



Ministère de l'Enseignement Supérieur, de la
Recherche Scientifique
Direction Générale des Études Technologiques



Institut Supérieur des Études Technologiques de Djerba
Département Technologies de l'informatique

Code projet	
----------------	--

Rapport de
STAGE DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de

**Licence en Technologies de
L'Informatique**

Parcours : MDW

**Développement d'une application web et
mobile de prise des commandes pour les
restaurants.**

Elaboré par:

Amen Allah Zemzem

Adem Nasr

Encadré par :

M. Tahar Haouet

Effectué à :

Entreprise : ALL TECH COMPANY

Encadreur : M. Tahar Haouet

Année universitaire : 2023/2024

Dédicaces

« Sous le ciel de l'effort, au jardin de l'espoir, »

Avec tout mon amour et ma profonde gratitude

je dédie ce travail à ma famille. À mes parents, pour leur amour inconditionnel, leur soutien sans faille et les sacrifices qu'ils ont consentis pour m'accompagner sur le chemin de la réussite. Votre présence et votre encouragement ont été mon roc et ma source d'inspiration chaque jour. À mes frères, pour leur complicité, leur soutien et leur humour, qui ont allégé les moments de stress et de doute.

Je dédie également ce mémoire à mes amis, véritables compagnons de route, qui ont partagé avec moi les joies et les peines de cette aventure académique. Merci pour tous les moments de détente, pour vos conseils, votre écoute et vos encouragements quand j'en avais le plus besoin.

Cette dédicace s'étend à toutes les personnes qui, de près ou de loin, ont contribué à mon parcours et ont enrichi ma vie d'expériences et de connaissances. C'est avec une immense reconnaissance que je pense à chacun de vous en partageant ce travail, le fruit de plusieurs années d'efforts et d'apprentissage.

Merci d'avoir été ma source d'inspiration et ma force.



Adam Nasri



Dédicaces

« La gratitude transforme ce que nous avons en suffisant, et bien plus encore »

Je dédie cet ouvrage

À ma mère qui m'a soutenu et encouragé durant ces années d'études. Qu'elle trouve ici le témoignage de ma profonde reconnaissance et de tout l'amour que j'ai pour elle. Sans son soutien indéfectible, ce chemin aurait été bien plus ardu.

À mon père qui m'a toujours inspiré et encouragé. Que cet ouvrage soit le reflet de ma gratitude envers lui. Ses conseils avisés et sa présence réconfortante ont été des piliers essentiels tout au long de mon parcours.

À mon frère et mes sœurs, qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Leur soutien chaleureux et leurs encouragements constants m'ont permis de garder le cap et de persévérer malgré les difficultés.

À ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

À tous mes amis qui m'ont toujours encouragé, et à qui je souhaite beaucoup de succès.

Enfin, à tous ceux qui, de près ou de loin, ont contribué à cette aventure, recevez ici
l'expression de ma profonde reconnaissance

Amen Allah Remzem

Remerciements

Au terme de ce projet enrichissant, réalisé en collaboration avec ALL TECH Company, je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à sa réussite. Je remercie tout d'abord M. Tahar Haouet, mon encadrant professionnel, pour sa confiance, son soutien constant et les connaissances précieuses qu'il m'a transmises tout au long de cette aventure.

Je suis également reconnaissant envers tous les membres de l'équipe d'ALL TECH Company pour leur accueil, leur disponibilité et leur collaboration efficace, qui ont grandement facilité la réalisation de ce projet. Un remerciement particulier à mes encadrants académiques de l'Institut Supérieur des Études Technologiques de Djerba, qui ont guidé mes réflexions et enrichi mon parcours éducatif.

Enfin, je souhaite exprimer ma gratitude envers les membres du jury pour leur évaluation et leurs conseils constructifs, ainsi qu'à ma famille et mes amis pour leur soutien indéfectible et leur encouragement continu. Ce projet n'aurait pas pu atteindre ses objectifs sans l'engagement et le dévouement de chacun, et je suis fier de partager les fruits de ce travail commun.

Sommaire

Introduction générale	1
Chapitre 1: Présentation générale de projet	
Introduction	3
1. Présentation de l'organisme d'accueil.....	3
2. Cadre général du projet	3
3. Présentation de l'application.....	3
4. Choix de la méthodologie de travail	4
4.1. Comparaison entre les méthodes agiles.....	4
4.2. Scrum.....	5
5. Langage de modélisation	7
6. Choix Technologiques.....	7
6.1. Outils logiciels	7
6.2. Langages.....	8
6.3. Frameworks.....	9
6.4. Technologies	10
Conclusion	10
Chapitre 2: Etude préalable	
Introduction	12
1. Etude de l'existant.....	12
1.1. Gestion de l'administration.....	12
1.2. Gestion d'agent.....	12
1.3. Gestion de client.....	13
2. Objectifs de l'application Web/ Mobile.....	13
3. Spécification des besoins.....	14
3.1. Besoins fonctionnels	14
3.2. Besoins non fonctionnels.....	14
4. Architecture Logicielle utilisée :MVC	15
5. Architecture système déploiement utilisée : 3-tiers	16
6. Product Backlog.....	17
6.1. Identification des acteurs	17
6.2. Eléments du product backlog	17
7. Diagramme de cas d'utilisation global	22
8. Diagramme de classe analyse	23
2. Planification des sprints.....	23
Conclusion	24

Sommaire

Chapitre 3: Sprint 1 - gestion des comptes.....	
Introduction.....	26
1. Backlog du sprint 1	26
1.1. Affectation des cas d'utilisations pour les acteurs.....	26
1.2. Sprint backlog	26
2. Spécification fonctionnelle.....	27
2.1 Diagramme de cas d'utilisation global de sprint 1.....	27
2.2 Description textuelle des cas d'utilisation	28
Conclusion	36
Chapitre 4: Sprint 2 - gestion des utilisateurs.....	
Introduction.....	38
1. Backlog du sprint 2.....	38
1.1. Affectation des cas d'utilisations aux acteurs.....	38
1.2. Sprint backlog	38
2. Spécification fonctionnelle.....	39
2.1 Diagramme de cas d'utilisation global de sprint 2.....	39
2.2 Description textuelle des cas d'utilisation	39
2.3 Description comportementale des histoires du sprint 2.....	42
2.4 Diagramme de classe de sprint 2	43
3.1. Test sprint 2.....	44
3.2. Interfaçages	44
Conclusion	46
Chapitre 5: Sprint 3 - gestion des familles articles et articles.....	
Introduction.....	48
1. Backlog du sprint 3	48
1.1. Affectation des cas d'utilisations aux acteurs.....	48
1.2. Sprint backlog	48
2. Spécification fonctionnelle.....	49
2.1 Diagramme de cas d'utilisation global de sprint 3.....	49
2.2 Description textuelle des cas d'utilisation	50
2.3 Description comportementale des histoires du sprint 3.....	51
2.4 Diagramme de classe de sprint 3	55
3. Test et interfaçage	55
3.1. Test sprint 3.....	55
3.2. Interfaçage	55
Conclusion	57
Chapitre 6: Sprint 4 - gestion des terrasses et tables.....	
Introduction.....	59
1. Backlog du sprint 4	59
1.1. Affectation des cas d'utilisations aux acteurs.....	59

Sommaire

1.2. Sprint backlog	59
2. Spécification fonctionnelle	60
2.1 Diagramme de cas d'utilisation global de sprint 4	60
2.2 Description textuelle des cas d'utilisation	60
2.3 Description comportementale des histoires du sprint 4	62
2.4 Diagramme de classe de sprint 4	65
3. Test et interfaçage	66
3.1. Test sprint 4	66
3.2. Interfaçage	66
Conclusion	67
Chapitre 7: Sprint 5 - gestion des commandes et demandes	
Introduction	69
1. Backlog du sprint 5	69
1.1. Affectation des cas d'utilisations aux acteurs	69
1.2. Sprint backlog	69
2. Spécification fonctionnelle	70
2.5 Diagramme de cas d'utilisation global de sprint 5	70
2.6 Description textuelle des cas d'utilisation	71
2.7 Description comportementale des histoires du sprint 4	72
2.8 Diagramme de classe de sprint 5	74
3. Test et interfaçage	75
3.1. Test sprint 5	75
3.2. Interfaçage	75
Conclusion	78
Chapitre 8: Sprint 6 - gestion de tableau de bord	
Introduction	80
1. Backlog du sprint 6	80
1.1. Affectation des cas d'utilisations aux acteurs	80
1.2. Sprint backlog	80
2. Spécification fonctionnelle	81
2.1 Diagramme de cas d'utilisation global de sprint 6	81
2.2 Description textuelle des cas d'utilisation	81
4. Test et interfaçage	82
4.1. Test sprint 5	82
4.2. Interfaçage	82
Conclusion	83
Conclusion générale	84
Webographie	85

Liste des figures

Figure 1 : Logo de l'organisme d'accueil.....	3
Figure 2: La méthode SCRUM	6
Figure 3:Equipe SCRUM	6
Figure 4: Procédure de gestion de l'administration	12
Figure 5: Procédure de gestion d'agent.....	13
Figure 6: Procédure de gestion de client	13
Figure 7: Architecture MVC	16
Figure 8:Architecture 3_tiers.....	17
Figure 9: Diagramme de cas d'utilisation global	22
Figure 10: Diagramme de classe analyse	23
Figure 11: Diagramme de cas utilisation global du sprint 1.....	28
Figure 12: Diagramme de séquence de cas d'utilisation « S'inscrire ».....	31
Figure 13: Diagramme de séquence de cas d'utilisation "S'authentifier"	32
Figure 14:Diagramme de séquence de cas d'utilisation "Mettre à jour profile"	32
Figure 15: Diagramme de séquence de cas d'utilisation "Récupérer mot de passe oublié".....	33
Figure 16:Diagramme de classe de sprint 1	33
Figure 17: Interface login	34
Figure 18: Interface login mobile	34
Figure 19: Interface registre	35
Figure 20: Interface registre mobile	35
Figure 21: Interface de récupération de mot de passe	36
Figure 22 : Interface de mettre à jour de profil de client.....	36
Figure 23: Diagramme de cas d'utilisation global de sprint 2.....	39
Figure 24: Diagramme de séquence de cas d'utilisation "Ajouter agent"	42
Figure 25: Diagramme de séquence de cas d'utilisation « Modifier utilisateur ».....	42
Figure 26: Diagramme de séquence de cas d'utilisation "Supprimer utilisateur"	43
Figure 27: Diagramme de classe de sprint 2	43
Figure 28: Interface de bord de l'admin.....	44
Figure 29: Interface de gestion des agents	45
Figure 30: Interface gestion des clients	45
Figure 31: Diagramme de cas d'utilisation global de sprint 3 de l'administrateur	49
Figure 32: Diagramme de cas d'utilisation global de sprint 3 de client.....	50
Figure 33: Diagramme de séquence de cas d'utilisation "Ajouter famille articles"	52
Figure 34: Diagramme de séquence de cas d'utilisation "Modifier famille articles"	52

Liste des figures

Figure 35: Diagramme de séquence de cas d'utilisation "Supprimer famille articles"	53
Figure 36: Diagramme de séquence de cas d'utilisation "Ajouter articles"	53
Figure 37: Diagramme de séquence de cas d'utilisation "Modifier articles"	54
Figure 38: Diagramme de séquence de cas d'utilisation "Supprimer articles"	54
Figure 39: Diagramme de classe sprint 3	55
Figure 40: Interface de gérer des familles articles.....	55
Figure 41: Interface de gérer les articles	56
Figure 42: Interface de modification d'un article	56
Figure 43: Interface de liste des commandes de client.....	57
Figure 44:Diagramme de cas d'utilisation global de sprint 4 de l'administrateur	60
Figure 45:Diagramme de séquence de cas d'utilisation "Ajouter terrasse"	62
Figure 46:Diagramme de séquence de cas d'utilisation "Modifier terrasses"	63
Figure 47:Diagramme de séquence de cas d'utilisation "Supprimer terrasse"	63
Figure 48: Diagramme de séquence de cas d'utilisation "Ajouter tables"	64
Figure 49:Diagramme de séquence de cas d'utilisation "Modifier tables"	64
Figure 50:Diagramme de séquence de cas d'utilisation "Supprimer tables"	65
Figure 51: Diagramme de classe de sprint 4	65
Figure 52: Interface gestion des tables	66
Figure 53: Interface de gestion des terrasses	66
Figure 54: Diagramme de cas d'utilisation global de sprint 5 de l'agent	70
Figure 55: Diagramme de cas d'utilisation global de sprint 5 de client.....	70
Figure 56: Diagramme de séquence de cas d'utilisation "Passer commande"	73
Figure 57:Diagramme de séquence de cas d'utilisation "Passer demande"	74
Figure 58: Diagramme de classe sprint 5	74
Figure 59: Interface de liste des commandes de l'agent	75
Figure 60: Interface de liste des commandes de client.....	76
Figure 61: Interface home de client.....	76
Figure 62 : Interface home de client avec side-menu.....	77
Figure 63: Interface de liste des demandes de l'agent	77
Figure 64 : Cas d'utilisation de sprint 6.....	81
Figure 65 : Dashboard administrateur	82

Liste des tableaux

Tableau 1:Comparaison de méthodologie de conception	4
Tableau 2 : Le backlog du produit	21
Tableau 3 : Planning des sprints.....	24
Tableau 4 :Classification des cas d'utilisations par acteur	26
Tableau 5 : Sprint 1 backlog	27
Tableau 6:Description textuelle du cas d'utilisation "S'inscrire"	28
Tableau 7: Description textuelle du cas d'utilisation "S'authentifier"	29
Tableau 8: Description textuelle du cas d'utilisation "Récupérer mot de passe oublié"	30
Tableau 9 : Description textuelle du cas d'utilisation "Mettre à jour profil".....	30
Tableau 10:Test du sprint1	34
Tableau 11:Affectation des cas d'utilisations aux acteurs du sprint 2	38
Tableau 12:Tableau 13: Sprint 2 backlog	39
Tableau 13: Description textuelle du cas d'utilisation "Ajouter utilisateur"	40
Tableau 14:Description textuelle du cas d'utilisation "Modifier utilisateur"	41
Tableau 15:Description textuelle du cas d'utilisation "Supprimer utilisateur".....	41
Tableau 16: Test du sprint 2	44
Tableau 17:Affectation des cas d'utilisations aux acteurs du sprint 3	48
Tableau 18: Sprint 3 backlog	49
Tableau 19:Description textuelle du cas d'utilisation "Gérer famille articles"	51
Tableau 20: Description textuelle du cas d'utilisation " Gérer articles "	51
Tableau 21: Test du sprint 3	55
Tableau 22: Affectation des cas d'utilisations aux acteurs du sprint 4.....	59
Tableau 23: Sprint 4 backlog	59
Tableau 24: Description textuelle du cas d'utilisation "Gérer terrasse"	61
Tableau 25:Description textuelle du cas d'utilisation "Gérer tables"	62
Tableau 26:Test du sprint 4	66
Tableau 27:Affectation des cas d'utilisations aux acteurs du sprint 5	69
Tableau 28:Sprint 5 backlog	70
Tableau 29:Description textuelle du cas d'utilisation "Gérer commandes"	71
Tableau 30:Description textuelle du cas d'utilisation "Gérer demandes"	72

Liste des tableaux

Tableau 31:Description textuelle du cas d'utilisation "Passer commandes et demandes"	72
Tableau 32:Test du sprint 5	75
Tableau 33:Affectation des cas d'utilisations aux acteurs du sprint 6.....	80
Tableau 34: Sprint 6 backlog	80
Tableau 35 : Consulter des statistiques sur les commandes	81
Tableau 36: Consulter la recette du jour	82
Tableau 32:Test du sprint 5	82

Introduction générale

Dans un monde en constante évolution technologique, le secteur de la restauration cherche continuellement à optimiser ses opérations et à améliorer l'expérience client. C'est dans cette optique que notre projet de fin d'études a été conçu, visant à développer une application web et mobile de prise de commandes pour les restaurants. Ce système est destiné à centraliser et à fluidifier les interactions entre les clients et le personnel, en simplifiant le processus de commande grâce à une interface conviviale et efficace.

L'objectif de ce projet est de minimiser les complexités associées aux opérations quotidiennes des restaurants, tout en offrant une expérience enrichie et personnalisée aux clients. Ce rapport détaille chaque phase du développement de l'application, de l'analyse préliminaire des besoins à la conception et mise en œuvre des fonctionnalités, en passant par les tests de validation et l'ajustement continu selon les retours utilisateurs. Notre démarche adopte une méthode agile, permettant une évolution du projet en accord avec les attentes changeantes et les exigences de l'industrie de la restauration.

Nous sommes convaincus que l'application représente une avancée significative dans la technologie de gestion des restaurants, offrant des solutions qui favorisent l'efficacité, la rapidité et la qualité du service. Cette innovation vise à transformer les pratiques traditionnelles et à intégrer de manière fluide les avancées numériques dans le domaine de la restauration.

Chapitre 1 : Présentation générale du projet

Introduction

Dans ce chapitre, nous commençons par présenter le projet en le situant dans son cadre général. Nous proposons alors d'introduire l'environnement du stage à travers une présentation de l'organisme d'accueil de ce projet puis de décrire le sujet à traiter ainsi que le travail demandé.

1. Présentation de l'organisme d'accueil



Figure 1 : Logo de l'organisme d'accueil

« ALL TECH Company » est une société située à Djerba, spécialisée dans le domaine de la sécurité électronique et informatique. Offrant une gamme étendue de solutions innovantes telles que les caméras de surveillance, les systèmes d'alarme et de domotique, la détection incendie et gaz, l'automatisme des portes, et la sécurité informatique, ALL TECH est une entreprise dynamique et en constante évolution avec les avancées technologiques. Dirigée par son gérant, Fahd Bel Hadj Ali, la société bénéficie d'une expertise approfondie pour répondre efficacement aux besoins croissants en matière de sécurité.

2. Cadre général du projet

Le projet est développé dans le cadre du mémoire de fin d'études pour l'obtention du diplôme de licence appliquée en Informatique de gestion, présenté à l'Institut Supérieur des Etudes Technologiques de Djerba.

Réalisé au sein de ALL TECH Company, le stage a pour objectif principal de développer une application de prise de commande pour les secteurs de l'hôtellerie et de la restauration. Cette application vise à optimiser la gestion des commandes et à améliorer l'interaction entre le personnel et les clients grâce à des technologies de pointe.

3. Présentation de l'application

L'application développée est une plateforme numérique destinée à faciliter la prise de commandes dans les restaurants et hôtels. Elle comprend divers sous-systèmes spécialisés pour une gestion efficace des menus, des commandes, des paiements, et de la clientèle. Conçue pour

répondre spécifiquement aux besoins des utilisateurs finaux, l'application intègre des fonctionnalités avancées tels que la personnalisation des commandes, la gestion des préférences des clients, et l'analyse des données de vente.

Développée en utilisant des technologies modernes comme .NET Core pour le backend et Angular et Ionic pour le frontend, l'application est sécurisée, performante et adaptée à une utilisation intuitive aussi bien sur les appareils mobiles que sur les ordinateurs de bureau.

4. Choix de la méthodologie de travail

Il existe plusieurs méthodes agiles qui peuvent être utilisées pour la conception de projets. Les méthodes agiles sont des approches itératives et incrémentales qui se concentrent sur la collaboration entre les membres de l'équipe de projet et sur la réponse rapide aux changements.

4.1. Comparaison entre les méthodes agiles

Le choix entre les méthodes reste toujours un problème déroutant, pour l'éviter nous avons formé un tableau qui représente une comparaison entre deux méthodes agiles les plus connues.

- SCRUM : est une méthode de gestion de projet agile qui se caractérise par des itérations régulières appelées "sprints". Cette approche, dont le nom est inspiré du terme anglais signifiant "mêlée"
- KANBAN : est un terme japonais.

Ci-dessous le tableau comparatif :

Caractéristiques	Scrum	Kanban
Objectifs	Livrables fonctionnels à la fin de chaque sprint	Livraison continue de fonctionnalités
Planning	Planification en début de sprint pour déterminer les tâches à accomplir	Pas de planification fixe, les tâches sont ajoutées au fur et à mesure
Visualisation	Tableau Scrum pour suivre les tâches en cours	Tableau Kanban pour visualiser les tâches en cours et les tâches à faire
Rôles clés	Scrum Master, Product Owner, développeurs	Aucun rôle spécifique
Cadre de temps	Sprints de deux à quatre semaines	Pas de cadres de temps fixes
Flexibilité	Possibilité de modifier les objectifs de sprint à mi-parcours	Grande flexibilité pour ajouter ou supprimer des tâches

Tableau 1:Comparaison de méthodologie de conception

4.2. Scrum

La méthode **SCRUM** est un excellent choix pour gérer des projets complexes, car elle offre des principes clairs pour organiser efficacement le travail.

SCRUM est un cadre qui est composé de plusieurs éléments fondamentaux : des rôles, des événements, des artefacts et des règles.

L'équipe **SCRUM** se compose de trois rôles clés : le Product Owner, l'équipe de développement et le Scrum Master.

- **Product Owner** : Le responsable de la gestion du Product backlog et celui qui établit la priorité des fonctionnalités à développer.
- **L'équipe de développement** : est chargée de transformer les besoins définis par le Product Owner en fonctionnalités utilisables.
- **Un scrum master** : le coach de l'équipe de développement et cherche à maximiser la productivité et le savoir-faire de celle-ci.

Les événements SCRUM sont :

Les événements SCRUM sont des moments clés dans la méthode de gestion de projet SCRUM. Ces événements permettent à l'équipe SCRUM de travailler de manière collaborative et itérative pour atteindre les objectifs du projet.

Les événements SCRUM comprennent :

- **Le Sprint** : un Sprint est une période de temps fixe durant laquelle l'équipe de développement travaille à créer une version utilisable et potentiellement livrable du produit. Les Sprints sont généralement d'une durée de 1 à 4 semaines.
- **Le Sprint Planning** : Cette réunion a pour but de définir les objectifs et les éléments du Product Backlog qui seront inclus dans le Sprint à venir. L'ensemble de l'équipe SCRUM participe à cette réunion qui peut durer jusqu'à huit heures.
- **Le Daily Scrum Meeting** : Il s'agit d'une réunion quotidienne de 15 minutes maximum au cours de laquelle chaque membre de l'équipe de développement répond à trois questions précises : qu'est-ce qu'ils ont réalisé depuis la dernière réunion, qu'est-ce qu'ils ont prévu de faire jusqu'à la prochaine réunion et quels sont les obstacles qui les empêchent de progresser.
- **La Revue de Sprint** : À la fin de chaque Sprint, l'équipe de développement présente les fonctionnalités développées pendant la période et reçoit les commentaires du Product Owner et des parties prenantes. Cette réunion peut durer jusqu'à quatre heures.
- **La Rétrospective de Sprint** : Cette réunion est un moment clé pour l'équipe SCRUM.

