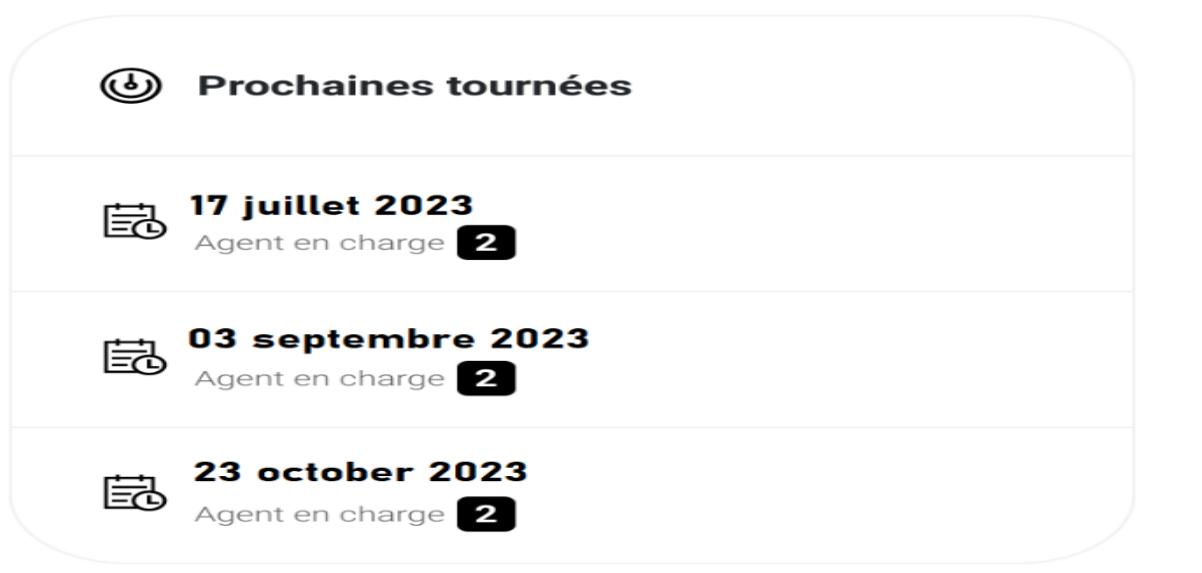


FIGURE 4.15 – Prochaine tournée

Cette section spécifique de l’interface est conçue pour offrir à l’utilisateur une visibilité immédiate sur les trois prochaines tournées prévues. Les informations sont présentées de manière organisée, avec la date la plus proche en tête de liste, suivie des dates ultérieures.

Cette fonctionnalité vise à aider l’utilisateur à planifier et à se préparer en conséquence. En ayant une vue d’ensemble des prochaines tournées, l’utilisateur peut anticiper les ressources nécessaires, coordonner les équipes et s’assurer que toutes les tâches sont effectuées en temps voulu.

4.4.5 Tests unitaires :

Pour assurer la qualité de notre travail, nous avons réalisé des tests unitaires.

Les tests unitaires sont des tests automatisés qui vérifient le bon fonctionnement des unités de code individuelles, telles que les classes, les fonctions ou les méthodes. Ils permettent de s’assurer que chaque composant fonctionne correctement de manière isolée, en vérifiant son comportement attendu et en détectant les éventuelles erreurs ou les cas limite.

La figure 4.16 montre le résultat de l'exécution des tests unitaires .