Elle permet de revenir sur le Sprint écoulé et de réfléchir sur les processus, les outils et les méthodes de travail utilisés pour identifier les points forts et les points à améliorer. Cette réunion est généralement réservée à l'équipe SCRUM uniquement et peut durer jusqu'à trois heures.

Les artefacts SCRUM sont :

Product backlog : Une liste de fonctionnalités, qui contient exactement ce que doit être réalisé et mise en place par le Product Owner.

Sprint backlog : est une liste du Product backlog dans laquelle en trouve les tâches à faire durant le sprint.

Product incrément : Le résultat d'un sprint.

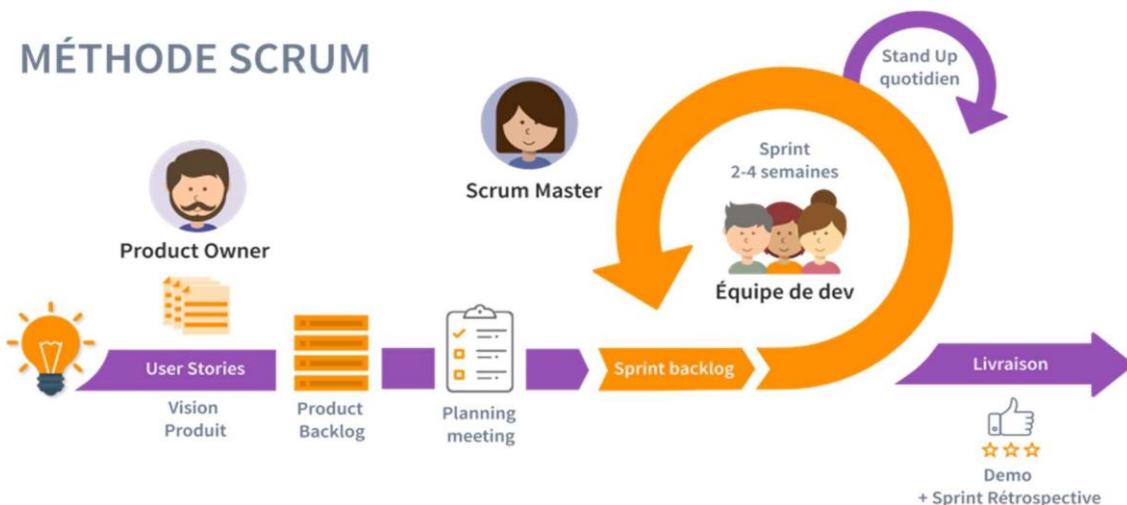


Figure 2: La méthode SCRUM



Figure 3: Équipe SCRUM

5. Langage de modélisation

Il convient de préciser qu'UML est constitué de plusieurs types de diagrammes, chacun servant à modéliser un aspect particulier d'un système logiciel. Voici quelques exemples de diagrammes UML :

- Le diagramme de cas d'utilisation : il permet de modéliser les différentes fonctionnalités attendues du système en se concentrant sur les interactions entre les acteurs et le système.
- Le diagramme de classes : il permet de modéliser les classes et les relations entre elles dans le système, en précisant les attributs et les méthodes de chaque classe.
- Le diagramme de séquence : il permet de modéliser les interactions entre les objets du système pour une séquence d'actions donnée.

En utilisant ces différents types de diagrammes, UML permet de décrire de manière précise et cohérente la structure et le comportement d'un système logiciel complexe, ce qui facilite la communication entre les différents acteurs du projet et permet une meilleure compréhension du système dans son ensemble.

6. Choix Technologiques

6.1. Outils logiciels

Nous commençons par décrire les outils logiciels utilisés tout au long de la conception et développement de notre projet.

- **Visual Studio Code**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégrer.[1]

- **SQLServer**

SQL Server est un système de gestion de bases de données relationnelles développé par Microsoft, souvent utilisé dans des environnements d'entreprise pour sa robustesse et ses intégrations avec d'autres produits Microsoft. [2]



- **Postman**

Postman est un environnement de développement d'API qui aide les utilisateurs à créer, tester, documenter, surveiller et publier la documentation de leurs API.[3]



- **Visual Studio**

Visual Studio est un IDE de Microsoft pour développer des applications sur



diverses plateformes, supportant plusieurs langages de programmation comme C#, C++, et Visual Basic.NET.[4]

- **StarUML**

StarUML est un logiciel de modélisation open source qui prend en charge UML (Unified Modeling Language) pour la modélisation de systèmes et de logiciels. Il fournit onze types de diagrammes différents et il accepte la notation UML 2.0.[5]



- **Microsoft IIS**

Microsoft IIS (Internet Information Services) est un serveur web développé par Microsoft pour héberger des applications web et des sites sur Windows.[6]



- **Android Studio**

Android Studio est un environnement de développement intégré (IDE) créé par Google pour le développement d'applications Android. Il propose des outils pour concevoir des interfaces, coder en Java ou Kotlin, et tester les applications via un émulateur intégré. Android Studio facilite également l'intégration des services Google.[7]



6.2. Langages

- **C#**



C# est un langage de programmation moderne et orienté objet créé par Microsoft.

Utilisé par des millions de développeurs à travers le monde, il est essentiel pour le développement d'applications sur la plateforme .NET. C# est largement utilisé pour construire des applications Windows, des services web, et des applications mobiles. Il offre une combinaison puissante de performance, de flexibilité et de sécurité, faisant de lui un choix privilégié pour de nombreux types de projets logiciels, allant des jeux vidéo aux solutions d'entreprise. [8]

- **TypeScript**

TypeScript est un langage de programmation open source développé par Microsoft. Il s'agit d'un sur-ensemble de JavaScript qui ajoute des fonctionnalités de typage statique au langage, permettant ainsi de détecter et prévenir les erreurs de manière plus efficace lors de la phase de développement. TypeScript est compatible avec tous les navigateurs modernes et peut être utilisé pour développer des applications tant côté serveur que côté client.[9]





- **JAVA**

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems. Utilisé par 9 millions de développeurs dans le monde, ce langage est nécessaire pour le fonctionnement de beaucoup d'applications et de sites Web. Il est la base de la plupart des applications en réseau : des jeux, des téléphones portables à Internet, du contenu web et des logiciels entreprises.[10]



- **HTML**

Le langage de base pour la création de pages web, HTML permet de structurer le contenu et les éléments multimédias sur le web. Il est essentiel pour tout développement web.[11]



- **CSS**

Un langage de style utilisé pour définir la présentation visuelle des pages web, CSS permet de contrôler la mise en page, les couleurs, les polices, et plus encore, rendant les pages web esthétiquement agréables et fonctionnellement pratiques.[12]

6.3. Frameworks



- **Angular**

Angular est un framework de développement d'applications web développé par Google. Il s'appuie sur TypeScript pour créer des applications web robustes et évolutives. Angular adopte le concept de composants réutilisables, permettant aux développeurs de construire des interfaces utilisateur interactives et dynamiques.[13]



- **IONIC**

Ionic est un framework de développement d'applications mobiles open source , utilisé pour créer des applications multiplateformes avec HTML, CSS, et JavaScript. Il repose sur Cordova ou Capacitor pour accéder aux fonctionnalités natives des appareils, permettant ainsi de développer rapidement des applications pour iOS, Android et le web .[14]



- **.NET Core**

.NET est un framework de développement logiciel créé par Microsoft, utilisé pour construire une large variété d'applications, y compris pour Windows, serveurs web, et mobiles. Il permet aux développeurs d'utiliser plusieurs langages de programmation comme C#, F#, et Visual Basic, et intègre une bibliothèque étendue de codes pré-écrits pour faciliter le développement et la maintenance des applications.[15]

- **Microsoft Word**



Un logiciel incontournable de traitement de texte qui fait partie de la suite Microsoft Office. Il est utilisé pour créer, éditer, formater et imprimer des documents textuels, allant des simples mémos aux rapports complexes avec des tableaux et des graphiques. [16]

- **Entity Framework**

Entity Framework est un framework ORM (Object-Relational Mapping)

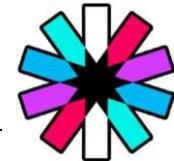


Développé par Microsoft pour .NET. Il permet aux développeurs de travailler avec des données sous forme d'objets et propriétés, sans avoir à écrire du code SQL directement. Entity Framework facilite la manipulation de bases de données en automatisant les tâches de création, lecture, mise à jour et suppression des données (CRUD). Il supporte les modèles de données basés sur le code, le schéma de la base de données, ou une combinaison des deux.[17]

6.4. Technologies

- **JWT**

Un JSON Web Token est un access token (jeton d'accès) aux normes [RFC 7519](#)



qui permet un échange sécurisé de donnée entre deux parties. Il contient toutes les informations importantes sur une entité.[18]

- **SMTP**

Le serveur SMTP de Gmail est utilisé pour l'envoi d'e-mails à partir de comptes Gmail. Pour obtenir des informations de configuration et d'utilisation.[19]



- **Capacitor**

Capacitor est un outil open source qui permet de construire des applications mobiles multiplateformes en utilisant des technologies web, telles qu'Angular, HTML et CSS. Il offre un ensemble de fonctionnalités pour accéder aux fonctionnalités natives des appareils, telles que la caméra, le GPS, les notifications push, etc.[20]

Conclusion

En somme, ce chapitre a été consacré à la présentation de l'environnement de notre projet de fin d'études, ainsi qu'à la méthode de travail que nous avons choisie pour mener à bien ce projet. Nous avons également présenté le langage de modélisation UML, qui nous a permis de représenter les différents éléments de notre projet de manière visuelle et structurée. Grâce à cette présentation, nous avons établi une base solide pour la suite de notre travail, qui consistera à spécifier les besoins du projet avec précision et à détailler les fonctionnalités de notre application de gestion de communication et de scolarité.

Chapitre 2 : Etude préalable

Introduction

Ce chapitre vise principalement à définir les besoins et les exigences du projet, mettant en évidence l'importance de cette étape pour garantir le succès du projet. Il se concentre sur l'identification des utilisateurs du système, l'analyse approfondie de leurs besoins et la spécification des exigences fonctionnelles et non fonctionnelles.

1. Etude de l'existant

L'analyse préliminaire a identifié que le processus de gestion repose fortement sur l'utilisation de supports papier, avec une intervention manuelle significative de la part du personnel. Cette méthode se révèle inefficace, entraînant des retards, des erreurs et une gestion laborieuse des opérations quotidiennes, surtout dans des établissements qui ne disposent pas de grands espaces administratifs.

1.1. Gestion de l'administration

Actuellement, la gestion administrative dans les restaurants est entravée par des pratiques désuètes basées sur le papier, ce qui complique le stockage et l'accès rapide aux informations nécessaires. La proposition d'une application digitalisée centraliserait ces données, rendant leur gestion plus efficiente et moins encline aux erreurs, et libérant de l'espace physique précieux dans des établissements aux ressources limitées.

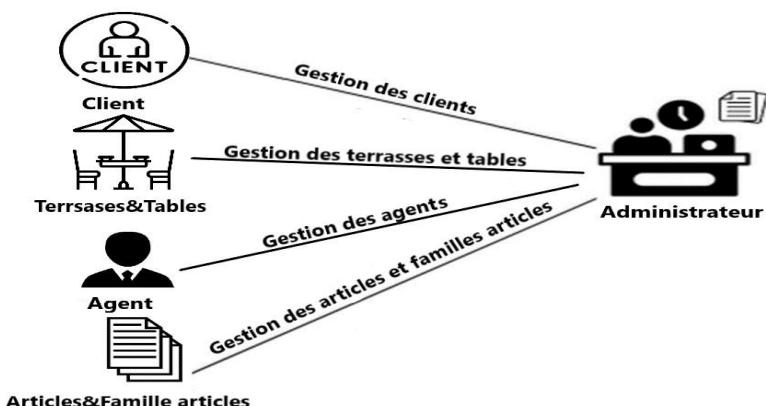


Figure 4: Procédure de gestion de l'administration

1.2. Gestion d'agent

Le personnel, incluant les serveurs et les cuisiniers, est actuellement freiné par des méthodes manuelles pour la prise et la validation des commandes. L'introduction d'une application automatisée simplifierait significativement ces tâches, permettant un traitement plus rapide des commandes et une meilleure coordination entre la cuisine et le service en salle, améliorant ainsi la productivité et réduisant le potentiel d'erreurs.

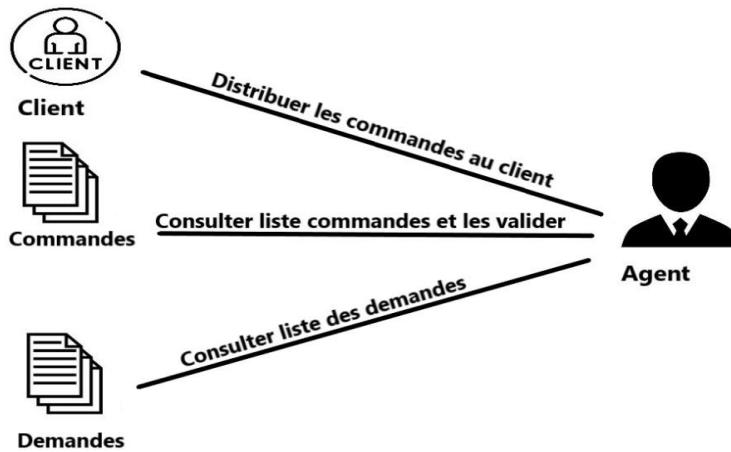


Figure 5: Procédure de gestion d'agent

1.3. Gestion de client

Les clients font souvent face à des attentes prolongées pour passer leurs commandes en utilisant des menus physiques. L'application mobile proposée offrirait une interface intuitive pour visualiser le menu et passer des commandes rapidement, ce qui réduirait considérablement le temps d'attente et améliorerait l'expérience client, tout en permettant une gestion efficace des préférences et des commandes précédentes.

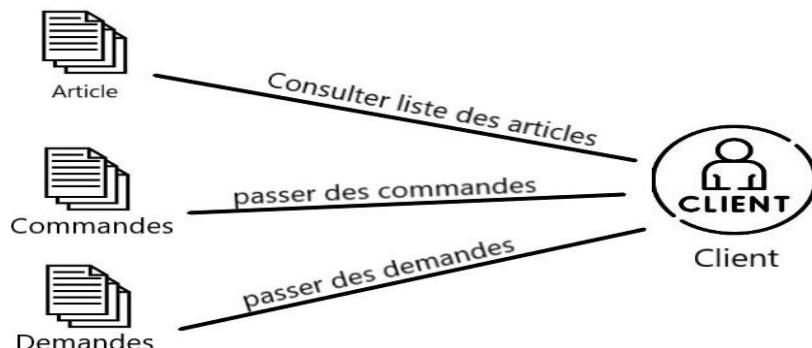


Figure 6: Procédure de gestion de client

2. Objectifs de l'application Web/ Mobile

L'application de prise de commande a pour but de réduire la dépendance aux documents papier et de simplifier l'ensemble du processus de prise de commande. Elle vise à accélérer les transactions, améliorer la coordination interne et offrir une expérience utilisateur optimisée. En rationalisant les opérations, l'application aidera les établissements de petite taille à maximiser l'utilisation de leurs espaces limités, en concentrant leurs ressources sur l'amélioration du service et la satisfaction des clients, plutôt que sur la gestion de documents papier.

3. Spécification des besoins

3.1. Besoins fonctionnels

Les principaux besoins fonctionnels de notre application se résument autour des points suivants : **Le système doit permettre à l'utilisateur de créer un compte, de s'authentifier et de se déconnecter.**

L'application permet à l'administration de :

- Gérer les articles par familles.
- Gérer les clients.
- Gérer les agents.
- Gérer les terrasses.
- Gérer les tables.
- L'authentification des agents.

L'application permet au client de :

- S'inscrire et voir les articles.
- Passer des commandes.
- Le suivi des commandes.
- Mettre à jour son profil.
- Envoyer demandes à l'agent.

L'application permet à l'agent de :

- Visiter les listes des commandes passées dans son Terrasse.
- Consulter liste des demandes.
- Valider ou annuler des commandes.

3.2. Besoins non fonctionnels

En plus des besoins fonctionnels qui décrivent les fonctionnalités spécifiques attendues de l'application, il est également important de prendre en compte les besoins non fonctionnels. Les besoins non fonctionnels se réfèrent aux caractéristiques globales et aux contraintes de l'application :

- **Sécurité** : L'application doit garantir la sécurité des données sensibles. Cela peut impliquer l'utilisation de protocoles de sécurité.

- **Performance** : L'application doit être réactive et performante, en minimisant les temps de chargement et en permettant une utilisation fluide même avec un grand nombre d'utilisateurs et de données.
- **Convivialité** : L'interface utilisateur de l'application doit être conviviale, intuitive et facile à utiliser, tant pour les apprenants que pour les formateurs et l'administration. Une bonne ergonomie et une navigation claire amélioreront l'expérience utilisateur.
- **Fiabilité** : L'application doit être fiable, évitant les pannes et les erreurs, afin de garantir un fonctionnement continu et sans interruption.
- **Évolutivité** : L'application doit être conçue pour être évolutive, capable de prendre en charge une croissance future, tant en termes d'utilisateurs que de fonctionnalités supplémentaires. Elle doit pouvoir s'adapter aux besoins changeants du centre de formation.
- **L'ergonomie** : L'application ne doit pas être compliquée à utiliser par l'utilisateur. Ainsi que les interfaces doivent être homogènes.
- **Compatibilité** : L'application doit être compatible sur tous les navigateurs ainsi que sur les différentes plateformes (IOS/Android).

4. Architecture Logicielle utilisée : MVC

Dans notre projet, nous avons fait recours à l'architecture MVC pour assurer une bonne qualité en termes de performance, et car les fonctionnalités auxquelles notre projet doit répondre exigent la mise en place de cette architecture. L'architecture MVC comprend 3 composants :

- **Modèle (Model)** : contient la logique de l'application, l'accès aux données et les règles de gestion.
- **Vue (View)** : contient une partie de l'interface utilisateur de notre application (HTML).

- **Contrôleur (Controller)** : gère la communication entre le Modèle et la Vue, en traitant les entrées utilisateurs et en mettant à jour le Modèle ou la Vue (.cs / .js).

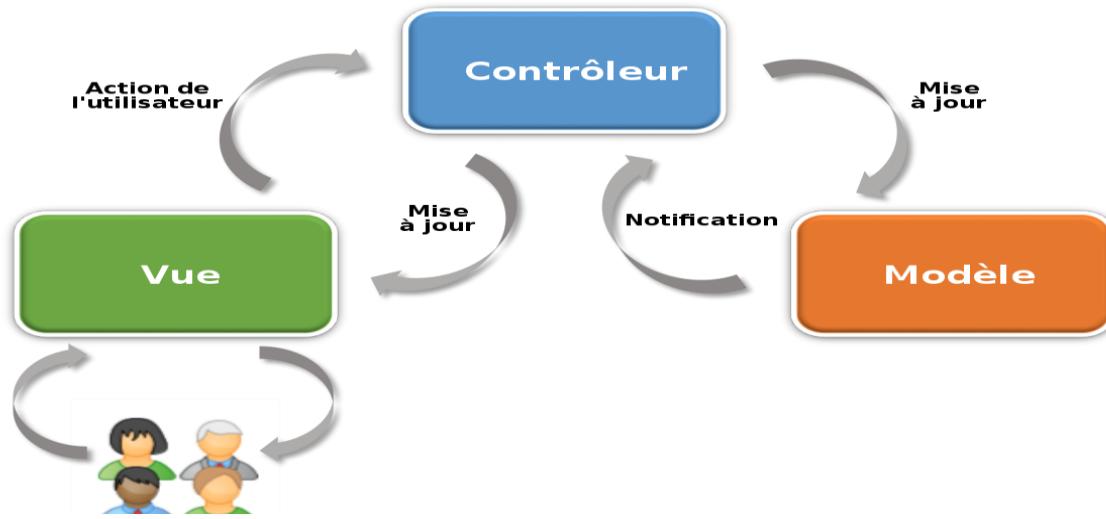


Figure 7: Architecture MVC

5. Architecture système déploiement utilisée : 3-tiers

Dans notre projet, nous avons fait recours à l'architecture 3-tiers pour assurer une bonne qualité en termes de performance, et car les fonctionnalités auxquelles notre projet doit répondre exigent la mise en place de cette architecture. [18]

L'architecture 3-tiers comprend 3 niveaux :

- Niveau présentation : contient une partie de l'interface utilisateur de notre application. (html)
- Niveau traitement métier des données, correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative (.cs / .js)
- Niveau de données : La couche d'accès aux données contient des méthodes permettant de se connecter à une base de données et d'effectuer des insertions, des mises à jour, des suppressions et l'obtention de données à partir d'une base de données basée sur nos données d'entrée.

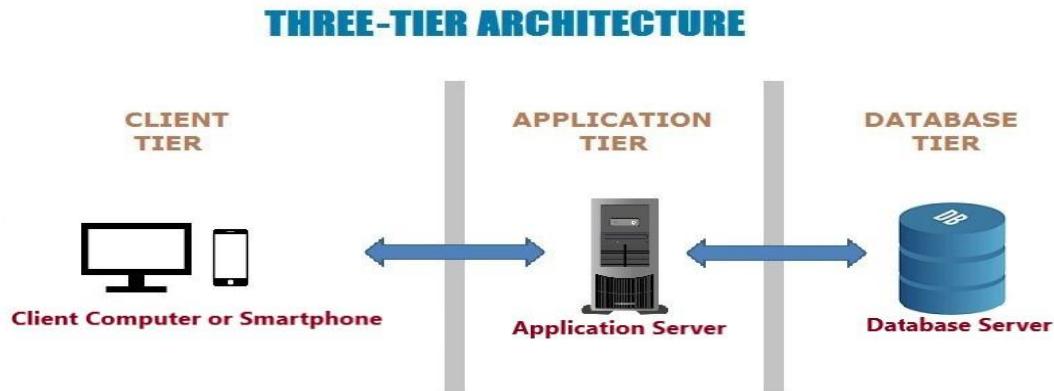


Figure 8:Architecture 3_tiers

Les avantages d'une architecture 3-tiers :

- Sécurité améliorée (le client n'a pas un accès direct à la base de données).
- Améliorer l'intégrité des données.
- Meilleure réutilisation.
- Haut degré de flexibilité.

L'inconvénient d'une architecture 3-tiers :

- Augmenter la complexité.