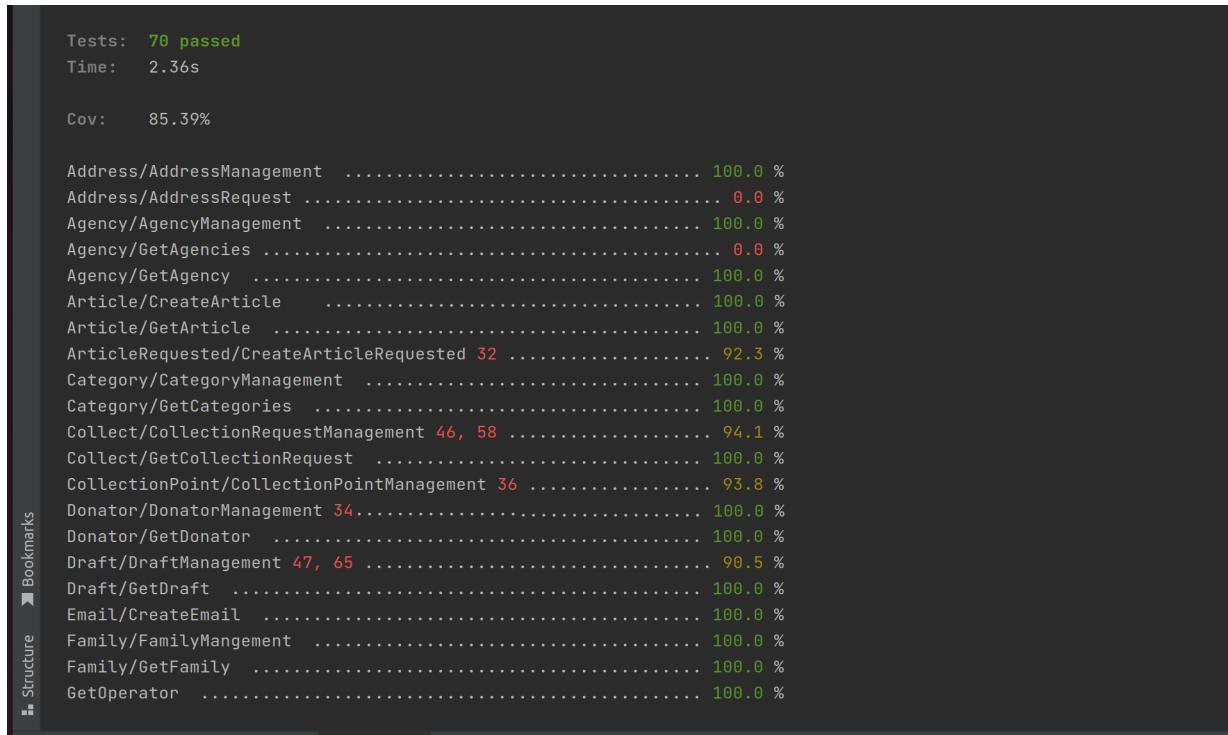


FIGURE 4.16 – Tests unitaires

Cette interface montre la proportion du code qui est exercée par les tests unitaires. Une couverture de test élevée implique que la plupart, voire la totalité, des lignes de code sont testées, ce qui contribue à une meilleure confiance dans la qualité du logiciel.

Conclusion

Dans ce chapitre nous avons traité les besoins spécifiques aux deuxième sprint , nous avons conçu les différents diagrammes et nous avons réalisé un deuxième incrément.

Chapitre 5

Sprint 3 : Conteneurisation

“ Tu peux tout accomplir dans la vie si tu as le courage de le rêver, l'intelligence d'en faire un projet réaliste, et la volonté de voir ce projet mené à bien.”

Sidney A. Friedman

Introduction

Dans le chapitre précédent, nous avons consacré notre temps à la conception et au développement de l'application. Maintenant, nous allons nous concentrer sur la partie de la conteneurisation en utilisant Docker, pour garantir le succès de notre projet,

5.1 Backlog du sprint 3

Dans cette section, nous allons explorer le backlog du produit du troisième sprint qui se présente comme suit :

TABLE 5.1 – Backlog du produit - sprint 3

ID	User Stories	Priorité
1	En tant qu'ingénieur devOps je veux installer docker	1
2	En tant qu'ingénieur devOps je veux installer docker compose	1
3	En tant qu'ingénieur devOps, je veux configurer le fichier docker file	1
4	En tant qu'ingénieur devOps, je veux créer les images docker	2
5	En tant qu'ingénieur devOps, je veux configurer le fichier docker-compose	2
6	En tant qu'ingénieur devOps, je veux lancer les services dans des conteneurs	3
7	En tant qu'ingénieur devOps, je veux gérer les conteneurs	3

5.2 Analyse et conception

Dans cette section, nous allons fournir une vue d'ensemble claire de notre système en présentant le diagramme d'activité, les composants de base et l'architecture physique.

5.2.1 Diagramme d'activité

La mise en place d'outils peut être une procédure complexe qui nécessite une planification et une exécution précise. Le diagramme d'activité illustré par la figure 5.1 permet de visualiser l'ensemble du processus de manière claire et concise, en identifiant les différentes étapes et les relations entre elles.

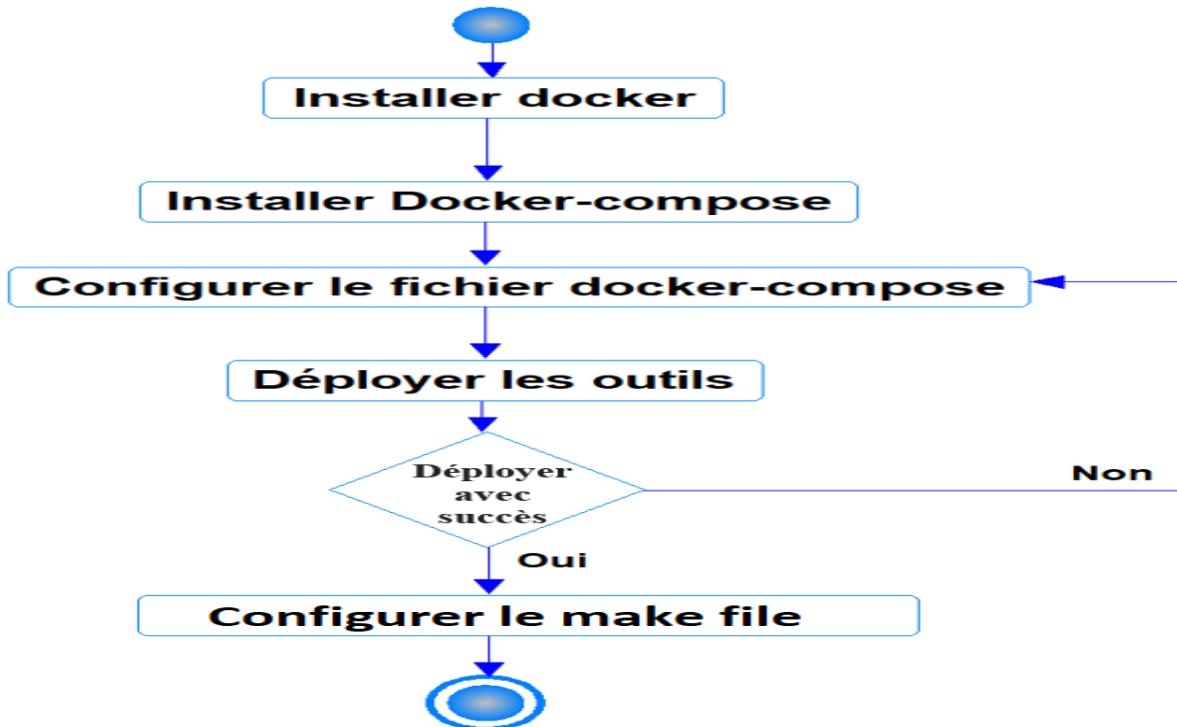


FIGURE 5.1 – Diagramme d'activité du sprint

5.2.2 Architecture physique

La figure 5.2 présente les étapes nécessaires pour déployer notre application dans des conteneurs, fournissant ainsi une architecture complète pour assurer le déploiement technique et fonctionnel de la solution.

Pour ce faire, nous utilisons un serveur hôte Ubuntu sur lequel nous installons Docker. Ensuite, nous procédons à l'installation de Docker Compose, afin de créer des conteneurs pour le backend, le frontend, la base de données et le serveur de mailing.

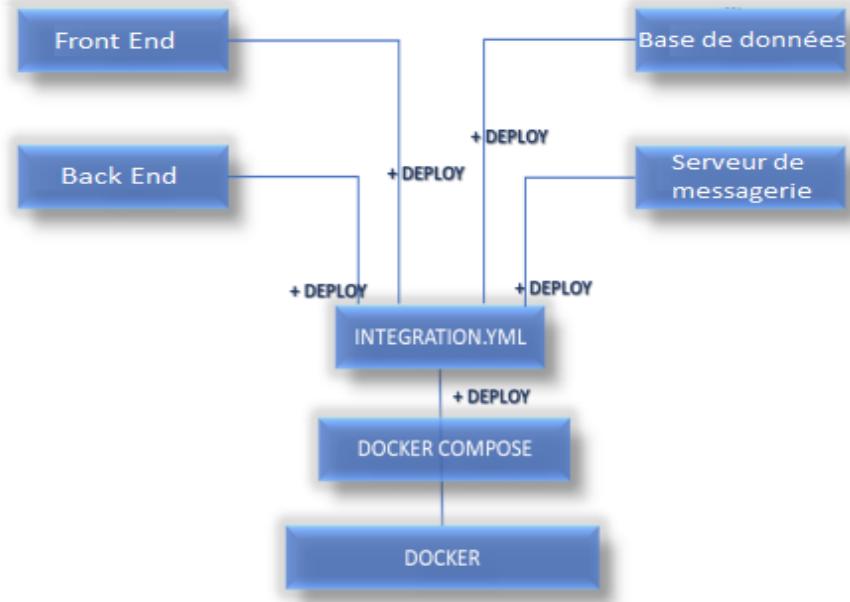


FIGURE 5.2 – Diagramme de Déploiement

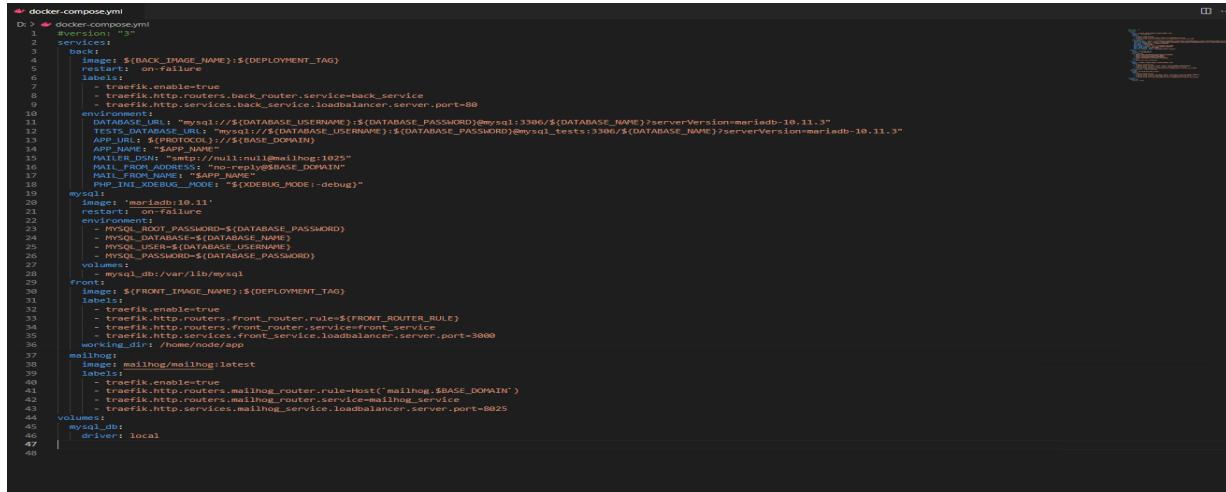
5.3 Réalisation

Cette section est dédiée à la présentation du travail accompli au cours de ce sprint à travers des captures d'écran.

5.3.1 Configuration service

Dans cette section, nous allons détailler les différentes étapes de construction de notre environnement Docker et de notre configuration Docker Compose.