6. Product Backlog

6.1. Identification des acteurs

Notre application sera utilisée principalement par trois acteurs :

- **Admin** : L'administrateur du restaurant qui est responsable de la gestion globale du restaurant.
- **Client** : Les clients sont les personnes qui visitent le restaurant et peuvent passer des commandes sur des plats/boissons...
- **Agent** : Les agents, qui sont également les serveurs, sont responsables de l'approbation et de la validation des commandes.

6.2. Eléments du product backlog

Product Backlog : est une liste qui contient les tâches nécessaires que l'équipe projet doit réaliser. **User Stories** : Sont de brèves descriptions des fonctionnalités introduites par le Product Owner.

Ordre de priorités dans le Product backlog :

Nous avons distingué trois niveaux de priorité : Faible, moyen et élevé. Pour ce faire, nous adoptons le modèle suivant :

« En tant que <Acteur>, je peux »

Remarque :

Nous avons créé une nouvelle entité <Utilisateur> parente pour les trois acteurs suivants (admin, client, agent) pour regrouper les actions et les fonctionnalités communes.

Le tableau 2 décrit le backlog du produit :

Module	Fonctionnalité	Id	User story	Priorité	Story points
Module Gestion des comptes	Gérer compte	1	En tant qu'utilisateur, je peux m'authentifier	Elevé	4
		2	En tant qu'un client, je peux modifier mon compte		2
		3	En tant qu'administrateur, je peux visualiser les listes des clients, des agent , des articles et familles articles		2
		4	En tant que client je peux créer un compte ou connecter avec Google ou Facebook		4
	Réinitialiser mot de passe	6	En tant qu'un client, je peux réinitialiser mon mot de passe si je l'ai oublié	Faible	4
Module Gestion d'utilisateur	Gérer utilisateur	1	En tant qu'administrateur, je peux archiver et désarchiver un client ou un agent ou un article.	Elevé	3
		2	En tant qu'administrateur, je peux ajouter, modifier et supprimer un client.		3
		3	En tant qu'administrateur, je peux ajouter, modifier et supprimer agent.		4
		4	En tant qu'administrateur je peux attribuer une terrasse à un agent.		4

Module Gestion des articles	Gérer les familles articles	1	En tant qu'administrateur, je peux créer des familles articles.	Elevé	3
		2	En tant qu'administrateur, je peux visualiser les familles articles.		2
		3	En tant qu'administrateur, je peux modifier et supprimer les familles articles		3
		4	En tant que client, je peux visualiser les familles articles		
	Gérer les articles	1	En tant qu'administrateur, je peux créer des articles selon leurs familles articles	Elevé	4
		2	En tant qu'administrateur, je peux visualiser les articles.		2
		3	En tant qu'administrateur, je peux modifier et supprimer les articles		3
		4	En tant que client, je peux visualiser, les articles		4

Module Gestion des terrasses et tables	Gérer les terrasses	1	En tant qu'administrateur, je peux créer des terrasses	Elevé	3
		2	En tant qu'administrateur, je peux visualiser les terrasses		2
		3	En tant qu'administrateur, je peux modifier et supprimer les terrasses		3

	Gérer les tables	1	En tant qu'administrateur, je peux créer des tables selon leurs terrasses		4
		2	En tant qu'administrateur, je peux visualiser les tables.	Elevé	2
		3	En tant qu'administrateur, je peux modifier et supprimer les tables		3

Module Gestion des commandes et demandes	Gérer les commandes	1	En tant qu'un client, je peux passer une commande.		3
		2	En tant qu'un client, je peux visualiser les commandes que j'ai passé et ses état.		2
		3	En tant qu'agent, je peux visualiser la liste des commandes dans mon terrasses	Elevé	2
		4	En tant qu'un agent, je peut valider les commandes et changer ses états.		3
Gérer les demandes		1	En tant qu'un client, je peux passer une demande a l'agent.		3
		2	En tant qu'un agent je peux visualiser la liste des demandes.	Elevé	2
		3	En tant qu'un agent je peux accepter les demandes par priorité.		3

Module Gestion de tableau de bord	Gérer le tableau de bord	1	En tant qu'administrateur, je peux consulter les commandes du jour	Elevé	3
		2	En tant qu'administrateur, je peux visualiser la recette du jour		2
		3	En tant qu'administrateur, je peux visualiser les meilleurs agents		3
		4	En tant qu'administrateur, je peux visualiser les meilleurs clients		3

Tableau 2 : Le backlog du produit

7. Diagramme de cas d'utilisation global

La figure 9 représente le diagramme de cas d'utilisation global qui regroupe les fonctionnalités demandées.

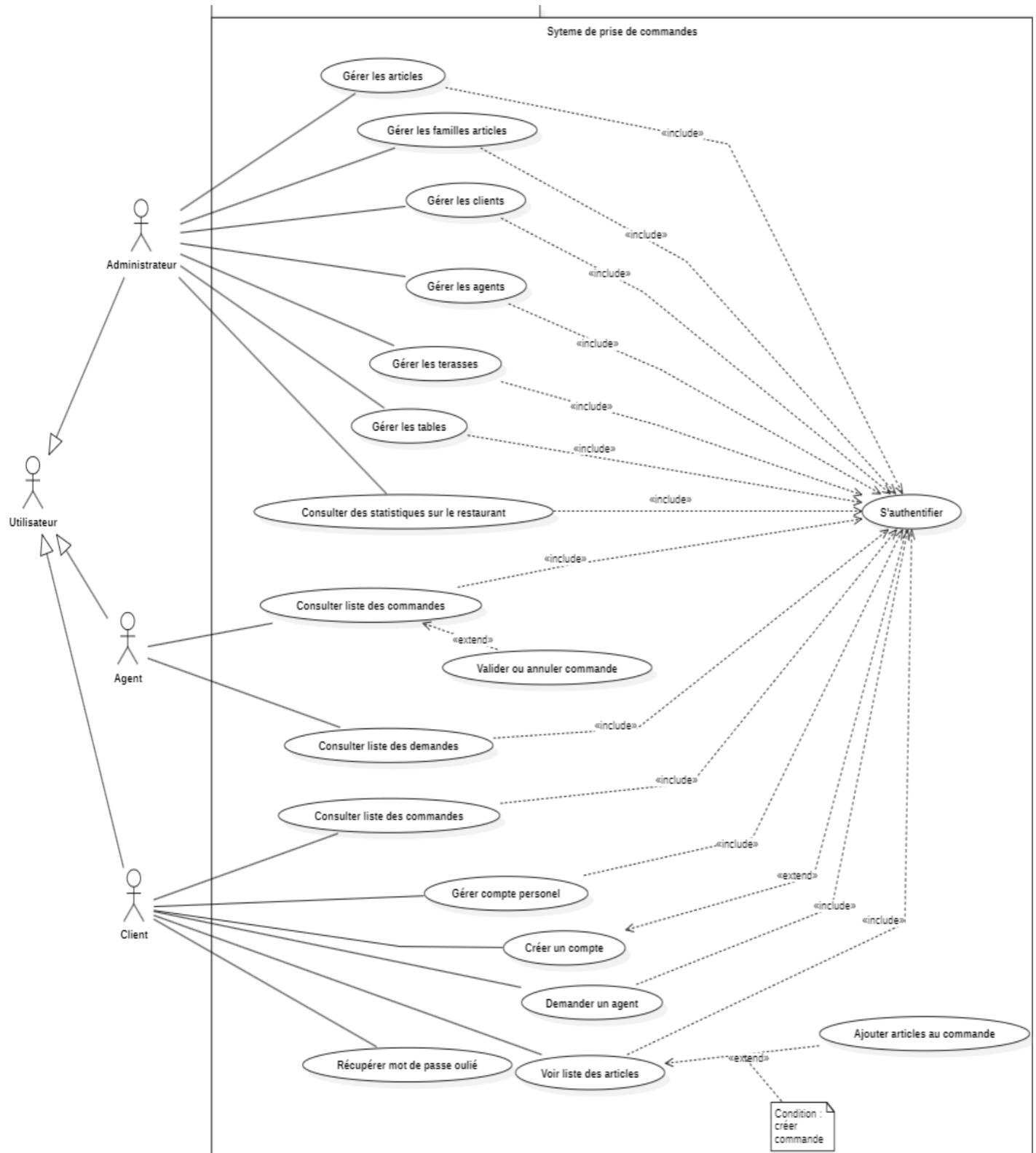


Figure 9: Diagramme de cas d'utilisation global

8. Diagramme de classe analyse

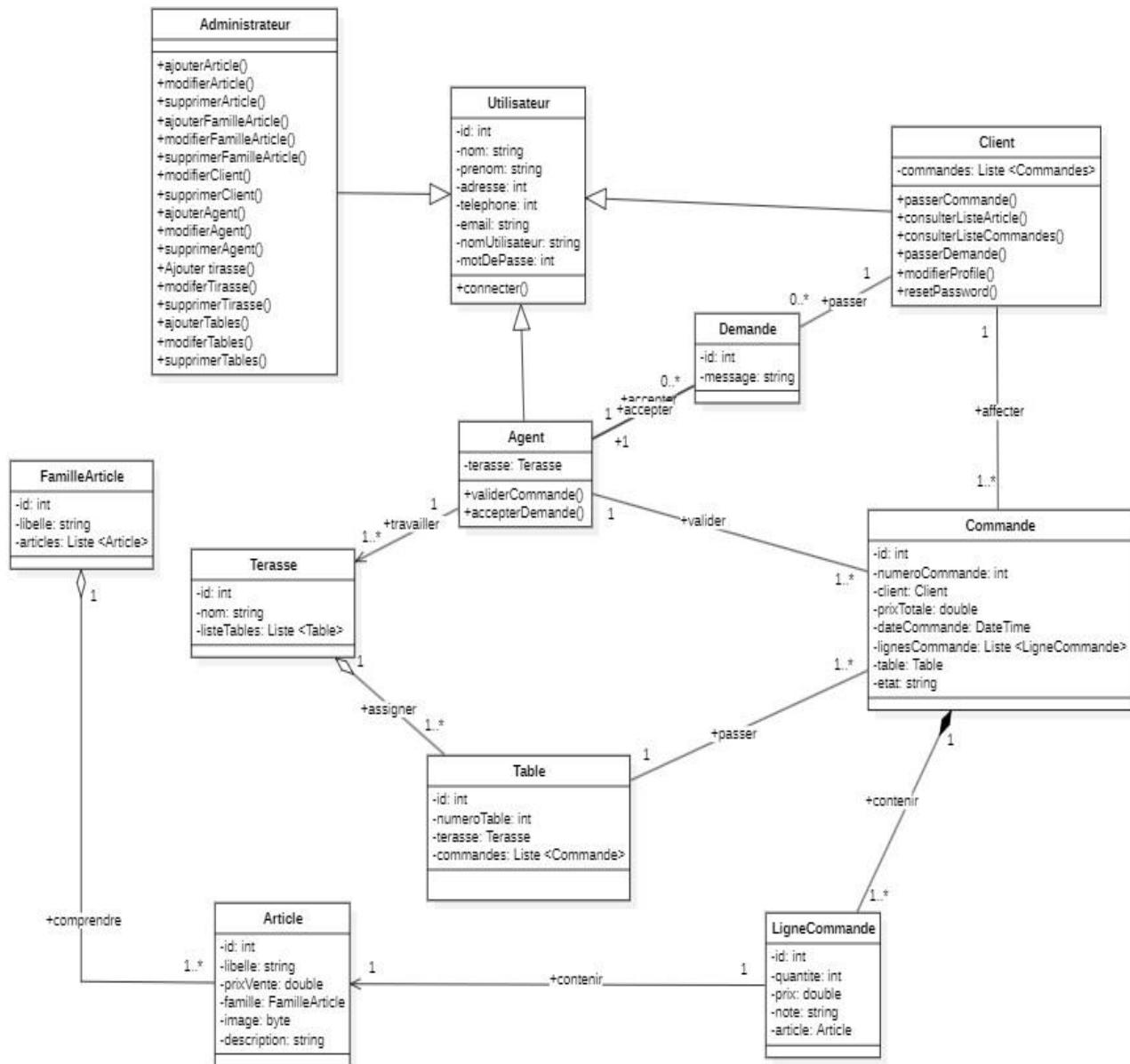


Figure 10: Diagramme de classe analyse

2. Planification des sprints

Après avoir priorisé les différentes "User Stories", nous avons organisé les besoins fonctionnels en 4 sprints. Chaque sprint se termine par une version complète du produit. Cette approche itérative nous permet de livrer des fonctionnalités de manière progressive tout en prenant en compte les retours des utilisateurs.

Sprints	Module	Durée
Sprint 1	Module gestion des comptes	3 semaines
Sprint 2	Module gestion des utilisateurs	3 semaines
Sprint 3	Module gestion des articles	3 semaines
Sprint 4	Module gestion des terrasses et tables	2 semaines
Sprint 5	Module gestion des commandes	2 semaines
Sprint 6	Module gestion de tableau de bord	3 semaines

Tableau 3 : Planning des sprints

Conclusion

Dans ce chapitre nous avons défini la liste des besoins fonctionnels et non fonctionnels et les acteurs et de ce projet. Ensuite, nous avons présenté les éléments du backlog du produit, puis, la planification des sprints. Dans ce qui suit, nous détaillerons les étapes de chaque sprint.

Chapitre 3 : Sprint 1 - Module gestion des comptes

Introduction

Dans ce chapitre, nous abordons le premier sprint qui se concentre sur le module de gestion des comptes. Nous commencerons par décrire les fonctionnalités de ce sprint à l'aide de diagrammes de cas d'utilisation, puis nous examinerons son comportement à travers des diagrammes de séquence. Enfin, nous présenterons le diagramme de classes, suivi de la mise en place des interfaces de réalisation. Ce sprint constitue une étape cruciale dans le développement de notre application, car il établit les bases solides pour la gestion des comptes des utilisateurs.

1. Backlog du sprint 1

Le premier sprint se concentre sur les fonctionnalités clés de démarrage, notamment l'authentification des utilisateurs, l'inscription de compte, la récupération de mot de passe oublié et la confirmation par e-mail. Ces fonctionnalités sont essentielles pour garantir un processus fluide et sécurisé dès le début de l'utilisation de l'application.

1.1. Affectation des cas d'utilisations pour les acteurs

Acteurs	Cas d'utilisation
Utilisateur (agent, client et admin)	Récupérer le mot de passe oublié
	Mettre à jour son profil
Client	S'inscrire

Tableau 4 :Classification des cas d'utilisations par acteur

1.2. Sprint backlog

Le sprint backlog est un outil qui rend visible tout le travail en déterminant les tâches de chaque user story dans le backlog Product. Notre sprint backlog est décrit comme suit :

Id	User stories	Id-tâches	Tâches
1	En tant qu'utilisateur, je peux m'authentifier	1.1	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « S'authentifier »

		1.2 1.3	-Développer « S'authentifier » -Tester
2	En tant que client, je peux modifier mon profil	2.1 2.2 2.3	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Mettre à jour profil » -Développer « Mettre à jour profil » -Tester
3	En tant que client, je peux récupérer le mot de passe oublié	3.1 3.2 3.3	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Récupérer mot de passe oublié » -Développer « Récupérer mot de passe oublié » -Tester
4	En tant que client, je peux créer compte ou connecter avec Google ou Facebook.	4.1 4.2 4.3	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « S'inscrire » -Développer « S'inscrire » -Tester

Tableau 5 : Sprint 1 backlog

2. Spécification fonctionnelle

Pour donner une vision globale du comportement fonctionnel du système, on utilise les diagrammes de cas d'utilisation. Dans cette partie, nous commençons par présenter le diagramme de cas d'utilisation global correspondant au sprint 1, par la suite, nous raffinerons par acteur.

2.1 Diagramme de cas d'utilisation global de sprint 1

La figure ci-dessous représente les cas d'utilisations qui sont notre centre d'intérêt tout au long de ce sprint

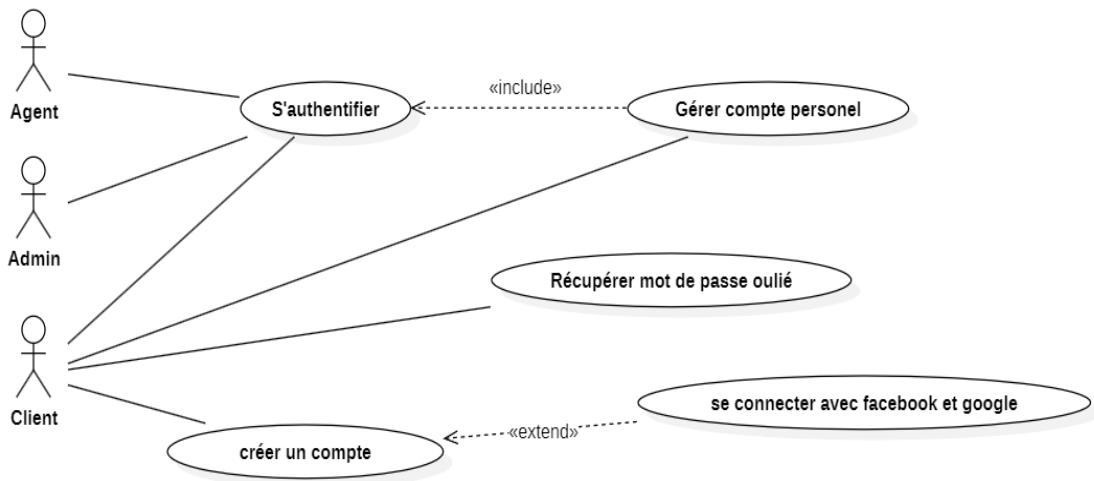


Figure 11: Diagramme de cas utilisation global du sprint 1

2.2 Description textuelle des cas d'utilisation

Dans cette partie nous décrivons textuellement les principaux cas d'utilisation qui apparaissent dans le diagramme de cas utilisation global de sprint 1.

Cas d'utilisation	S'inscrire
Acteurs	Client
Précondition	Le login n'existe pas déjà dans le système.
Postcondition	Le client est inscrit avec succès et peut accéder à son compte.
Scénario nominal	<ol style="list-style-type: none"> 1) L'application génère le formulaire de création de compte pour chaque client. 2) Le client saisit les informations nécessaires et les enregistre. 3) Le client clique sur l'un des icônes de Google ou Facebook pour s'inscrire. 4) Le client peut maintenant accéder à son compte et commencer à utiliser les fonctionnalités de l'application.
Scénario alternatif	<ol style="list-style-type: none"> 1) Le client introduit des données incorrectes. 2) Si le compte existe déjà le système affiche un message d'erreur.

Tableau 6: Description textuelle du cas d'utilisation "S'inscrire"

Cas d'utilisation	S'authentifier
Acteurs	Utilisateur (client, Agent, Administrateur)
Précondition	L'utilisateur doit posséder un compte et connaître son login et mot de passe
Postcondition	L'utilisateur est authentifié avec succès et accède à l'interface tableau de bord appropriée en fonction de son rôle (apprenant, formateur ou administrateur).
Scénario nominal	<ul style="list-style-type: none"> 1) Le système affiche le formulaire d'authentification. 2) L'utilisateur introduit son login et le mot de passe. 3) Le système vérifie les données introduites et affiche l'interface de tableau de bord correspondante au rôle de l'utilisateur.
Scénario alternatif	<ul style="list-style-type: none"> 1) Si l'utilisateur introduit des données incorrectes (login ou mot de passe incorrect), le système affiche un message d'erreur. 2) L'utilisateur peut réessayer de saisir les informations correctes ou demander une réinitialisation de son mot de passe. 3) Si compte de l'utilisateur est non validé, le système affiche un message d'erreur. 4) Si l'utilisateur est archivé, le système affiche un message d'erreur.

Tableau 7: Description textuelle du cas d'utilisation "S'authentifier"

Cas d'utilisation	Récupérer mot de passe oublié
Acteurs	Client
Précondition	Le client a oublié son mot de passe et souhaite le récupérer.
Postcondition	Le client reçoit un e-mail avec un lien de réinitialisation de mot de passe.
Scénario nominal	<ul style="list-style-type: none"> 1) Le client sélectionne l'option "Mot de passe oublié" sur l'écran d'authentification. 2) Le système affiche un formulaire où le client doit saisir son adresse e-mail associée à son compte. 3) Le client saisit son adresse e-mail et soumet le formulaire. 4) Le système vérifie que l'adresse e-mail saisie existe dans la base de données. 5) Si l'adresse e-mail est valide, le système génère un lien de réinitialisation de mot de passe unique et l'envoie à l'adresse e-mail de client.

	<p>6) Le client reçoit l'e-mail de réinitialisation de mot de passe avec le lien.</p> <p>7) Le client clique sur le lien de réinitialisation dans l'e-mail.</p> <p>8) Le système affiche un formulaire où le client peut saisir un nouveau mot de passe.</p> <p>9) Le client saisit le nouveau mot de passe et le confirme.</p> <p>10) Le système vérifie la validité du nouveau mot de passe et le met à jour dans la base de données.</p> <p>11) Le client est redirigé vers l'écran d'authentification où il peut se connecter avec son nouveau mot de passe.</p>
Scénario alternatif	<p>1) Si l'adresse e-mail saisie n'est pas valide (inexistante dans la base de données), le système affiche un message d'erreur demandant au client de vérifier l'adresse e-mail saisie.</p>

Tableau 8: Description textuelle du cas d'utilisation "Récupérer mot de passe oublié"

Cas d'utilisation	Mettre à jour profil
Acteurs	Client
Précondition	Le client doit être authentifié dans le système.
Postcondition	Les informations du profil de client sont mises à jour.
Scénario nominal	<p>1) Le client accède à son profil dans l'application.</p> <p>2) Le client modifie les informations qu'il souhaite mettre à jour (par exemple, nom, prénom, adresse, numéro de téléphone, etc.).</p> <p>3) Le client enregistre les modifications.</p> <p>4) Le système met à jour les informations du profil de client.</p>
Scénario alternatif	<p>1) Le client ne souhaite pas effectuer de modifications et annule l'opération.</p> <p>2) Le système ne met pas à jour les informations du profil de client.</p>

Tableau 9 : Description textuelle du cas d'utilisation "Mettre à jour profil"

2.3 Description comportementale des histoires du sprint 1

Dans le cadre de ce premier sprint, nous avons utilisé des diagrammes de séquence pour illustrer le comportement du système. Ces diagrammes permettent de représenter les interactions entre les objets et les acteurs du système au fil du temps, en se concentrant sur un cas d'utilisation spécifique, également appelé une histoire d'acteur. Dans la section suivante, nous présenterons les diagrammes de séquence correspondant aux cas d'utilisation clés de notre sprint 1.