Nous avons utilisé un fichier docker-compose.yml pour configurer Docker Compose, où nous avons défini les spécifications requises pour construire et déployer les conteneurs de manière coordonnée. Cette configuration est illustrée dans la figure 5.3, et permet de mettre en place un environnement cohérent pour l'exécution de notre application.



```

version: "3"
services:
  back:
    image: ${BACK_IMAGE_NAME}:${DEPLOYMENT_TAG}
    restart: on-failure
    labels:
      - traefik.enable=true
      - traefik.http.routers.back_router.service=back_service
      - traefik.http.services.back_service.loadbalancer.server.port=80
    environment:
      - DATABASE_URL="mysql://${DATABASE_USERNAME}:${DATABASE_PASSWORD}@mysql:3306/${DATABASE_NAME}?serverVersion=mariadb-10.11.3"
      - TESTS_DATABASE_URL="mysql://${DATABASE_USERNAME}:${DATABASE_PASSWORD}@mysql_tests:3306/${DATABASE_NAME}?serverVersion=mariadb-10.11.3"
      - APP_URL="http://$HOST_IP:$PORT"
      - MAIL_FROM_ADDRESS="no-reply@${BASE_DOMAIN}"
      - MAIL_NAME="SAPP NAME"
      - PHP_INI_XDEBUG_MODE="${XDEBUG_MODE}-debug"
    mysql:
      image: "mariadb:10.11"
      restart: always
      environment:
        - MYSQL_ROOT_PASSWORD=${DATABASE_PASSWORD}
        - MYSQL_DATABASE=${DATABASE_NAME}
        - MYSQL_USER=${DATABASE_USERNAME}
        - MYSQL_PASSWORD=${DATABASE_PASSWORD}
      volumes:
        - mysql_db:/var/lib/mysql
    front:
      image: ${FRONT_IMAGE_NAME}:${DEPLOYMENT_TAG}
      labels:
        - traefik.enable=true
        - traefik.http.routers.front_router.rule=${FRONT_ROUTER_RULE}
        - traefik.http.routers.front_router.service=front_service
        - traefik.http.services.front_service.loadbalancer.server.port=3000
      working_dir: /home/node/app
    mailhog:
      image: mailhog/mailhog:latest
      labels:
        - traefik.enable=true
        - traefik.http.routers.mailhog_router.rule=Host("${BASE_DOMAIN}")
        - traefik.http.routers.mailhog_router.service=mailhog_service
        - traefik.http.services.mailhog_service.loadbalancer.server.port=1025
  volumes:
    mysql_db:
      driver: local

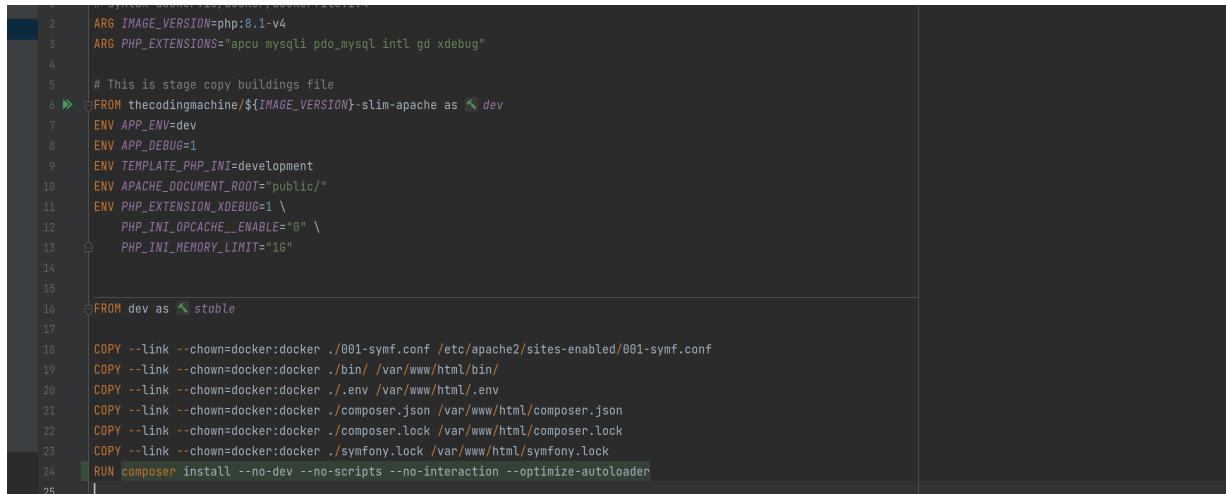
```

FIGURE 5.3 – Fichier Docker compose

Dans ce fichier, nous avons inclus toutes les spécifications nécessaires pour déployer les services indispensables à notre application. Cela comprend notamment le backend, le frontend, la base de données et le service de messagerie