Dans notre contexte, le terme "Boundary" fait référence à une interface graphique ou à une autre interface Homme/Machine, telle qu'un écran tactile ou un PocketPC.

La classe "Entity" quant à elle est utilisée pour modéliser des informations durables et souvent persistantes. Enfin, la classe "Control" est utilisée pour représenter la coordination, le déroulement et le contrôle d'autres objets.

- **Diagramme de séquence « S'inscrire »**

La figure 12 illustre l'enchaînement des séquences nécessaires à la réalisation du cas d'utilisation s'inscrire pour l'utilisateur.

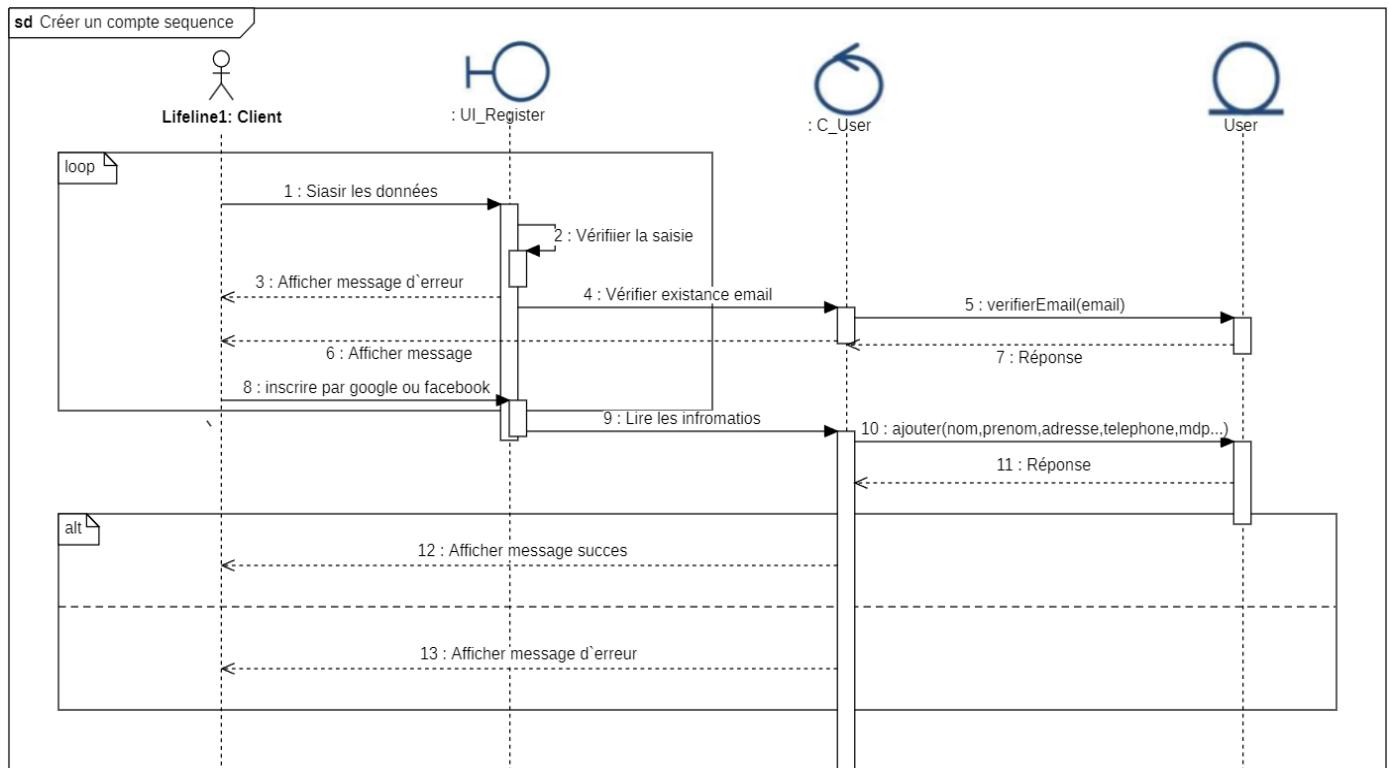


Figure 12: Diagramme de séquence de cas d'utilisation « S'inscrire »

- **Diagramme de séquence « S'authentifier »**

La figure 13 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation s'authentifier pour l'utilisateur.

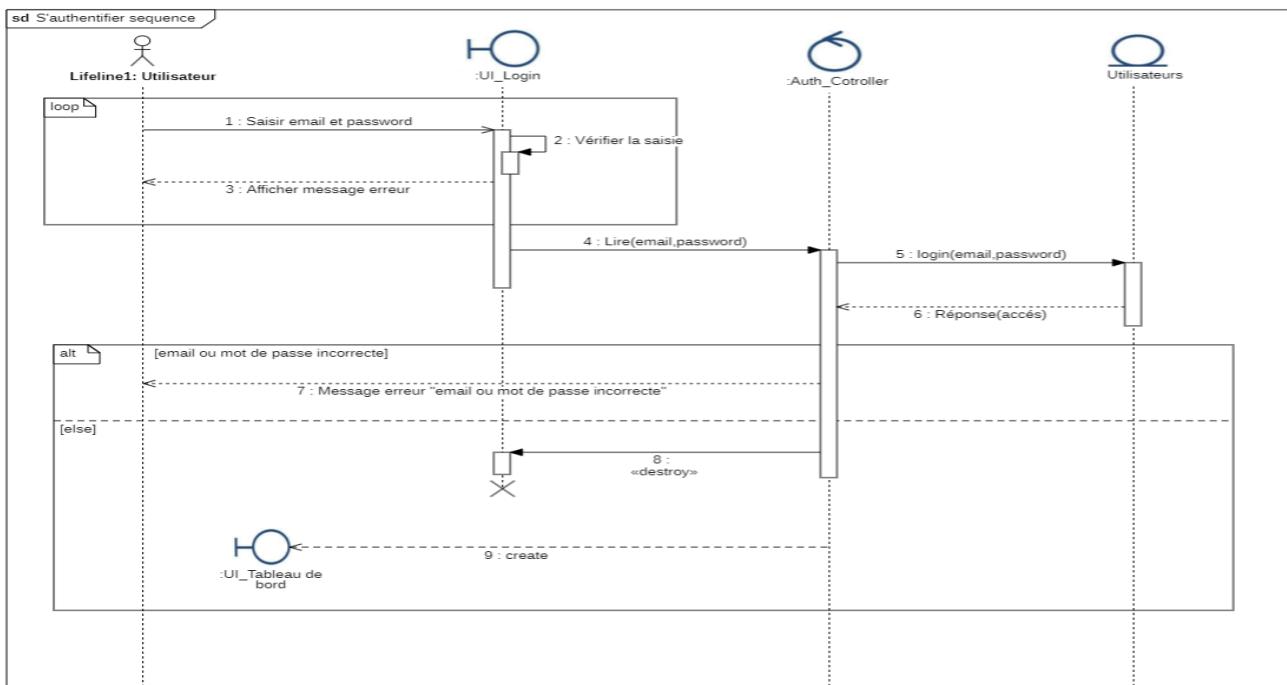


Figure 13: Diagramme de séquence de cas d'utilisation "S'authentifier"

- **Diagramme de séquence « Mettre à jour profil »**

La figure 14 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation s'authentifier pour l'utilisateur.

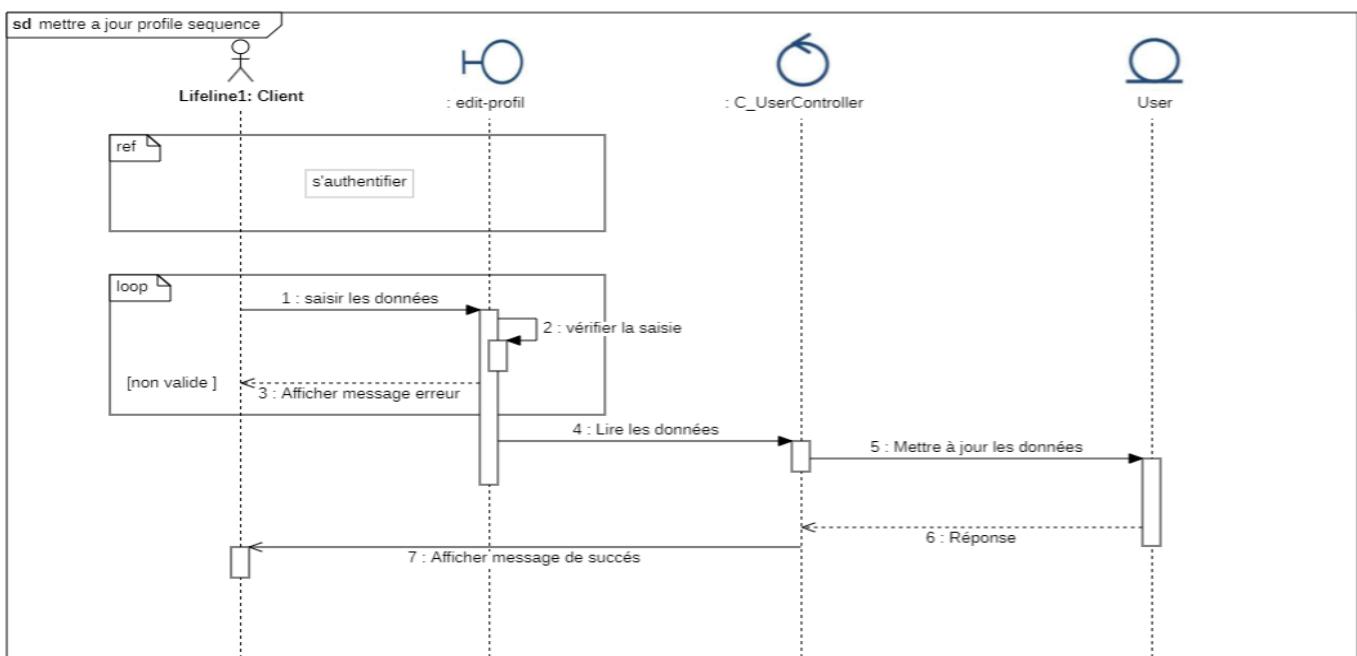


Figure 14: Diagramme de séquence de cas d'utilisation "Mettre à jour profile"

- **Diagramme de séquence « Récupérer mot de passe oublié »**

La figure 15 illustre l'enchaînement des séquences nécessaires à la réalisation du cas d'utilisation s'authentifier pour l'utilisateur.

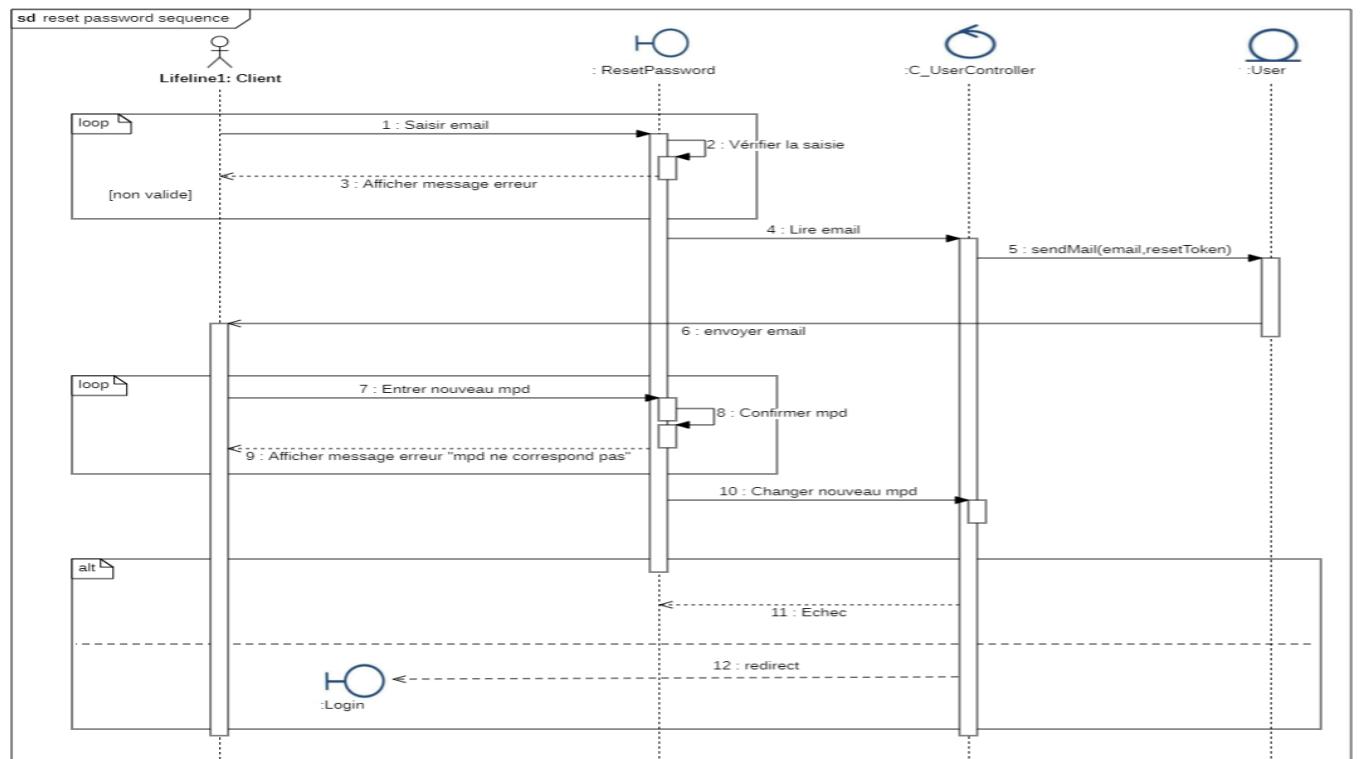


Figure 15: Diagramme de séquence de cas d'utilisation "Récupérer mot de passe oublié"

2.4 Diagramme de classe de sprint 1

Pour le premier sprint de notre projet, nous avons établi le diagramme de classes illustré par la figure 16. Il comporte 4 classes.

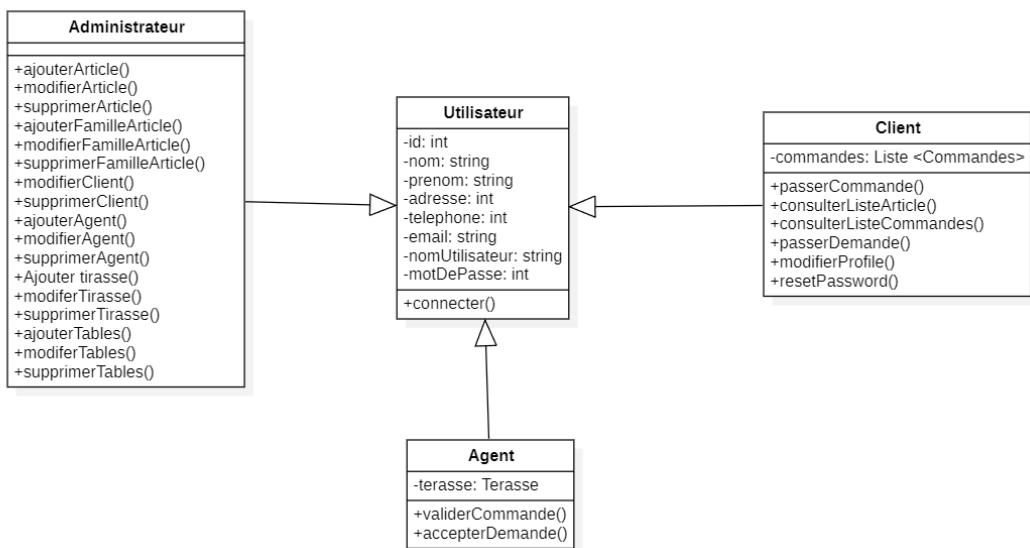


Figure 16: Diagramme de classe de sprint 1

3. Test et interfaçage

3.1. Test sprint 1

Cas	Résultat
Récupérer le mot de passe oublié	Conforme
Mettre à jour son profil	
S'inscrire	

Tableau 10:Test du sprint1

3.2. Interfaçages

Pour illustrer la réalisation du premier sprint, nous présenterons les interfaces des principales fonctionnalités offertes par le module gestion de compte.

- **Interface de login**

Figure 17: Interface login

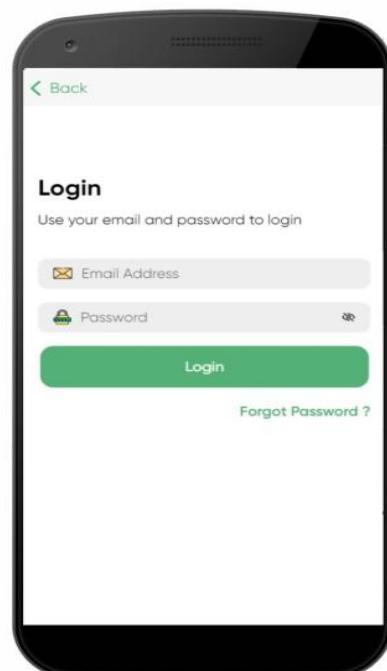


Figure 18: Interface login mobile

- **Interface de registre**

The screenshot shows a registration form titled "Register". The page header includes a back arrow and the text "Create an account to explore more". Below the header are seven input fields: "Prenom" (placeholder icon), "Name" (placeholder icon), "Email Address" (placeholder icon), "Phone Number" (placeholder icon), "Address" (placeholder icon), "Password" (placeholder icon with a visibility toggle), and "Confirm Password" (placeholder icon with a visibility toggle). A file upload section labeled "Upload Photo" with a "Choisir un fichier" button and a message "Aucun fichier choisi" is present. At the bottom is a large green "REGISTER" button.

Figure 19: Interface registre

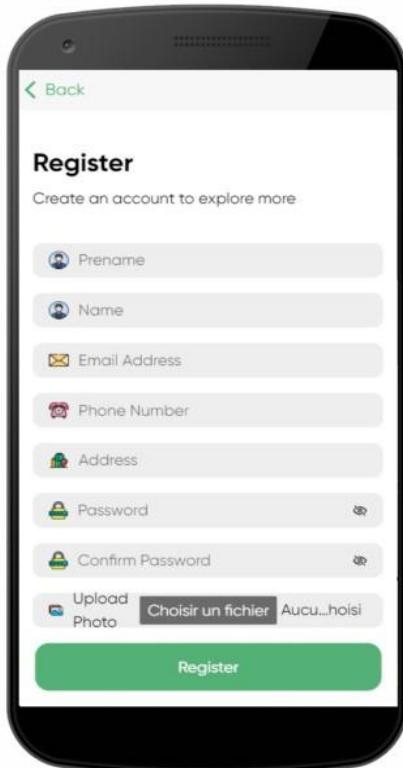


Figure 20: Interface registre mobile

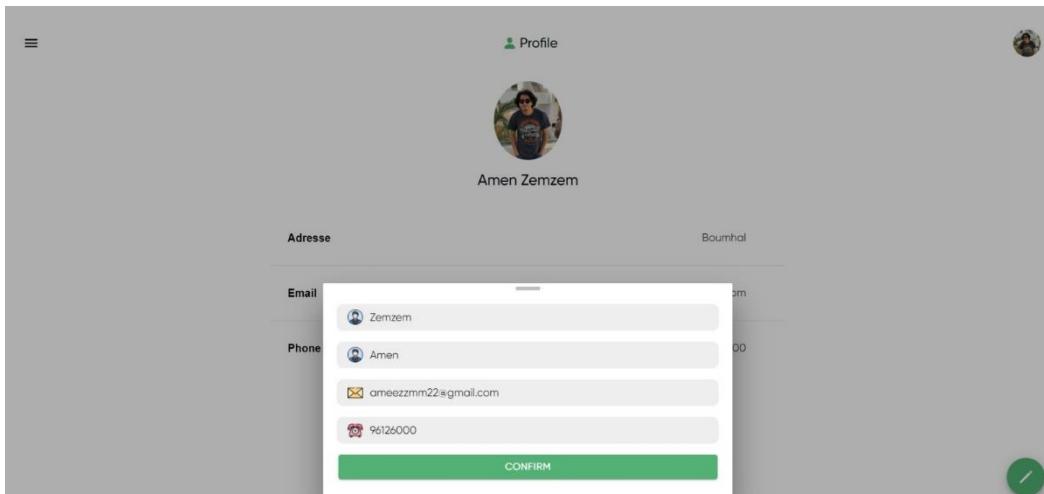
- **Interface de récupération de mot de passe**



The screenshot shows a 'Forgot Password' page. At the top left is a back arrow. Below it, the title 'Forgot Password' is displayed in bold. A sub-instruction 'Enter your email used on your account' follows. A light gray input field labeled 'Email Address' contains the placeholder '@'. To its right is a green rectangular button with the word 'CONFIRM' in white capital letters.

Figure 21: Interface de récupération de mot de passe

- **Interface de mettre à jour le profil**



The screenshot shows a 'Profile' update interface. At the top center is a user icon and the text 'Profile'. Below it is a circular profile picture of a person. Underneath the picture is the name 'Amen Zemzem'. On the left side, there's a vertical menu bar with three horizontal lines. In the center, there's a section titled 'Adresse' with a 'Profile' icon. To the right of the address section is the name 'Boumhal'. Below this, there are two tabs: 'Email' (which is selected) and 'Phone'. Under the 'Email' tab, there are four input fields: 'Zemzem' (with a person icon), 'Amen' (with a person icon), 'ameezznm22@gmail.com' (with an envelope icon), and '96126000' (with a phone icon). To the right of these fields is a vertical progress bar showing '5m' at the top and '00' at the bottom. At the bottom of the input area is a green 'CONFIRM' button. On the far right, there's a circular edit icon with a pencil symbol.

Figure 22 : Interface de mettre à jour de profil de client

Conclusion

Dans ce chapitre, nous avons planifié et exécuté le premier sprint de notre projet. Nous avons identifié les différentes fonctionnalités à réaliser et les avons décrites en détail à travers les descriptions fonctionnelles et comportementales des histoires. Nous avons également élaboré le diagramme de classes correspondant au sprint et avons procédé à la réalisation des interfaces nécessaires. Nous sommes maintenant prêts à passer au chapitre suivant où nous aborderons le deuxième sprint et poursuivrons le développement de notre application.

Chapitre 4 : Sprint 2 - Module gestion des utilisateurs

Introduction

Le chapitre 4 se concentre sur le sprint 2 dédié à la gestion des utilisateurs. Nous décrirons les fonctionnalités telles que l'ajout, la gestion des rôles, la modification et la suppression des utilisateurs. Les diagrammes de cas d'utilisation et de séquence seront présentés pour illustrer les interactions. Enfin, les interfaces de réalisation seront exposées.