Service Backend

Nous avons commencé par spécifier dans notre Dockerfile l'image Docker que nous avions créée pour le service backend. La figure 5.3 représente le contenu de ce Dockerfile spécifique au backend.



```

ARG IMAGE_VERSION="php:8.1-v4"
ARG PHP_EXTENSIONS="apcu mysqli pdo_mysql intl gd xdebug"

# This is stage copy buildings file
FROM thecodingmachine/${IMAGE_VERSION}-slim-apache as dev
ENV APP_ENV=dev
ENV APP_DEBUG=1
ENV TEMPLATE_PHP_INIT=development
ENV APACHE_DOCUMENT_ROOT="public/"
ENV PHP_EXTENSION_XDEBUG=1 \
    PHP_INI_OPCACHE__ENABLE="0" \
    PHP_INI_MEMORY_LIMIT="1G"

FROM dev as stable

COPY --link --chown=docker:docker ./001-symf.conf /etc/apache2/sites-enabled/001-symf.conf
COPY --link --chown=docker:docker ./bin/ /var/www/html/bin/
COPY --link --chown=docker:docker ./env /var/www/html/.env
COPY --link --chown=docker:docker ./composer.json /var/www/html/composer.json
COPY --link --chown=docker:docker ./composer.lock /var/www/html/composer.lock
COPY --link --chown=docker:docker ./symfony.lock /var/www/html/symfony.lock
RUN composer install --no-dev --no-scripts --no-interaction --optimize-autoloader

```

FIGURE 5.4 – docker file backend

Dans notre Dockerfile, nous avons spécifié la configuration suivante :

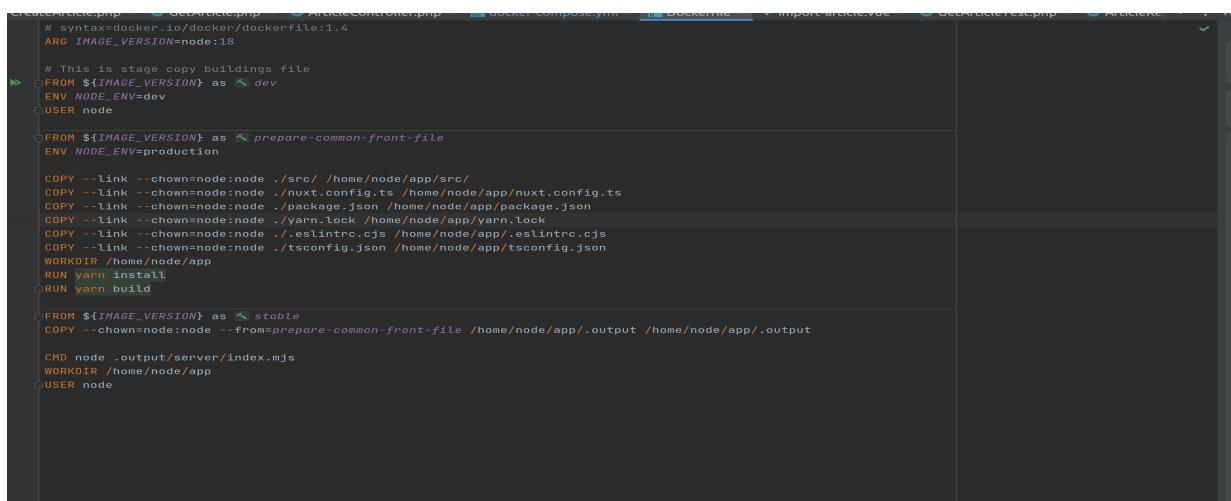
- Définir l'image de base pour la phase de développement.
- déclarer les variables d'environnement, telles que l'environnement de développement, le mode débogage, le modèle de configuration PHP..
- fixer une phase stable basée sur la phase de développement (dev), qui sera utilisée pour la production.
- Copier certains fichiers dans le conteneur en utilisant des options pour définir les propriétés de lien et de propriétaire.
- Exécuter la commande "composer install" pour installer les dépendances de notre application.

Ces étapes contribuent à la construction de l'image Docker pour notre service backend.

Dans notre service backend, nous avons également configuré le redémarrage du service en cas d'échec. De plus, nous avons défini des variables d'environnement pour configurer l'application backend, telles que les informations de connexion à la base de données, l'URL de l'application, le nom de l'application, les informations de configuration pour le service de messagerie, les informations de configuration pour le stockage et le mode Xdebug.

Service Frontend

Pour le service frontend, nous avons commencé par spécifier l'image Docker que nous avons créée en utilisant notre Dockerfile. Voici le contenu de ce Dockerfile spécifique au frontend :



```

# syntax=docker.io/docker/dockerfile:1.4
ARG IMAGE_VERSION=node:18

# This is stage copy buildings file
FROM ${IMAGE_VERSION} as dev
ENV NODE_ENV=dev
USER node

FROM ${IMAGE_VERSION} as prepare-common-front-file
ENV NODE_ENV=production

COPY --link --chown=node:node ./src/ /home/node/app/src/
COPY --link --chown=node:node ./nuxt.config.ts /home/node/app/nuxt.config.ts
COPY --link --chown=node:node ./package.json /home/node/app/package.json
COPY --link --chown=node:node ./yarn.lock /home/node/app/yarn.lock
COPY --link --chown=node:node ./eslintrc.cjs /home/node/app/eslintrc.cjs
COPY --link --chown=node:node ./tsconfig.json /home/node/app/tsconfig.json
WORKDIR /home/node/app
RUN yarn install
RUN yarn build

FROM ${IMAGE_VERSION} as stable
COPY --chown=node:node --from=prepare-common-front-file /home/node/app/.output /home/node/app/.output

CMD node .output/server/index.mjs
WORKDIR /home/node/app
USER node

```

FIGURE 5.5 – docker file frontend

Dans ce Dockerfile de la partie frontend, nous avons spécifié la configuration suivante :

- Définir l'image de base pour la phase de développement.
- Déclarer une variable d'environnement pour l'environnement de développement.