1. Backlog du sprint 2

1.1. Affectation des cas d'utilisations aux acteurs

Nous allons classifier les trois cas d'utilisation du deuxième sprint par acteur. Nous trouvons ci-dessous un tableau qui montre l'affectation de chaque acteur à ses cas d'utilisations.

Acteurs	Cas d'utilisation
Administrateur	Gérer utilisateur (agent, client)

Tableau 11:Affectation des cas d'utilisations aux acteurs du sprint 2

1.2. Sprint backlog

Le sprint backlog du sprint 2 a pour objectif de rendre visible le travail à réaliser en décomposant les tâches associées à chaque user story du backlog produit. Voici la description de notre sprint backlog pour ce sprint :

Id	User stories	Id-tâches	Tâches
1	En tant qu'admin, je peux visualiser la liste des utilisateurs (client, agent) ainsi qu'ajouter un utilisateur (ajouter sauf agent)	1.1 1.2 1.3	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Ajouter agent » -Développer « Ajouter agent » -Tester
2	En tant qu'admin, je peux visualiser la liste des utilisateurs (client, agent) et modifier utilisateur	2.1 2.2 2.3	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Modifier utilisateur » -Développer « Modifier utilisateur » -Tester

3	En tant qu'admin, je peux visualiser la liste des utilisateurs (client, agent) et supprimer utilisateur.	3.1 3.2 3.3	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Supprimer utilisateur » -Développer « Supprimer utilisateur » -Tester
----------	--	--	--

Tableau 13: Sprint 2 backlog

2. Spécification fonctionnelle

2.1 Diagramme de cas d'utilisation global de sprint 2

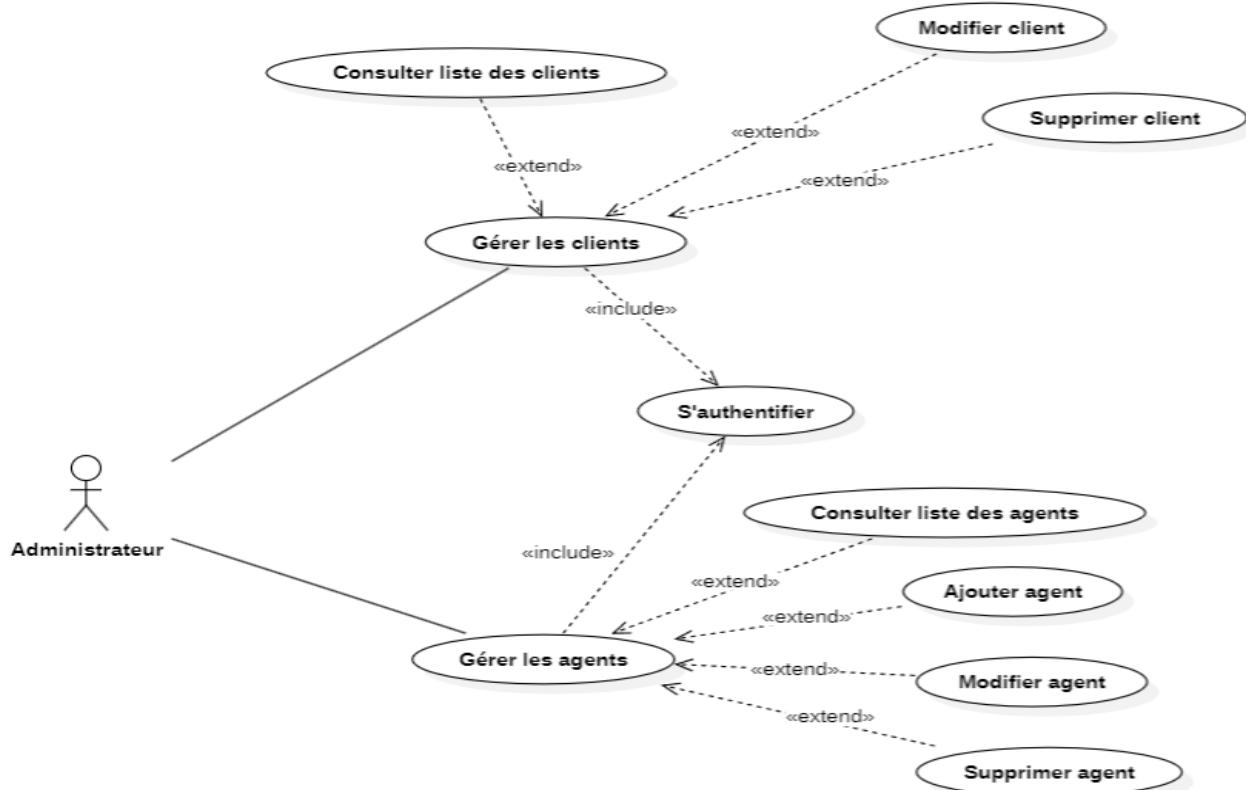


Figure 23: Diagramme de cas d'utilisation global de sprint 2

2.2 Description textuelle des cas d'utilisation

Cas d'utilisation	Ajouter agent
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour ajouter un agent.
Postcondition	Un nouvel agent est ajouté dans le système avec les informations fournies.

Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la page de création d'agent. 2) L'administrateur saisit les informations requises telles que le nom, l'adresse e-mail, etc. 3) L'administrateur valide les informations. 4) Le système vérifie la validité des données et crée un nouvel agent avec les informations fournies. 5) Le système affiche un message de confirmation de la création de l'agent.
Scénario alternatif	<ol style="list-style-type: none"> 1) L'administrateur saisit des informations incorrectes ou manquantes. 2) Le système affiche un message d'erreur indiquant les champs invalides ou manquants. 3) L'administrateur corrige les informations incorrectes ou ajoute les informations manquantes. 4) L'administrateur valide à nouveau les informations.

Tableau 12: Description textuelle du cas d'utilisation "Ajouter utilisateur"

Cas d'utilisation	Modifier utilisateur
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour modifier un utilisateur.
Postcondition	Les informations de l'utilisateur sont mises à jour avec les modifications apportées.
Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la page de modification d'utilisateur. 2) L'administrateur sélectionne l'utilisateur à modifier. 3) L'administrateur apporte les modifications nécessaires telles que le nom, l'adresse e-mail, etc... 4) L'administrateur enregistre les modifications. 5) Le système vérifie la validité des données et met à jour les informations de l'utilisateur.

	<ol style="list-style-type: none"> 6) Le système affiche un message de confirmation de la modification de l'utilisateur.
--	---

Scénario alternatif	<ol style="list-style-type: none"> 1) L'administrateur saisit des informations incorrectes ou manquantes lors de la modification. 2) Le système affiche un message d'erreur indiquant les champs invalides ou manquants. 3) L'administrateur corrige les informations incorrectes ou ajoute les informations manquantes. 4) L'administrateur enregistre à nouveau les modifications.
----------------------------	--

Tableau 13:Description textuelle du cas d'utilisation "Modifier utilisateur"

Cas d'utilisation	Supprimer utilisateur
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour supprimer un utilisateur.
Postcondition	L'utilisateur est supprimé et n'est plus actif dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la page de gestion des utilisateurs. 2) L'administrateur sélectionne l'utilisateur à supprimer. 3) L'administrateur active l'option supprimer pour l'utilisateur. 4) Le système affiche un message de confirmation de suppression de l'utilisateur.
	<ol style="list-style-type: none"> 1) Si l'utilisateur est actuellement engagé dans des transactions actives, le système affiche un message indiquant qu'il ne peut être supprimé. 2) En cas de problème technique pendant la suppression, le système informe l'administrateur de l'échec de l'opération. 3) Si l'administrateur annule la suppression, aucune action n'est prise et l'utilisateur reste actif.

Tableau 14:Description textuelle du cas d'utilisation "Supprimer utilisateur"

2.3 Description comportementale des histoires du sprint 2

- **Diagramme de séquence « Ajouter agent »**

La figure 24 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « ajouter agent ».

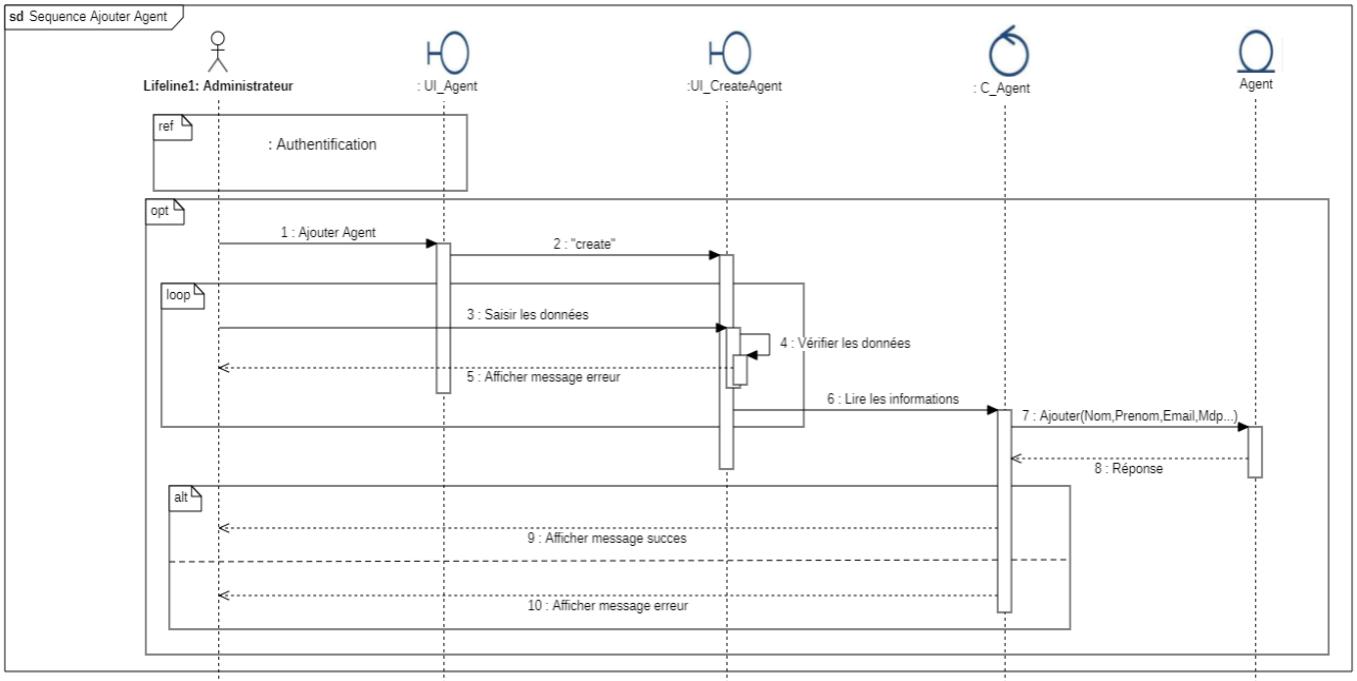


Figure 24: Diagramme de séquence de cas d'utilisation "Ajouter agent"

- **Diagramme de séquence « Modifier utilisateur »**

La figure 25 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « modifier utilisateur ».

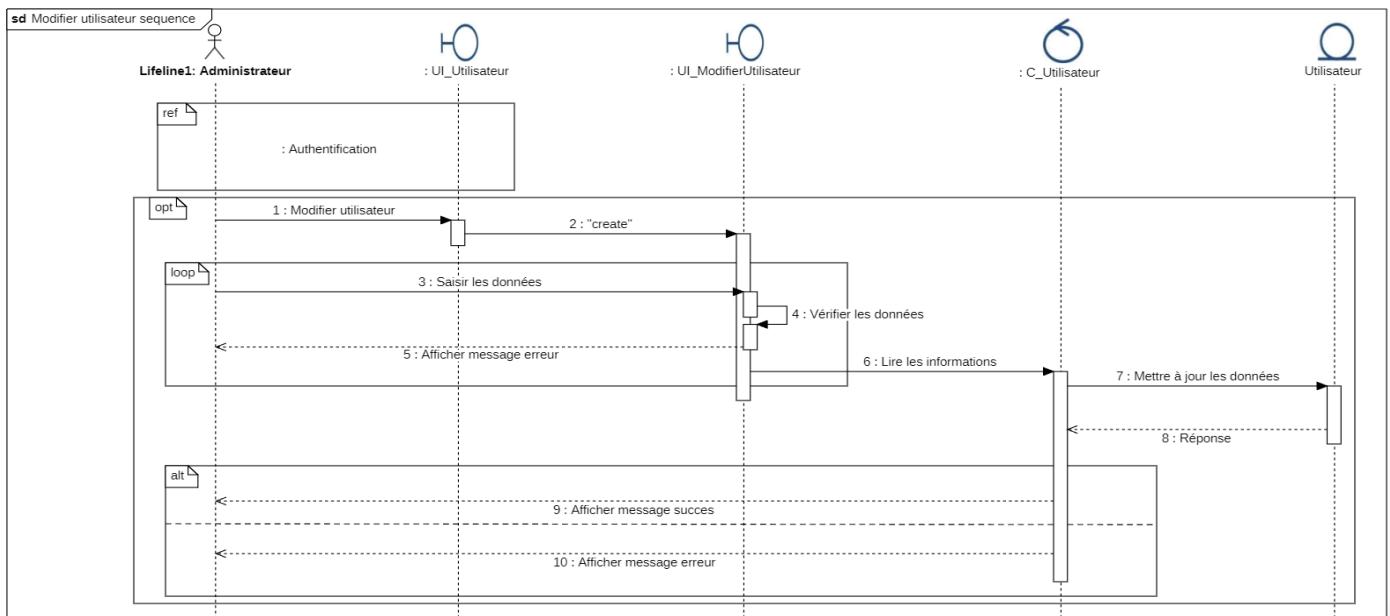


Figure 25: Diagramme de séquence de cas d'utilisation « Modifier utilisateur »

- **Diagramme de séquence « Supprimer utilisateur »**

La figure 26 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Supprimer utilisateur ».

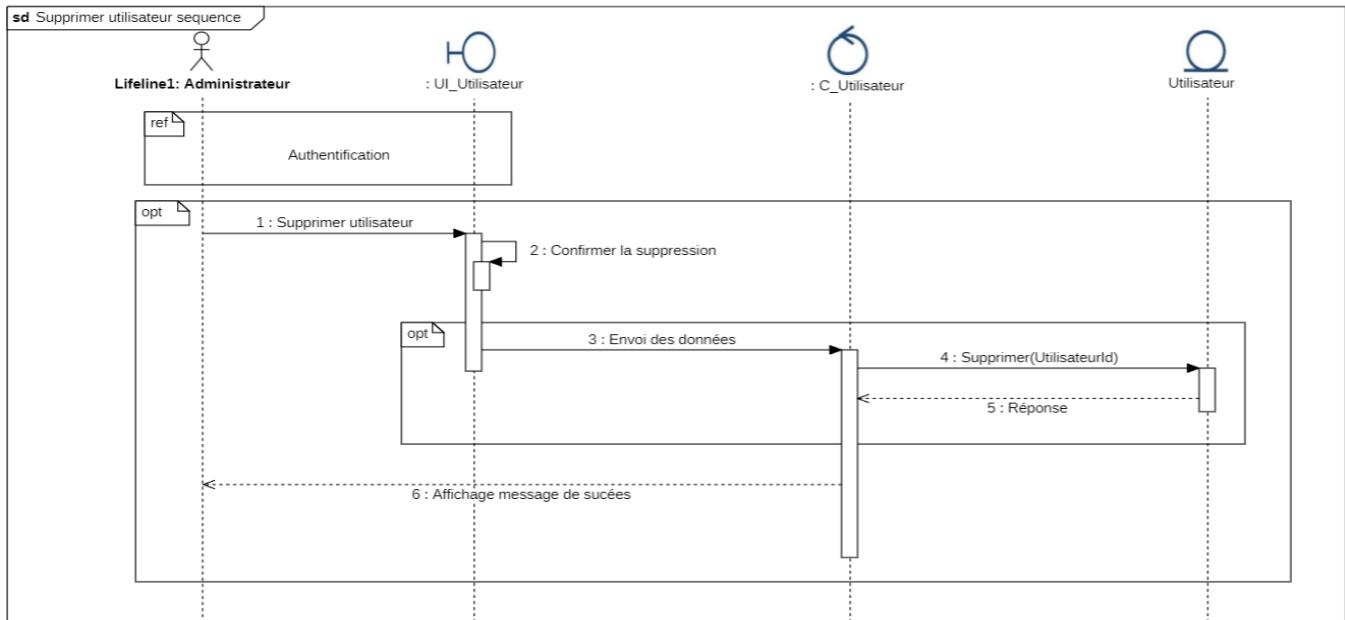


Figure 26: Diagramme de séquence de cas d'utilisation "Supprimer utilisateur"

2.4 Diagramme de classe de sprint 2

Pour le deuxième sprint de notre projet, nous avons établi le même diagramme de classes illustré par la figure 27 mais pas les mêmes fonctionnalités. Il comporte 4 classes.

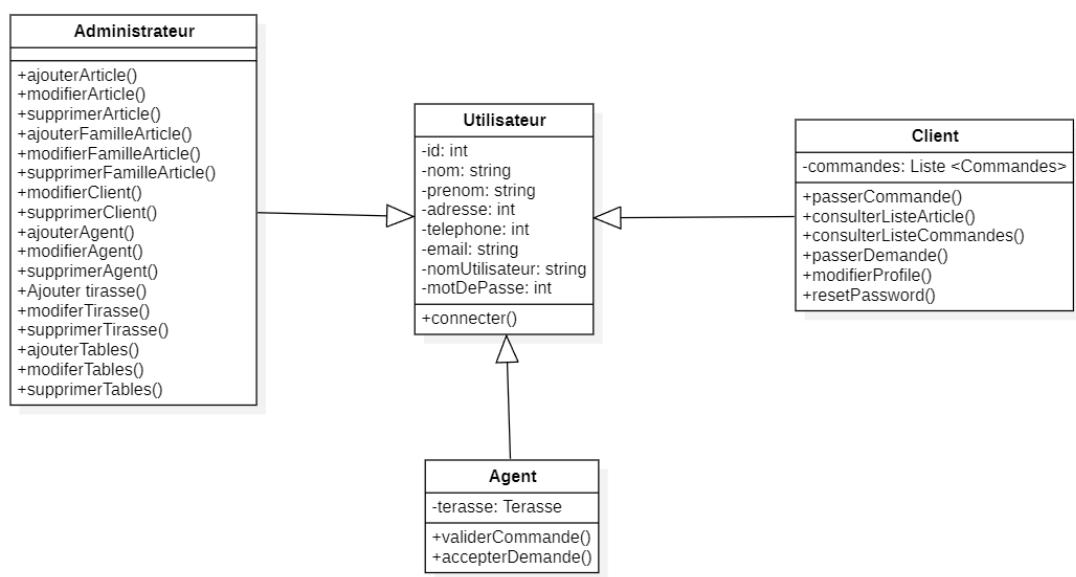


Figure 27: Diagramme de classe de sprint 2

3. Test et interfaçage

3.1. Test sprint 2

Cas	Résultat
Gérer utilisateur (agent, client)	Conforme

Tableau 15: Test du sprint 2

3.2. Interfaçages

Pour illustrer la réalisation du deuxième sprint, nous présenterons les interfaces des principales fonctionnalités offertes par le module gestion d'utilisateurs

- **Interface admin tableau de bord**

The screenshot shows the Admin Dashboard interface. On the left, there is a sidebar menu with the following items:

- Dashboard
- Agents List
- Clients List
- Products List
- Categories List
- Tables List
- Terraces List
- Sign out

The main content area has two main sections:

- Articles**: A section showing "Top Products" with 10 items. The table includes columns: Libelle (Label), Prix Vente (Sale Price), Famille (Family), and Action (Actions). The products listed are: Seafood pizza (30), Margherita pizza (20), Beef burger (15), French fries (5), Crispy chicken burger (25), and Pasta 4 cheese (35).
- User Management**: A section titled "Gérer utilisateur (agent, client)" which is currently "Conforme".

Figure 28: Interface de bord de l'admin

- **Interface gestion des agents**

The screenshot shows a dashboard titled 'Agents'. On the left, a sidebar displays 'Agents' with a count of 2. On the right, a main panel shows 'Top Agents' with a count of 2. Below this, a table lists two agents:

Id	Image	Nom	Prenom	Email	Adresse	Telephone	Tirasse	Action
12		nasr	adem	adem@gmail.com	skhira	90909090	Tirasse 1	
14		gharbi	kamel	kamel@gmail.com	sokra	90909090	Tirasse 2	

Figure 29: Interface de gestion des agents

- **Interface gestion des clients**

The screenshot shows a dashboard titled 'Clients'. On the left, a sidebar displays 'Clients' with a count of 2. On the right, a main panel shows 'Top Clients' with a count of 2. Below this, a table lists two clients:

Id	Image	Nom	Prenom	Email	Adresse	Telephone	Action
17		Zemzem	Amen	ameezmm22@gmail.com	Boumhal	96126000	
20		Faydi	Wael	waelwafaydi@gmail.com	Sidi Hsin	90909090	

Figure 30: Interface gestion des clients

Conclusion

Dans le chapitre consacré au sprint 2, nous avons mis l'accent sur la gestion des utilisateurs. Nous avons réalisé ces fonctionnalités en suivant les descriptions fonctionnelles et comportementales des histoires, en effectuant des tests approfondis pour assurer la fiabilité et la sécurité du système. Ce sprint a renforcé les bases de notre application et a permis une gestion efficace des utilisateurs.

Chapitre 5 : Sprint 3 - Module gestion des articles et familles articles

Introduction

Dans ce chapitre, nous aborderons le troisième sprint de notre projet, qui se concentre sur la gestion des articles et familles articles. Nous explorerons les fonctionnalités liées à la création, la modification et la gestion de ces éléments essentiels pour notre système.

1. Backlog du sprint 3

1.1. Affectation des cas d'utilisations aux acteurs

Le tableau ci-dessous présente l'affectation de chaque cas d'utilisation à son acteur pour notre troisième sprint :

Acteurs	Cas d'utilisation
Admin	Gérer les articles
	Gérer les familles articles
Client	Ajouter un article au commande
	Visualiser les articles

Tableau 16:Affectation des cas d'utilisations aux acteurs du sprint 3

1.2. Sprint backlog

Id	User stories	Id-tâches	Tâches
1	En tant qu'administrateur, je peux consulter la liste des familles articles existants ainsi qu'ajouter, modifier et supprimer une famille articles	1.1 1.2 1.3	-Etablir le prototype, le diagramme de cas d'utilisation et les diagrammes de séquence « Ajouter famille articles », « Modifier famille articles » et « Supprimer famille articles », -Développer « Gérer famille articles » -Tester

2	En tant qu'administrateur, je peux consulter la liste des articles existante ainsi qu'ajouter, modifier et supprimer un article	2.1	-Etablir le prototype, le diagramme de cas d'utilisation et les diagrammes de séquence « Ajouter article », « Modifier article » et « Supprimer article »
		2.2	-Développer « Gérer article »
		2.3	-Tester
3	En tant qu'un client, je peux ajouter un article à une commande	3.1	-Développer l'interface utilisateur permettant à un client d'ajouter un article à une commande.
4	En tant que client, je peux visualiser les familles articles et les articles	4.1	-Développer l'interface utilisateur permettant la visualisation des familles d'articles et des articles.

Tableau 17: Sprint 3 backlog

2. Spécification fonctionnelle

2.1 Diagramme de cas d'utilisation global de sprint 3

On passe au raffinement des diagrammes par acteur.

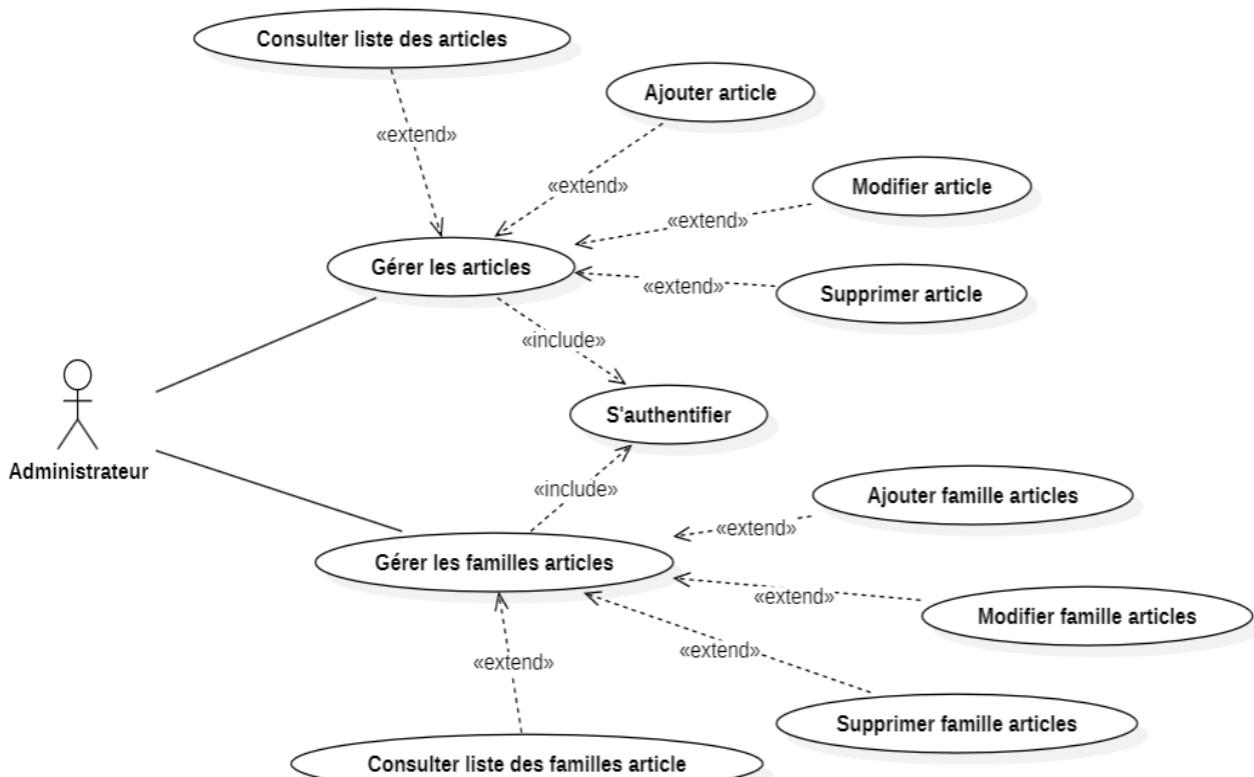


Figure 31: Diagramme de cas d'utilisation global de sprint 3 de l'administrateur

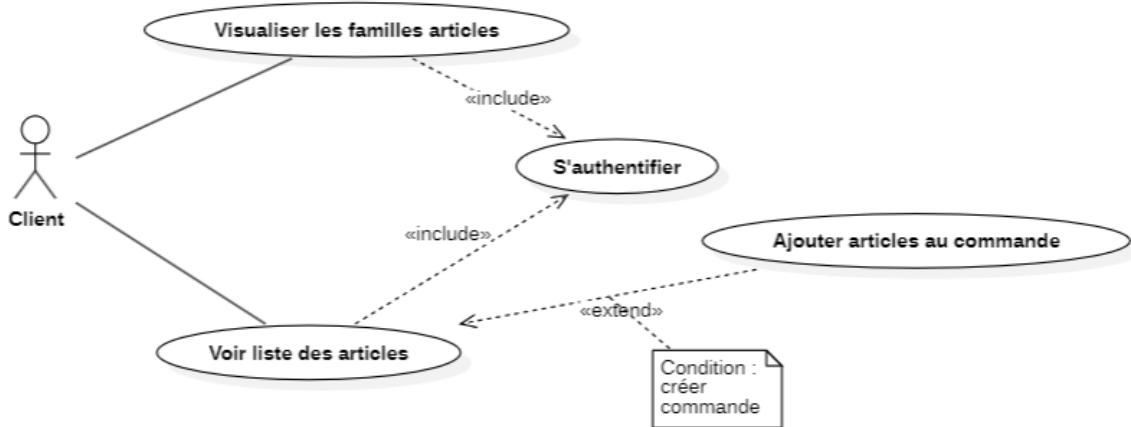


Figure 32: Diagramme de cas d'utilisation global de sprint 3 de client

2.2 Description textuelle des cas d'utilisation

Cas d'utilisation	Gérer famille articles
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour gérer les familles articles.
Postcondition	Les modifications apportées aux famille articles sont enregistrées dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la fonctionnalité de gestion des familles articles 2) Le système affiche la liste des familles articles existants. 3) L'administrateur peut choisir d'ajouter une nouvelle famille article en fournissant les informations nécessaires telles que le titre ... 4) Le système enregistre la nouvelle famille article dans la base de données. 5) L'administrateur peut sélectionner une famille article existant pour le modifier en mettant à jour les informations. 6) Le système enregistre les modifications apportées au famille article. 7) L'administrateur a la possibilité de supprimer une famille article existant en le sélectionnant et en confirmant la suppression. 8) Le système supprime la famille article de la base de données.

Scénario alternatif	1) Si aucune famille d'articles n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'une famille d'articles, le système affiche un message d'erreur et l'administrateur est informé.
----------------------------	---

Tableau 18:Description textuelle du cas d'utilisation "Gérer famille articles"

Cas d'utilisation	Gérer articles
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour gérer les articles.
Postcondition	Les modifications apportées aux articles sont enregistrées dans le système.
Scénario nominal	1) L'administrateur accède à la fonctionnalité de gestion des articles. 2) Le système affiche la liste des articles existante. 3 L'administrateur peut choisir d'ajouter un nouvel article en fournissant les informations nécessaires telles que le nom, la description, le prix et son famille articles. 4) Le système enregistre le nouvel article dans la base de données. 5) L'administrateur peut sélectionner un article existant pour le modifier en mettant à jour les informations. 6) Le système enregistre les modifications apportées à l'article. 7) L'administrateur a la possibilité de supprimer un article existante en le sélectionnant et en confirmant la suppression. 8) Le système supprime l'article de la base de données.
Scénario alternatif	1) Si aucun article n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'un article, le système affiche un message d'erreur et l'administrateur est informé.

Tableau 19: Description textuelle du cas d'utilisation " Gérer articles "

2.3 Description comportementale des histoires du sprint 3

- **Diagramme de séquence « Ajouter famille articles »**

La figure 33 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Ajouter famille articles ».

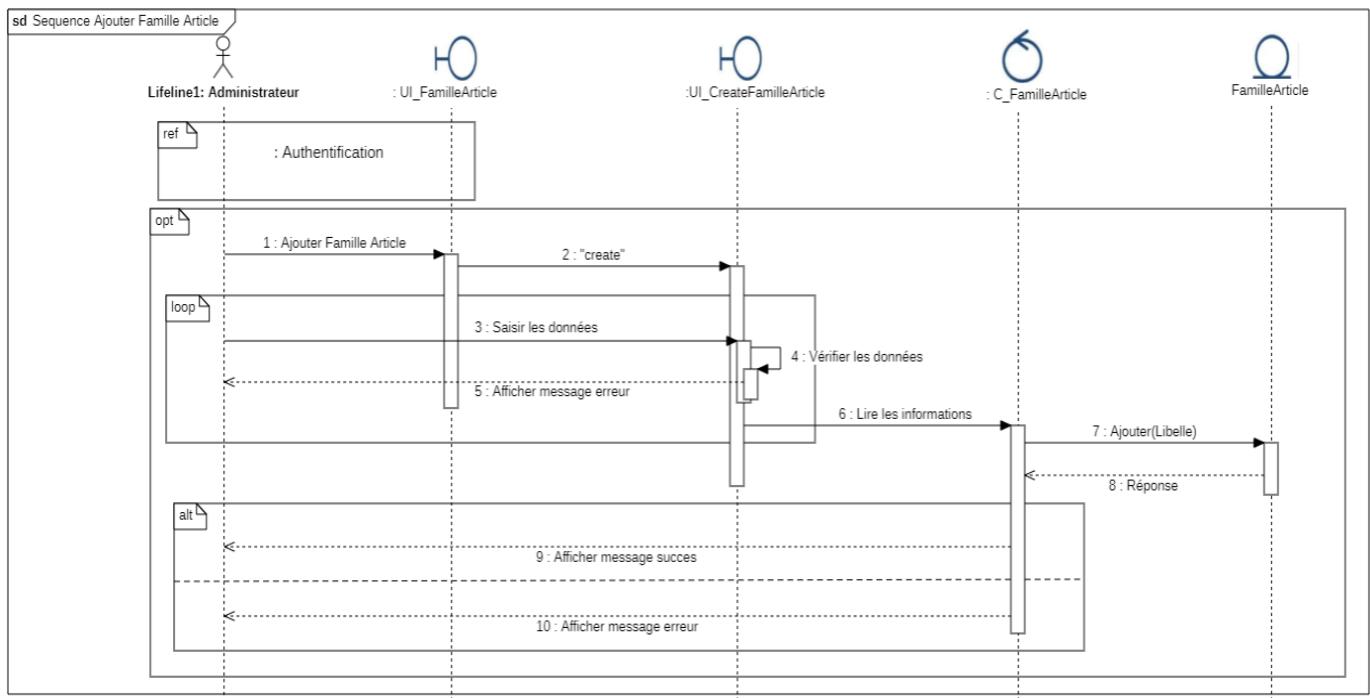


Figure 33: Diagramme de séquence de cas d'utilisation "Ajouter famille articles"

- Diagramme de séquence « Modifier famille articles »**

La figure 34 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Modifier famille articles ».

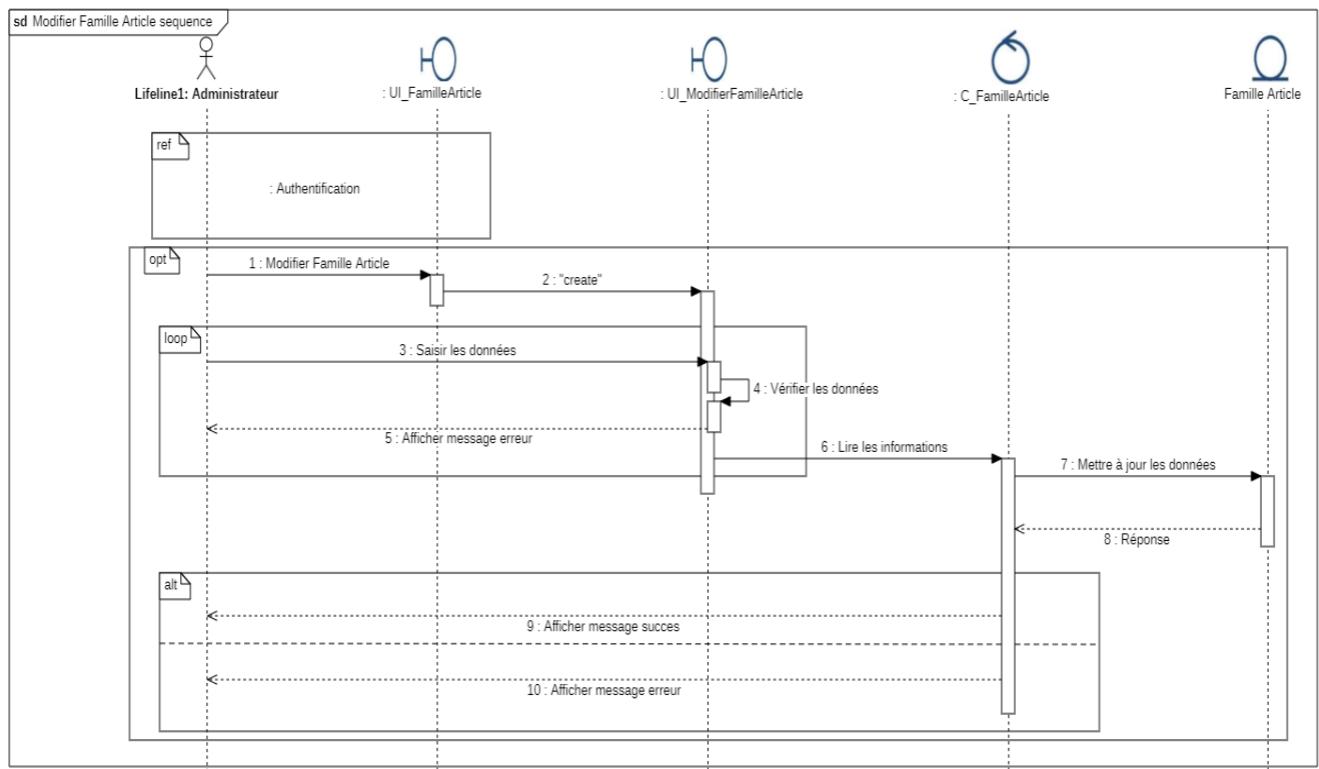


Figure 34: Diagramme de séquence de cas d'utilisation "Modifier famille articles"

- **Diagramme de séquence « Supprimer famille articles »**

La figure 35 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Supprimer famille articles ».

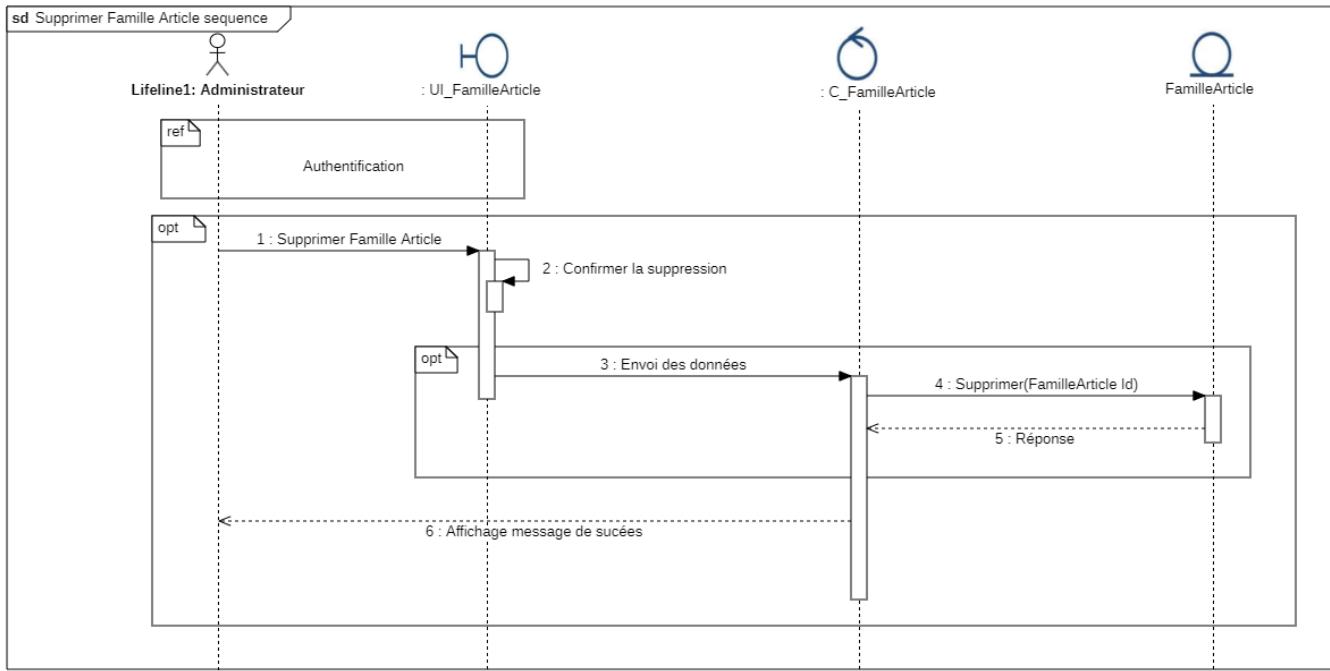


Figure 35: Diagramme de séquence de cas d'utilisation "Supprimer famille articles"

- **Diagramme de séquence « Ajouter articles »**

La figure 36 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Ajouter articles ».

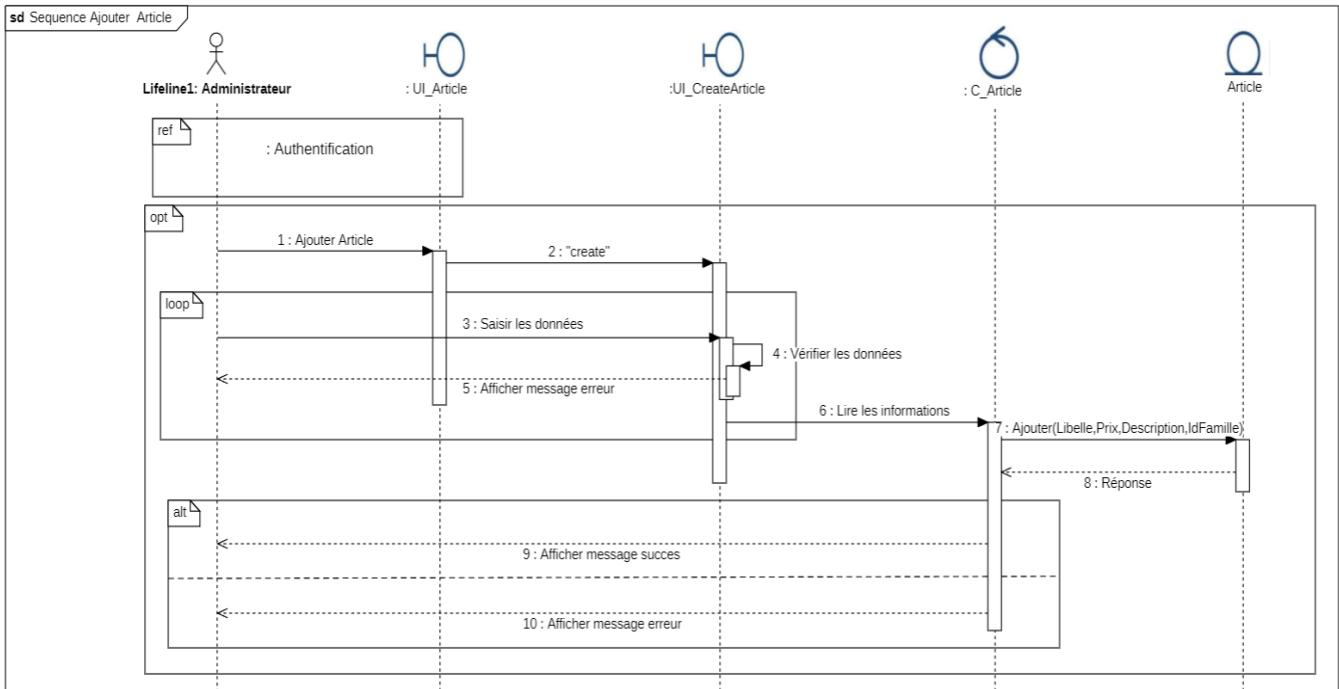


Figure 36: Diagramme de séquence de cas d'utilisation "Ajouter articles"

- **Diagramme de séquence « Modifier articles »**

La figure 37 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Modifier articles ».

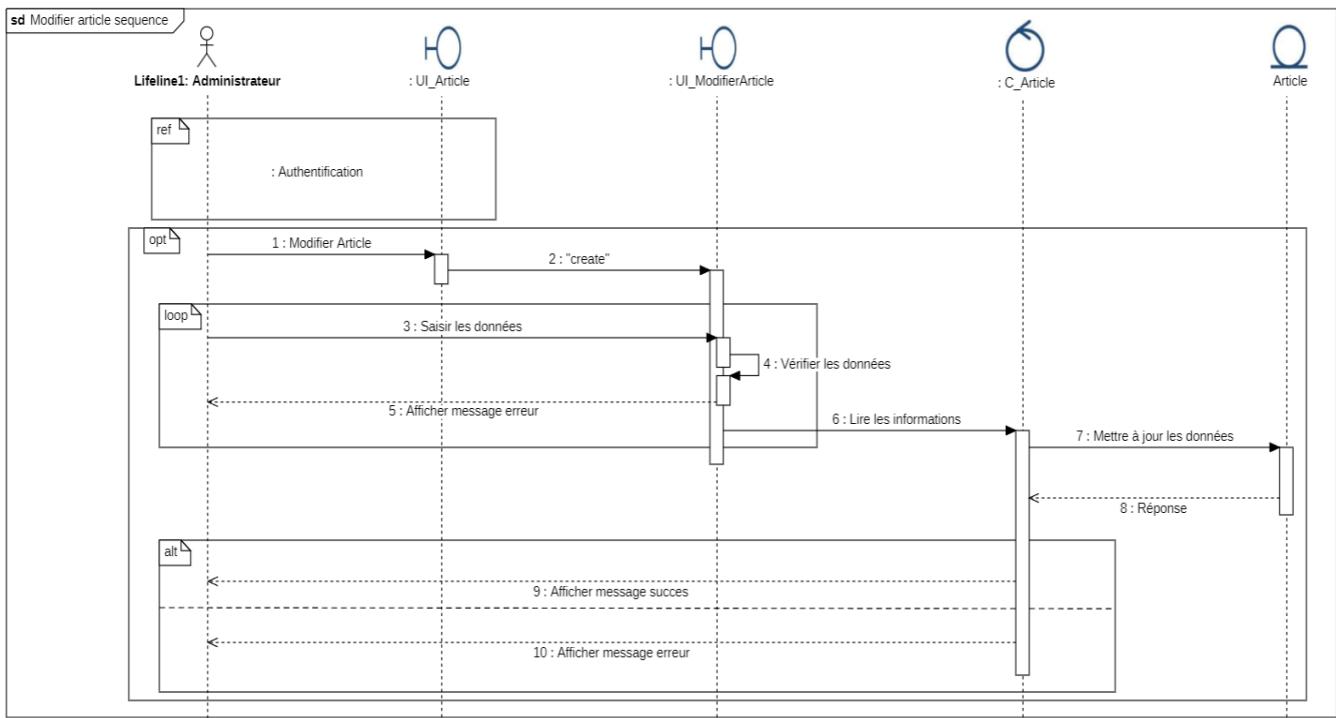


Figure 37: Diagramme de séquence de cas d'utilisation "Modifier articles"

- **Diagramme de séquence « Supprimer articles »**

La figure 38 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Supprimer articles ».