- Copier certains fichiers dans le conteneur en utilisant des options pour définir les propriétés de lien et de propriétaire.
- Accéder au répertoire de l'application frontend et exécuter les commandes "yarn install" et "yarn build" pour installer les dépendances et générer les fichiers de production.

Ces étapes dans notre Dockerfile contribuent à la construction de l'image Docker pour notre service frontend. Nous avons également spécifié les informations suivantes :

- Définir une image Docker pour utiliser ce service.
- Ajouter des labels pour configurer Traefik qui agit en tant que reverse proxy.
- Mettre en place d'un service frontend qui utilise le port 3000 pour être accessible .

Service Base de données

Pour le service base de données , nous avons commencé par spécifier l'image Docker maria db importé du Docker Hub, nous avons également spécifié les informations suivantes :

- Configurer les variables d'environnement suivants : L'utilisateur, nom de la base et les mots de passe.
- Définir le redémarrage de service en cas de problème.
- Créer un volume pour le partage du backup des données entre les conteneurs Docker et l'hôte.

Service de Mailing

Nous avons configuré le service *mailhog* de la manière suivante :

- Nous avons utilisé l'image Docker `mailhog/mailhog:latest` provenant de Docker Hub.
- Pour intégrer Traefik, un reverse proxy, nous avons ajouté les labels suivants :
 - `traefik.enable=true` pour activer la gestion du service par Traefik.
 - `traefik.http.routers.mailhog_router.rule=Host('mailhog.$BASE_DOMAIN')` pour définir la règle de routage dans Traefik. Cela signifie que le trafic entrant avec l'hôte `mailhog.$BASE_DOMAIN` sera dirigé vers ce service.
 - `traefik.http.routers.mailhog_router.service=mailhog_service` pour indiquer à Traefik d'utiliser le service `mailhog_service`.
 - `traefik.http.services.mailhog_service.loadbalancer.server.port=8025` pour spécifier le port du serveur à utiliser pour le service `mailhog_service`. Traefik redirigera le trafic vers le port 8025 de ce service.

Grâce à cette configuration, lorsque le trafic est envoyé à l'hôte `mailhog.$BASE_DOMAIN`, Traefik le redirigera vers le service MailHog qui utilise l'image spécifiée. Cela permettra

d'utiliser MailHog pour capturer et visualiser les e-mails dans notre environnement Docker.

5.3.2 Make file

Nous avons crée un fichier "Makefile" pour automatiser des tâches liées à Docker. Il est souvent appelé un "Makefile Docker" ou un "Makefile pour Docker". Ce type de Makefile permet d'automatiser des actions courantes dans le contexte de Docker, telles que la construction d'images, le lancement de conteneurs et la gestion des volumes.

```

M Makefile ×
D: > M Makefile
1/
18 # connect to the back container
19 .PHONY: bbash
20 bbash: ;\
21     docker compose exec back bash;
22
23 # connect to the front container
24 .PHONY: fbash
25 fbash: ;\
26     docker compose exec front bash;
27
28 # Launch migration
29 .PHONY: migrate
30 migrate: ;\
31     docker compose exec back composer -- run console doctrine:migrations:migrate -n
32
33 # Force doctrine update from entities
34 .PHONY: db-dev-update
35 db-dev-update: ;\
36     docker compose exec back composer -- run console doctrine\:\schema\:\update -f
37
38 db-dev-fixtures: ;\
39     docker compose exec back composer -- run console doctrine:f:load -n
40 # Launch generate migration (recommended) from database diff
41 # doctrine:schema:validate --skip-sync is used to check the database mapping before generating the migration,
42 # skip-sync is used to skip database and mapping sync validation
43 db-mig-diff: ;\
44     docker compose exec back composer -- run console doctrine:schema:validate --skip-sync && \
45     docker compose exec back composer -- run console make:migration
46
47 # Fully reset the database deleting all tables and re-adding the fixtures
48 .PHONY: reset-database
49 reset-database: ;\
50     docker compose exec back composer -- run console doctrine:database:drop -n --force && \
51     docker compose exec back composer -- run console doctrine:database:create -n && \
52     docker compose exec back composer -- run console doctrine:schema:update -f --complete && \
53     docker compose exec back composer -- run console doctrine:f:load -n
54

```

FIGURE 5.6 – Make file

Le Makefile que nous avons implémenté utilise différentes commandes parmi lesquelles on a :

1. bbash : Cette cible nous permet de nous connecter au conteneur back en utilisant la commande docker-compose exec back bash. Cela ouvre un shell bash interactif dans le conteneur back.
2. fbash : Cette cible nous permet de nous connecter au conteneur front en utilisant la commande docker-compose exec front bash. Cela ouvre un shell bash interactif dans le conteneur front.