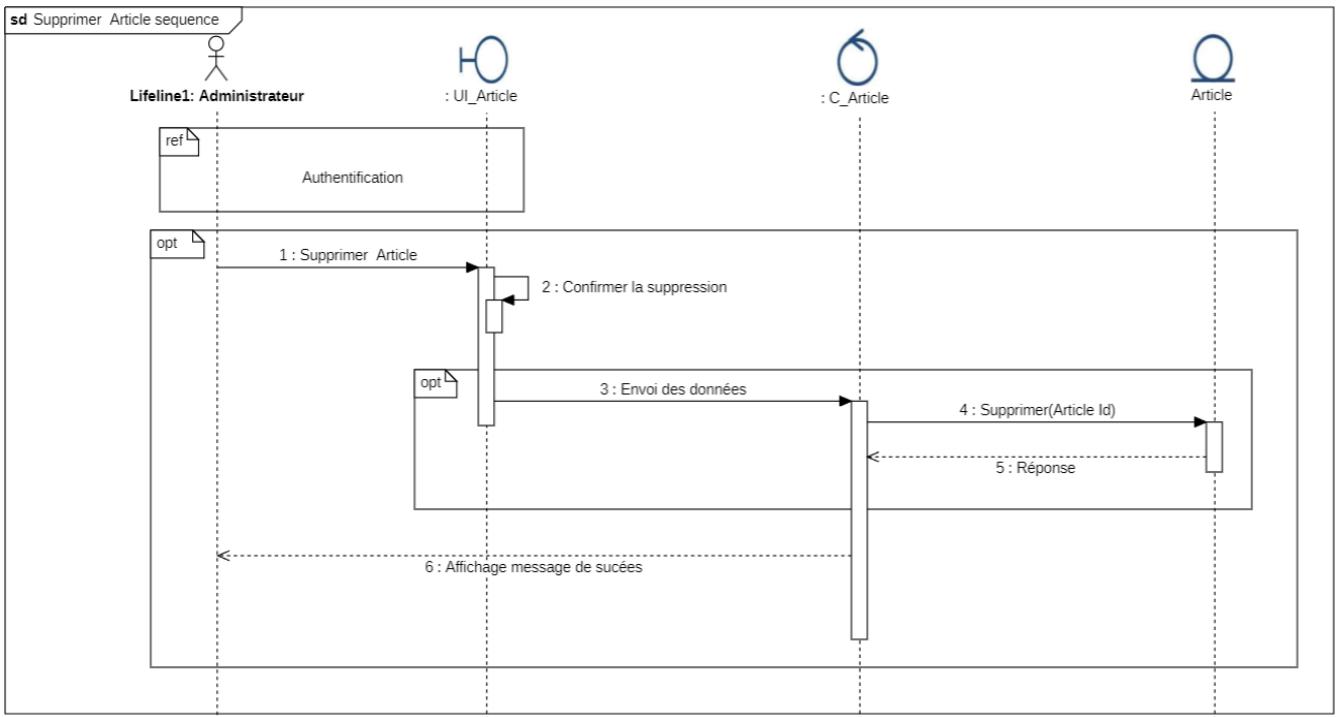


Figure 38: Diagramme de séquence de cas d'utilisation "Supprimer articles"

2.4 Diagramme de classe de sprint 3

La figure illustre le diagramme de classes pour le troisième sprint de notre projet. Il comporte 6 classes.

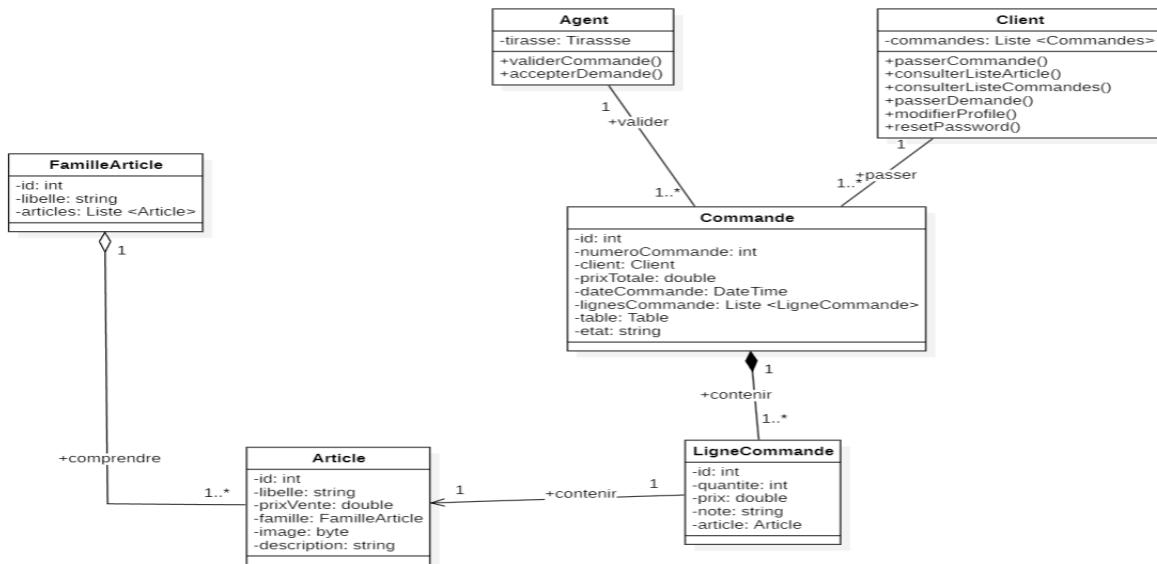


Figure 39: Diagramme de classe sprint 3

3. Test et interfaçage

3.1. Test sprint 3

Cas	Résultat
Gestion des familles articles	
Gestion des articles	Conforme

Tableau 20: Test du sprint 3

3.2. Interfaçage

- Interface de gérer des familles articles(catégorie)

The screenshot shows a web-based application interface for managing article families. The top navigation bar includes links for 'Categories', 'Clients', 'Commandes', and 'Utilisateurs'. The main content area is titled 'Categories' and displays a table of categories with the following data:

ID	Image	Libelle	Action
10		pasta	
11		Burger	
12		Pizza	
13		Fries	
17		Desert	
19		Gazouz	

Figure 40: Interface de gérer des familles articles

- Interface de gérer des articles

The screenshot shows a web-based application interface for managing articles. At the top, there are two cards: 'Products' (10 items) and 'Top Products' (10 items). Below them is a search bar with the placeholder 'Search'. A green header bar contains the title 'Articles' and a '+' button. The main area is a table listing products with columns: Id, Image, Libelle, Prix Vente, Famille, and Action. The table rows represent the following data:

Id	Image	Libelle	Prix Vente	Famille	Action
5		Seafood pizza	30	Pizza	trash edit
6		Margherita pizza	20	Pizza	trash edit
7		Beef burger	15	Burger	trash edit
8		French fries	5	Fries	trash edit
9		Crispy chicken burger	25	Burger	trash edit
10		Pasta 4 cheese	35	pasta	trash edit

Figure 41: Interface de gérer les articles

- Interface de modification d'un article

The screenshot shows a modal dialog titled 'Edit Article' overlying the main 'Articles' interface. The modal contains fields for 'Libelle' (Seafood pizza), 'Image' (placeholder image), and 'Description' (Seafood pizza features a crispy crust, tomato or garlic sauce, and to...). It also includes a file upload field ('Upload Photo') with the placeholder 'Choisir un fichier' and a dropdown menu ('Famille') set to 'Pizza'. At the bottom of the modal is a green 'UPDATE ARTICLE' button.

Figure 42: Interface de modification d'un article

- **Interface de liste des commandes de client**

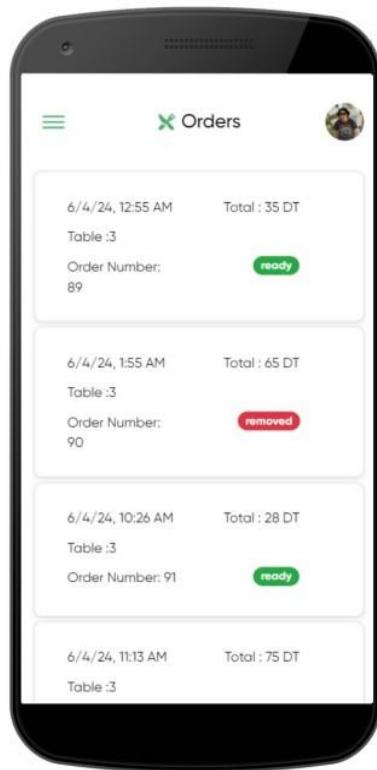


Figure 43: Interface de liste des commandes de client

Conclusion

Dans le chapitre consacré au sprint 3, nous avons focalisé notre attention sur la gestion des articles. Ce sprint a permis de mettre en place une gestion efficace des articles, offrant ainsi une expérience améliorée pour les utilisateurs.

Chapitre 6 : Sprint 4 - Module gestion des terrasses et tables

Introduction

Dans ce chapitre, nous explorons le quatrième sprint de notre projet, centré sur la gestion efficace des terrasses et des tables. Nous aborderons les détails des fonctionnalités développées pour la création, la modification, et la gestion de ces éléments clés, en mettant en lumière les progrès techniques et les ajustements apportés pour répondre aux exigences du projet.

1. Backlog du sprint 4

1.1. Affectation des cas d'utilisations aux acteurs

Le tableau ci-dessous présente l'affectation de chaque cas d'utilisation à son acteur pour notre troisième sprint :

Acteurs	Cas d'utilisation
Admin	Gérer les terrasses
	Gérer les tables

Tableau 21: Affectation des cas d'utilisations aux acteurs du sprint 4

1.2. Sprint backlog

Id	User stories	Id-tâches	Tâches
1	En tant qu'administrateur, je peux consulter la liste de terrasses existantes ainsi qu'ajouter, modifier et supprimer une terrasse.	1.1 1.2 1.3	-Etablir le prototype, le diagramme de cas d'utilisation et les diagrammes de séquence « Ajouter terrasses », « Modifier terrasse » et « Supprimer terrasse », -Développer « Gérer terrasses » -Tester
2	En tant qu'administrateur, je peux consulter la liste des tables existante ainsi qu'ajouter, modifier et supprimer une table.	2.1 2.2 2.3	-Etablir le prototype, le diagramme de cas d'utilisation et les diagrammes de séquence « Ajouter tables », « Modifier table » et « Supprimer tables » -Développer « Gérer tables » -Tester

Tableau 22: Sprint 4 backlog

2. Spécification fonctionnelle

2.1 Diagramme de cas d'utilisation global de sprint 4

On passe au raffinement des diagrammes par acteur.

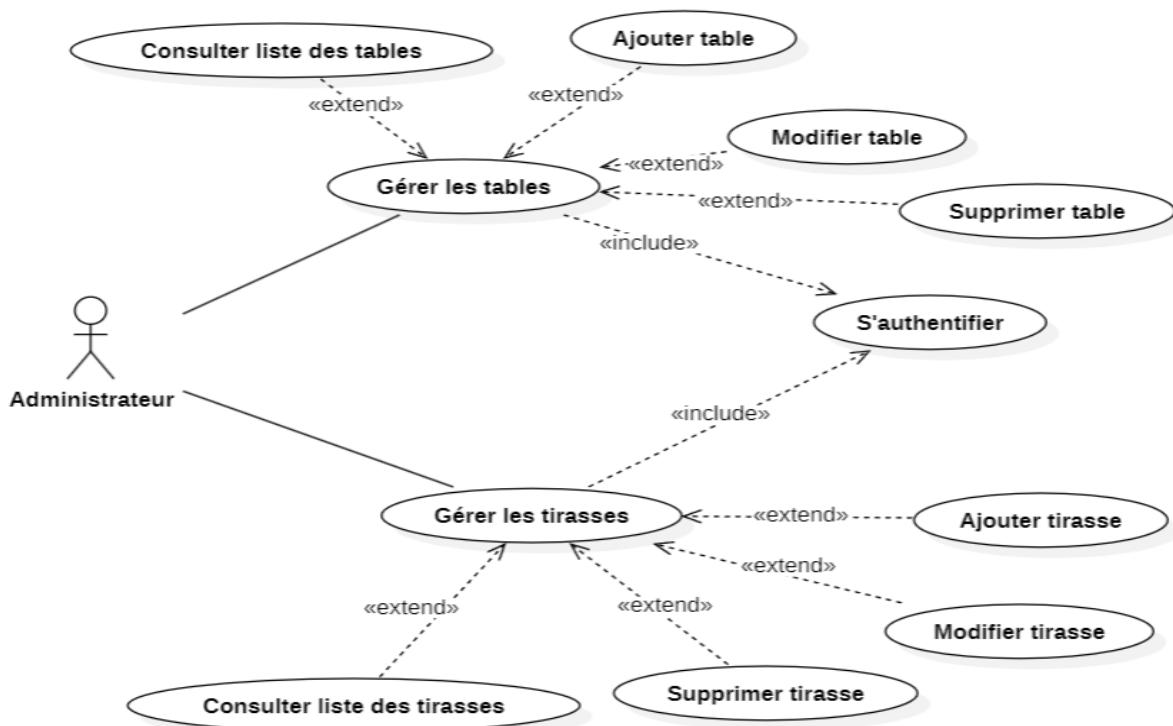


Figure 44:Diagramme de cas d'utilisation global de sprint 4 de l'administrateur

2.2 Description textuelle des cas d'utilisation

Cas d'utilisation	Gérer terrasses
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour gérer les terrasses.
Postcondition	Les modifications apportées aux terrasses sont enregistrées dans le système.

Scénario nominal	<p>1) L'administrateur accède à la fonctionnalité de gestion des terrasses.</p> <p>2) Le système affiche la liste de terrasses existantes.</p> <p>3) L'administrateur peut choisir d'ajouter une nouvelle terrasse en fournissant les informations nécessaires telles que le numéro ...</p> <p>4) Le système enregistre la nouvelle terrasse dans la base de données.</p> <p>5) L'administrateur peut sélectionner une terrasse existante pour la modifier en mettant à jour les informations.</p> <p>6) Le système enregistre les modifications apportées à la terrasse.</p> <p>7) L'administrateur a la possibilité de supprimer une terrasse existante en la sélectionnant et en confirmant la suppression.</p> <p>8) Le système supprime la terrasse de la base de données.</p>
-------------------------	---

Scénario alternatif	<p>1) Si aucun terrasse n'est disponible, le système affiche un message indiquant aucune donnée disponible.</p> <p>2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'un terrasse, le système affiche un message d'erreur et l'administrateur est informé.</p>
----------------------------	--

Tableau 23: Description textuelle du cas d'utilisation "Gérer terrasse"

Cas d'utilisation	Gérer table
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié et avoir les droits nécessaires pour gérer les terrasses.
Postcondition	Les modifications apportées aux articles sont enregistrées dans le système.

Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la fonctionnalité de gestion des tables. 2) Le système affiche la liste des tables existante. 3 L'administrateur peut choisir d'ajouter une nouvelle table en fournissant les informations nécessaires telles que le numéro et sa terrasse. 4) Le système enregistre la nouvelle table dans la base de données. 5) L'administrateur peut sélectionner une table existante pour le modifier en mettant à jour les informations. 6) Le système enregistre les modifications apportées à la table. 7) L'administrateur a la possibilité de supprimer une table existante en le sélectionnant et en confirmant la suppression. 8) Le système supprime la table de la base de données.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucune table n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'une table, le système affiche un message d'erreur et l'administrateur est informé.

Tableau 24:Description textuelle du cas d'utilisation "Gérer tables"

2.3 Description comportementale des histoires du sprint 4

- **Diagramme de séquence « Ajouter terrasse »**

La figure 45 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Ajouter terrasse ».

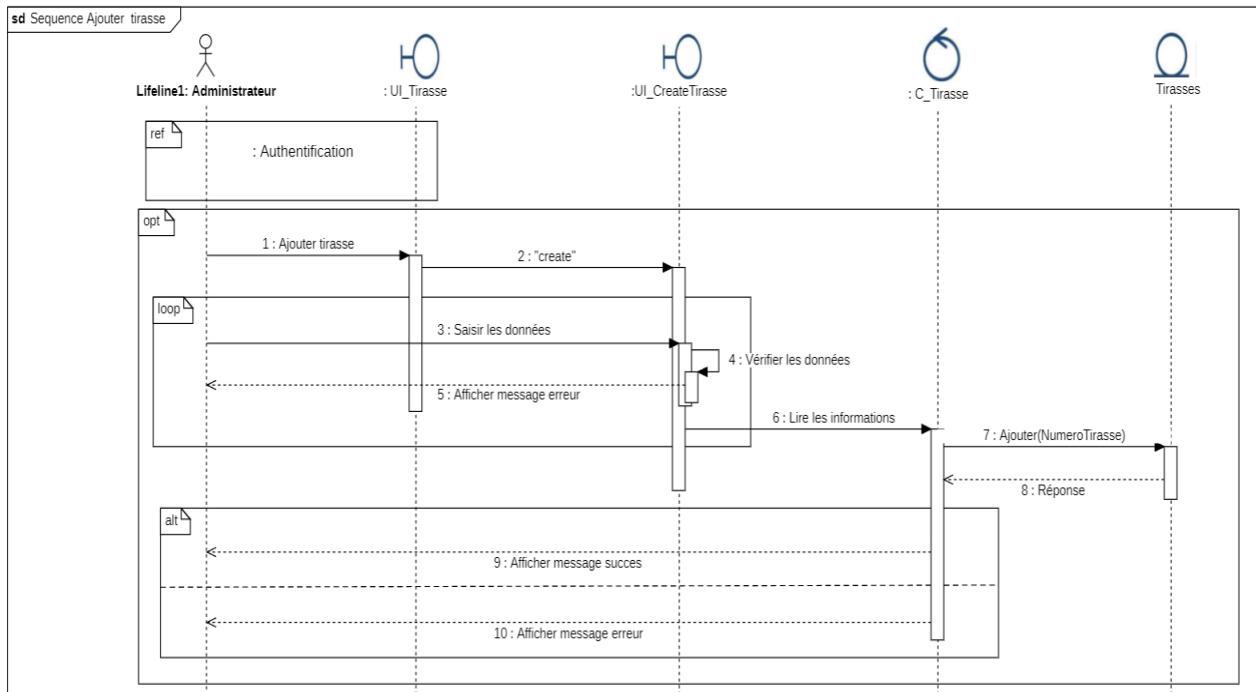


Figure 45:Diagramme de séquence de cas d'utilisation "Ajouter terrasse"

- **Diagramme de séquence « Modifier terrasse »**

La figure 46 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Modifier terrasse ».

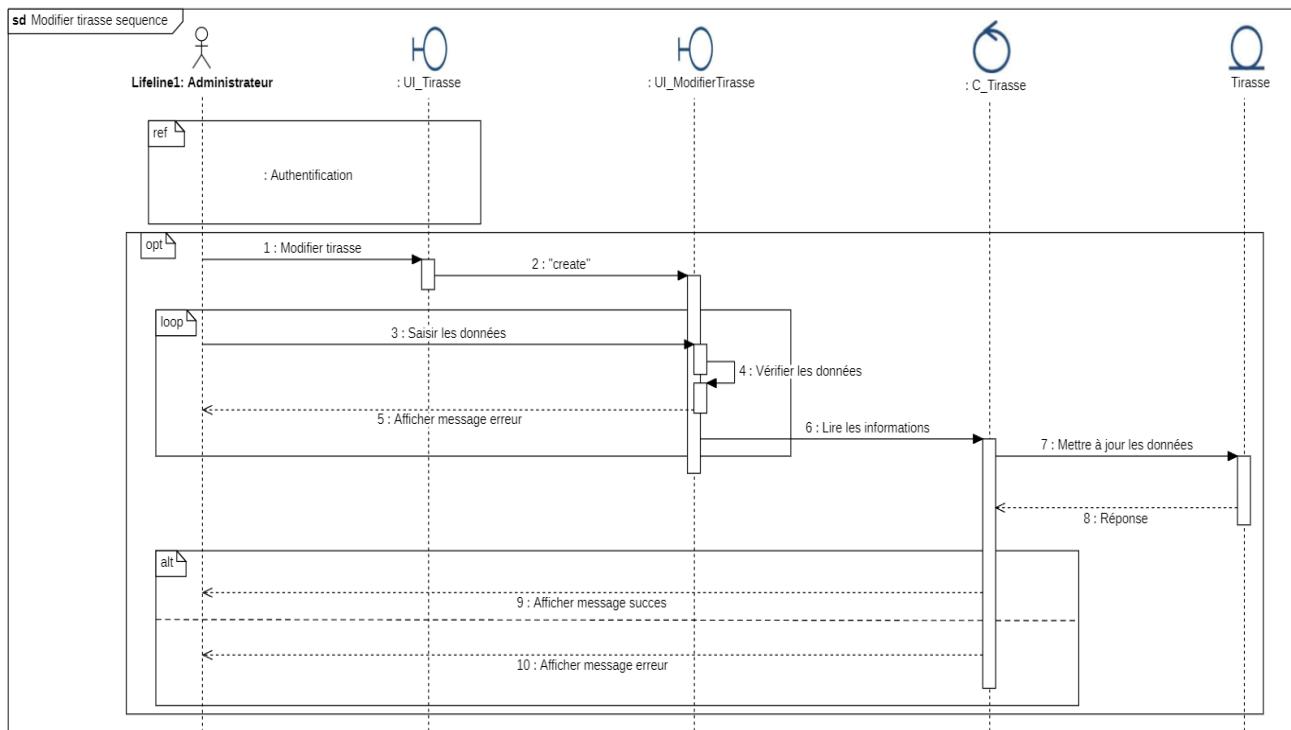


Figure 46:Diagramme de séquence de cas d'utilisation "Modifier terrasse"

- **Diagramme de séquence « Supprimer terrasse »**

La figure 47 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Supprimer terrasse ».