3. **migrate** : Cette cible lance les migrations de base de données en exécutant la commande docker-compose exec back composer – run console doctrine :migrations :migrate -n. Cela exécute les migrations Doctrine pour mettre à jour le schéma de la base de données.
4. **db-dev-update** : Cette cible force la mise à jour de la base de données en exécutant la commande docker-compose exec back composer – run console doctrine schema update -f. Cela met à jour le schéma de la base de données en fonction des entités existantes.
5. **db-dev-fixtures** : Cette cible charge les fixtures de développement dans la base de données en exécutant la commande docker-compose exec back composer – run console doctrine :f :load -n.
6. **db-mig-diff** : Cette cible génère une migration à partir de la différence entre le schéma de la base de données actuel et les entités Doctrine. Cela est réalisé en exécutant les commandes doctrine :schema :validate –skip-sync pour valider le mapping de la base de données, puis make :migration pour générer la migration.
7. **reset-database** : Cette cible réinitialise complètement la base de données. Elle supprime toutes les tables, recrée le schéma de la base de données, puis charge les fixtures à l'aide des commandes appropriées.
8. **blogs** : Cette cible affiche les journaux (logs) du conteneur *back* en utilisant la commande docker-compose logs back -f. Les journaux sont affichés en temps réel.
9. **flogs** : Cette cible affiche les journaux (logs) du conteneur *front* en utilisant la commande docker-compose logs front -f.

Il y a également d'autres cibles définies dans le Makefile pour gérer le lancement et l'arrêt des conteneurs Docker, exécuter des tâches de maintenance de la base de données.

5.3.3 Visualisation

Dans cette section, nous souhaitons illustrer nos réalisations dans la partie Docker en incluant quelques captures d'écran.

CONTAINER ID	IMAGE	STATUS	NAMES
48ffb2323dda	git.thecodingmachine.com:444/tom-projects/envie-autonomie/back:latest	Up 18 hours	envie-autonomie-back-1
86022628bcfc	phpmyadmin	Up 18 hours	envie-autonomie-phpmyadmin-1
fdf286ac97cd	minio/minio:latest	Up 18 hours	envie-autonomie-minio-1
19d0ae9737c4	mailhog/mailhog:latest	Up 18 hours	envie-autonomie-mailhog-1
52ada64a7b81	mariadb:10.11	Up 18 hours	envie-autonomie-mysql_tests-1
424e304b3420	traefik:3.0	Up 18 hours	tcm_proxy
df8c1dfa2175	git.thecodingmachine.com:444/tom-projects/envie-autonomie/front:latest	Up 18 hours	envie-autonomie-front-1
f6e2d4b7297d	mariadb:10.11	Up 18 hours	envie-autonomie-mysql-1

FIGURE 5.7 – La plateforme conteneurisée

La figure 5.7 représente le résultat détaillé de l'exécution de la commande "docker ps". Elle offre une vue complète de tous les conteneurs qui composent notre application.

Les figures 5.8, 5.9 montrent distinctement les différents composants de déploiement de notre application, tels que le frontend et la base de données SQL .

5.3. Réalisation

67

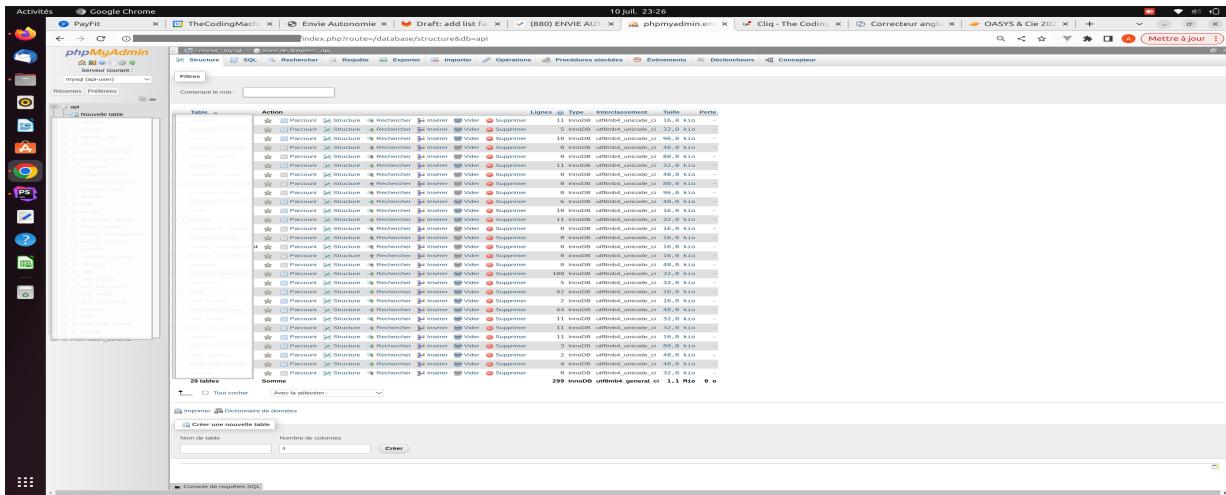


FIGURE 5.8 – Base de données conteneurisée

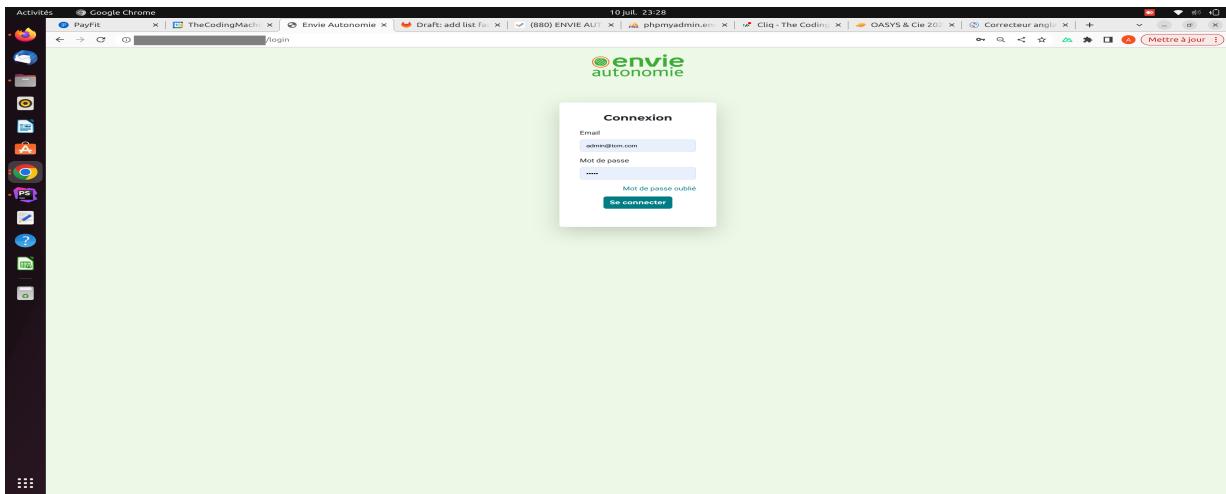


FIGURE 5.9 – La plateforme conteneurisée

Cette séparation efficace des différents composants est un élément clé de notre architecture basée sur Docker. Chaque conteneur exécute une partie spécifique, ce qui permet une gestion et une évolutivité plus aisées. Par exemple, nous pouvons faire évoluer indépendamment les ressources allouées à chaque composant en fonction de ses besoins spécifiques, tout en préservant l'isolation entre eux.

Cette approche basée sur des conteneurs nous offre une plus grande flexibilité dans le déploiement et la gestion de notre application. En cas de besoin, nous pouvons facilement ajouter de nouveaux conteneurs pour prendre en charge de nouvelles fonctionnalités ou pour gérer la charge de manière équilibrée. De plus, en utilisant Docker, nous bénéficions de la portabilité des conteneurs, ce qui facilite le déploiement de notre application sur

différentes plateformes sans avoir à se soucier des dépendances logicielles spécifiques à chaque environnement.

En résumé, l'exécution de la commande "docker ps" met en évidence la réussite de notre approche basée sur les conteneurs.

Conclusion

Au cours de ce chapitre, nous avons présenté le balckog de sprint ainsi que les différentes étapes de la partie conteneurisation et virtualisation de notre application.

Conclusion générale

Ce projet de fin d'étude nous a permis de réaliser d'importants progrès dans le développement d'une plateforme novatrice de gestion de collecte et de remise en bon état du matériel médical. Nous avons débuté en présentant le contexte du projet et en effectuant une étude approfondie de l'existant, ce qui nous a permis de mieux comprendre les besoins et les défis spécifiques du secteur de la santé.

Ensuite, nous avons procédé à une analyse détaillée des besoins, en identifiant les fonctionnalités clés de la plateforme et en établissant le backlog du produit. Nous avons pris soin de choisir les technologies les plus adaptées pour assurer la performance et la flexibilité de la plateforme, tout en tenant compte de notre environnement de développement.

Puis, nous avons entamé la mise en œuvre de la plateforme en réalisant le contenu du premier et deuxième sprint. Nous avons accordé une attention particulière au développement des fonctionnalités, telles que la gestion des articles , tournées collectes ,rôles et agences,et nous avons veillé à réaliser des tests approfondis pour garantir la qualité et la fiabilité du système.

Par la suite nous avons travaillé sur le Troisième sprint.nous avons Implémenté la dockerisation au niveau de notre projet afin de rendre le travail plus facile .

En conclusion, ce rapport témoigne de notre engagement à contribuer activement à la transformation numérique des besoins clients dans le domaine de la santé, ainsi que des besoins de la société en matière de dockerisation. Ce stage nous a permis d'acquérir une précieuse valeur professionnelle en renforçant nos compétences techniques, notre capacité d'adaptation et notre compréhension des enjeux de la transformation digitale. En termes de perspectives, nous pouvons envisager de travailler sur l'intégration et le déploiement contenu et réalisation d'une application mobile.

Webographie

- [1] “The coding machine.” <https://thecodingmachine.com/contact/agence-de-developpement-paris/>. [Consulté le 3 Mars 2023].
- [2] “Architecture mvc.” <https://developer.mozilla.org/fr/docs/Glossary/MVC>. [Consulté le 15 Mars 2023].
- [3] “Phpstorm.” <https://www.jetbrains.com/fr-fr/phpstorm/>. [Consulté le 20 Mars 2023].
- [4] “Git.” <https://intellij-idea.softonic.com/>. [Consulté le 20 Mars 2023].
- [5] “Symfony.” <https://www.adimeo.com/blog/quest-ce-que-symfony-et-quels-sont-ses-avantages> [Consulté le 1 Avril 2023].
- [6] “Vue.js.” <https://v2.fr.vuejs.org/v2/guide>. [Consulté le 2 Avril 2023].