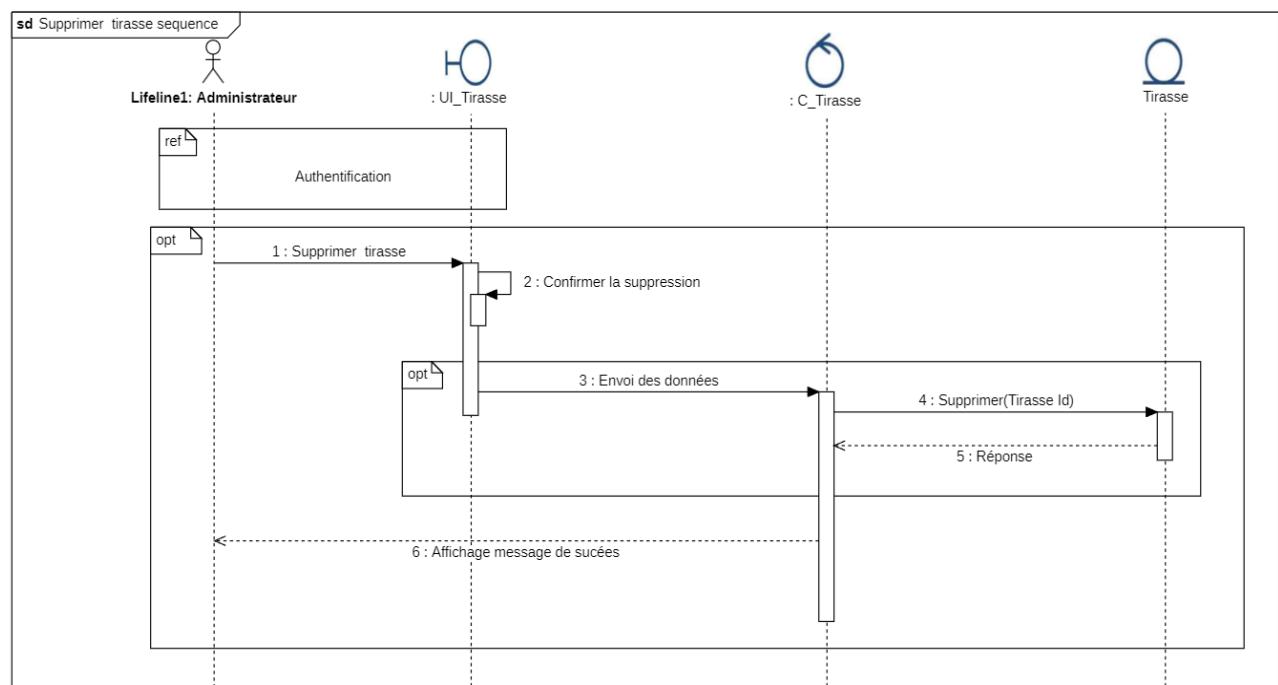


Figure 47:Diagramme de séquence de cas d'utilisation "Supprimer terrasse"

- **Diagramme de séquence « Ajouter table »**

La figure 48 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Ajouter tables ».

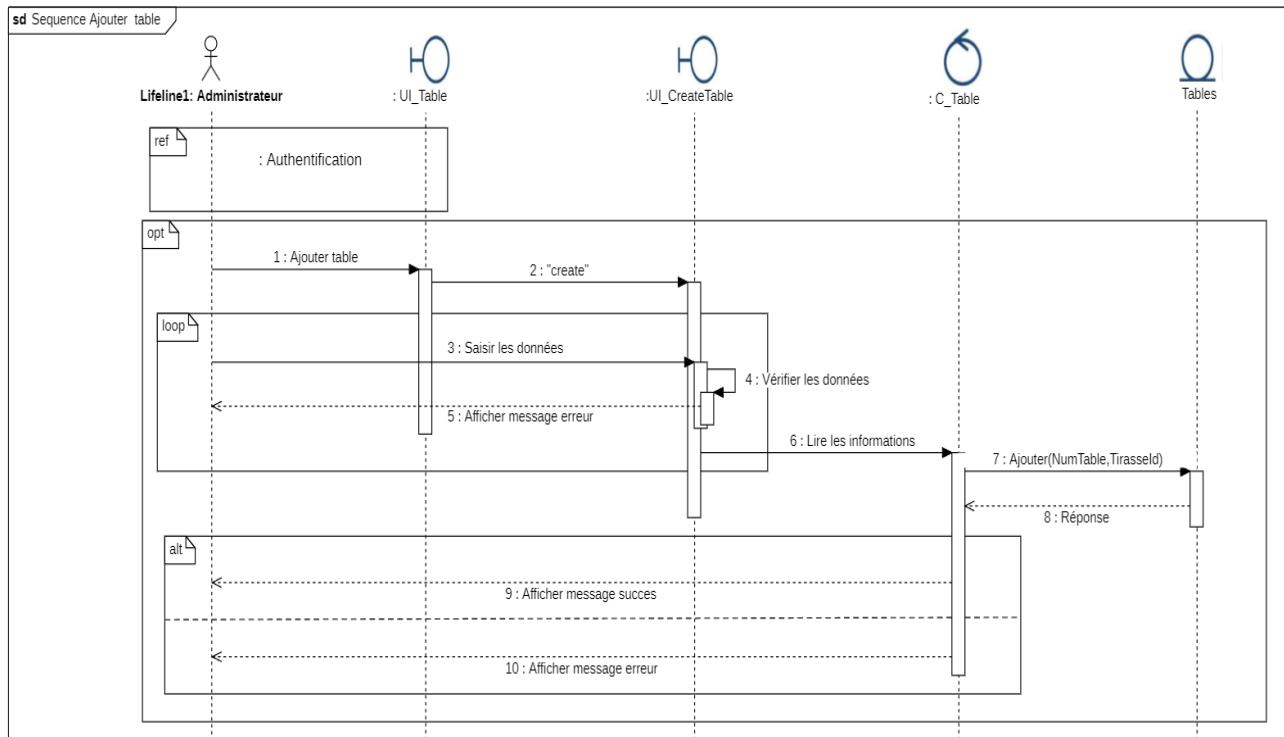


Figure 48: Diagramme de séquence de cas d'utilisation "Ajouter tables"

- **Diagramme de séquence « Modifier table »**

La figure 49 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « Modifier tables ».

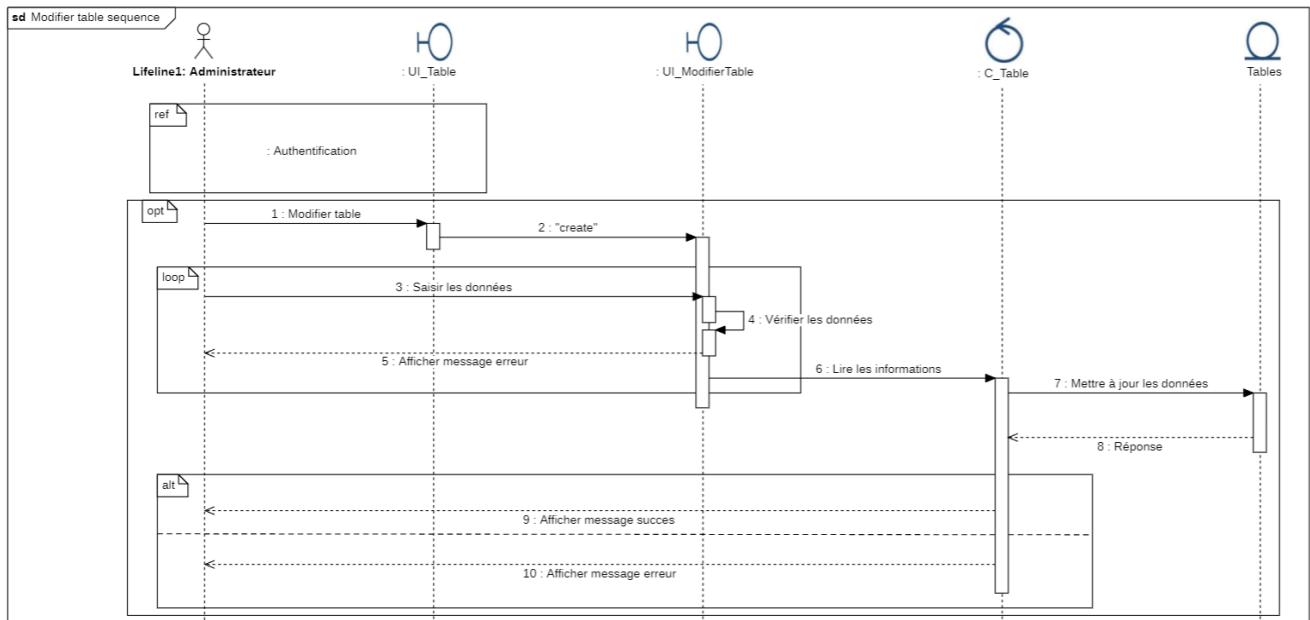


Figure 49: Diagramme de séquence de cas d'utilisation "Modifier tables"

- **Diagramme de séquence « Supprimer table »**

La figure 50 illustre l'enchaînement des séquences nécessaires à la réalisation du cas d'utilisation « Supprimer tables ».

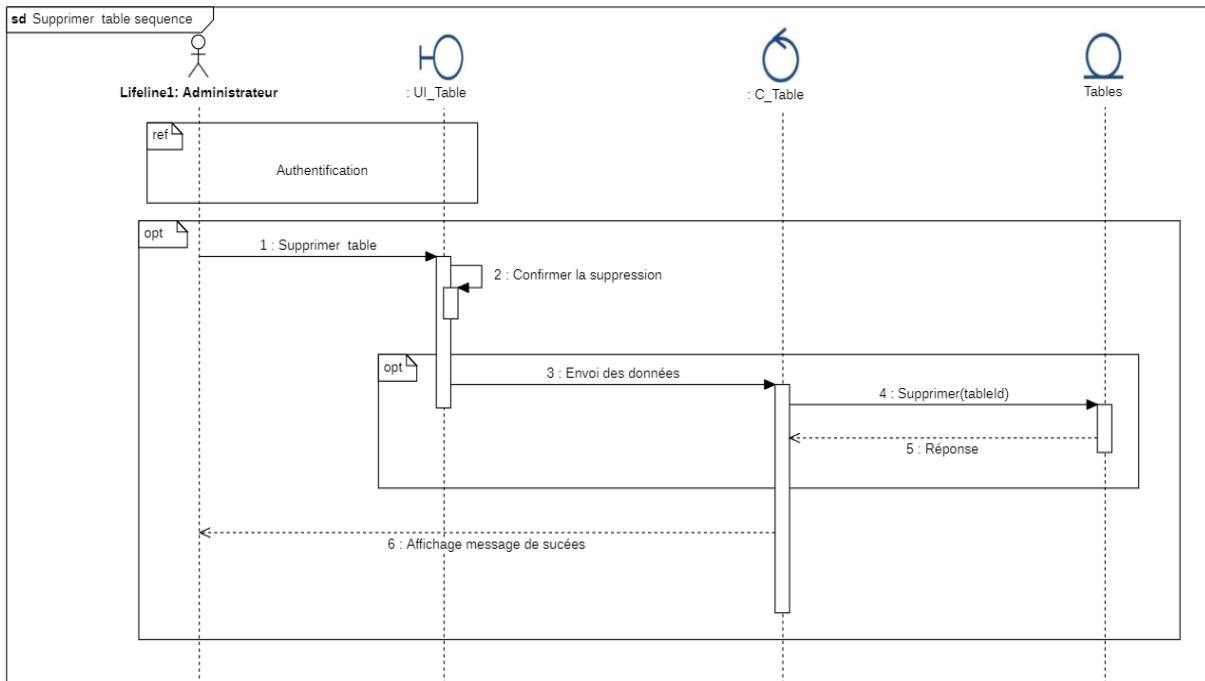


Figure 50:Diagramme de séquence de cas d'utilisation "Supprimer tables"

2.4 Diagramme de classe de sprint 4

La figure illustre le diagramme de classes pour le troisième sprint de notre projet. Il comporte 6 classes.

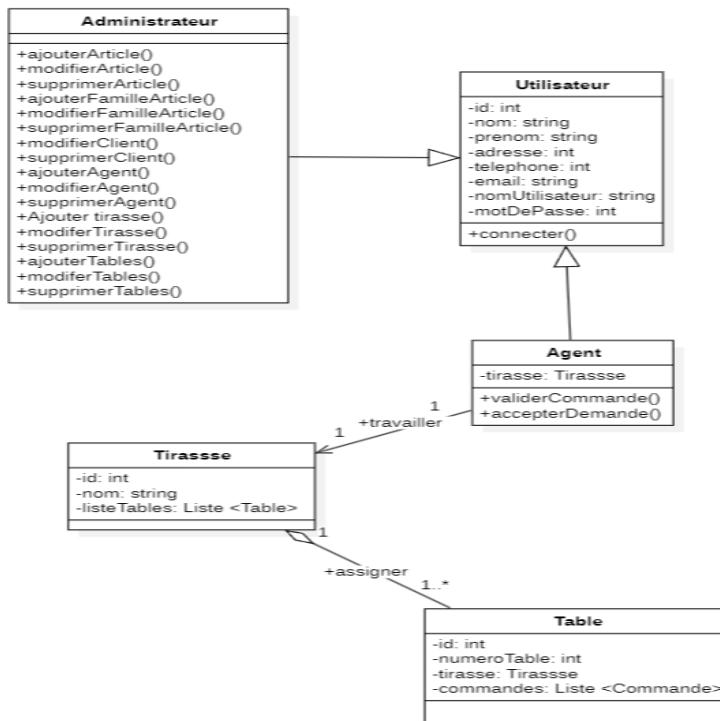


Figure 51: Diagramme de classe de sprint 4

3. Test et interfaçage

3.1. Test sprint 4

Cas	Résultat
Gestion les terrasses	
Gestion les tables	Conforme

Tableau 25:Test du sprint 4

3.2. Interfaçage

- Interface gestion des tables

The screenshot shows a web-based application interface for managing tables. At the top, there's a header with a menu icon, a 'Tables' icon, and a user profile icon. Below the header, a sidebar on the left shows a list titled 'Tables' with a count of 3. The main area contains a table with the following data:

Id	Numéro	Terrasse	Action
4	1	Terasse 1	Delete Edit
5	2	Terasse 1	Delete Edit
6	3	Terasse 2	Delete Edit

Figure 52: Interface gestion des tables

- Interface gestion des terrasses

The screenshot shows a web-based application interface for managing terraces. At the top, there's a header with a menu icon, a 'Terraces' icon, and a user profile icon. Below the header, a sidebar on the left shows a list titled 'Terraces' with a count of 3. The main area contains a table with the following data:

Id	Name	Agent	Action
6	Terasse 1	adem	Delete Edit
7	Terasse 2	kamel	Delete Edit
8	Terasse 3		Delete Edit

Figure 53: Interface de gestion des terrasses

Conclusion

Le quatrième sprint a été crucial pour renforcer les capacités de notre système grâce à l'introduction de fonctionnalités avancées de gestion des terrasses et des tables. Les résultats des tests ont validé l'efficacité de ces fonctionnalités, marquant une étape importante vers l'amélioration continue de notre plateforme. Ce sprint a non seulement amélioré notre système, mais a aussi posé les bases nécessaires pour les développements futurs.

Chapitre 7 : Sprint 5 - Module gestion des commandes et demandes

Introduction

Dans ce chapitre, nous explorons le cinquième sprint de notre projet, désormais centré sur la gestion efficace des commandes et des demandes. Nous détaillerons les fonctionnalités développées pour créer, modifier et gérer les commandes, en soulignant les progrès techniques et les ajustements nécessaires pour répondre aux exigences de notre système.

1. Backlog du sprint 5

1.1. Affectation des cas d'utilisations aux acteurs

Le tableau ci-dessous présente l'affectation de chaque cas d'utilisation à son acteur pour notre troisième sprint :

Acteurs	Cas d'utilisation
Agent	Gérer les commandes. Traiter les demandes des clients.
Client	Passer des commandes. Poster des demandes. Consulter l'historique des commandes.

Tableau 26:Affectation des cas d'utilisations aux acteurs du sprint 5

1.2. Sprint backlog

Id	User stories	Id-tâches	Tâches
1	En tant qu'agent, je peux consulter la liste des commandes existantes et les confirmer ou les annuler.	1.1	- Développer et tester les fonctionnalités de gestion des commandes incluant la consultation, la confirmation et l'annulation des commandes
2	En tant qu'agent, je peux accepter les demandes des clients telles que les ajouts, le nettoyage de tables, etc.	2.1	- Implémenter et tester les fonctionnalités de traitement des demandes.

3	En tant que client, je peux passer des commandes, poster des demandes et consulter mon historique de commandes.	3.1	<ul style="list-style-type: none"> - Établir les prototypes, les diagrammes de cas d'utilisation et les diagrammes de séquence pour passer des commandes, poster des demandes. - Développer et tester les interfaces clients.
----------	---	------------	---

Tableau 27:Sprint 5 backlog

2. Spécification fonctionnelle

2.5 Diagramme de cas d'utilisation global de sprint 5

On passe au raffinement des diagrammes par acteur.

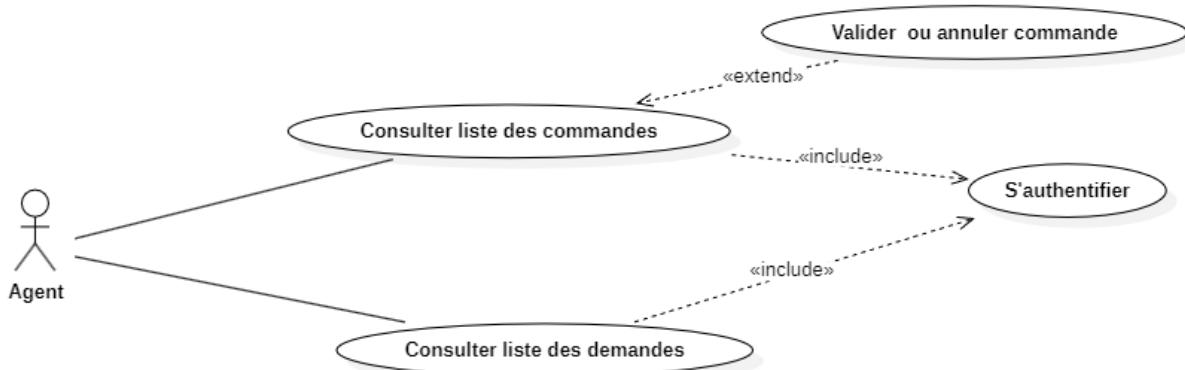


Figure 54: Diagramme de cas d'utilisation global de sprint 5 de l'agent

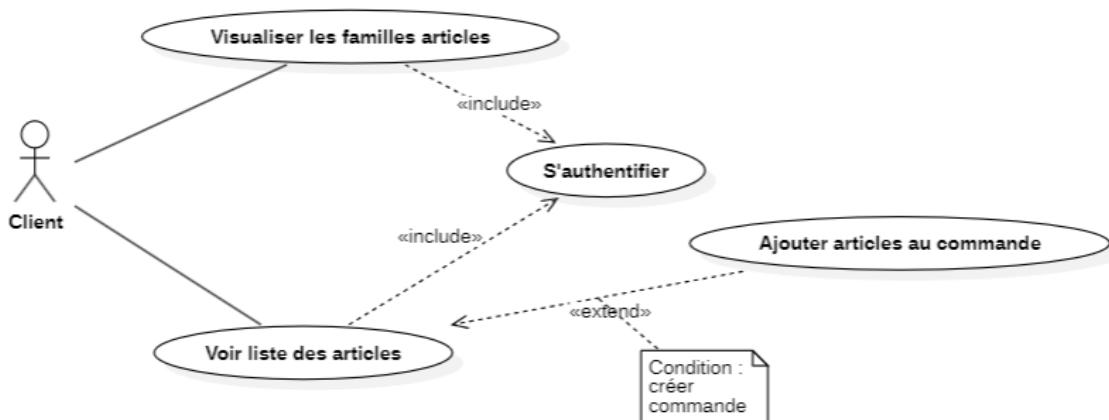


Figure 55: Diagramme de cas d'utilisation global de sprint 5 de client

2.6 Description textuelle des cas d'utilisation

Cas d'utilisation	Gérer commandes
Acteurs	Agent
Précondition	L'agent doit être authentifié avec les droits nécessaires pour gérer les commandes.
Postcondition	Les modifications apportées aux commandes sont enregistrées dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) L'agent accède à la fonctionnalité de gestion des commandes. 2) Le système affiche la liste des commandes existantes. 3) L'agent peut choisir de confirmer ou d'annuler une commande selon la situation. 4) Le système met à jour le statut de la commande en conséquence.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucune commande n'est disponible, le système n'indique aucune donnée. 2) Si une erreur survient lors de la confirmation ou de l'annulation d'une commande, le système affiche un message d'erreur.

Tableau 28:Description textuelle du cas d'utilisation "Gérer commandes"

Cas d'utilisation	Gérer demandes
Acteurs	Agent
Précondition	L'agent doit être authentifié et avoir les droits nécessaires pour gérer les demandes.
Postcondition	Les demandes des clients sont traitées et enregistrées.
Scénario nominal	<ol style="list-style-type: none"> 1) L'agent reçoit une demande d'un client (par exemple: nettoyage de table, addition). 2) L'agent traite (accepter) la demande et met à jour le système. 3) Le système confirme l'achèvement de la demande au client.

Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucune commande n'est disponible, le système n'indique aucune donnée. 2) Si une erreur survient lors de la confirmation ou de l'annulation d'une commande, le système affiche un message d'erreur.
----------------------------	---

Tableau 29:Description textuelle du cas d'utilisation "Gérer demandes"

Cas d'utilisation	Passer des commandes des demandes
Acteurs	Client
Précondition	L'agent doit être authentifié et avoir les droits nécessaires pour passer des commandes et demandes.
Postcondition	La commande et les demandes sont enregistrées dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) Le client passe une commande et/ou poste une demande via l'interface client. 2) Le système enregistre la commande et notifie l'agent. 3) Le système fournit une mise à jour au client sur le statut de la commande ou de la demande.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucune commande n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de la saisie de la commande, le système affiche un message d'erreur.

Tableau 30:Description textuelle du cas d'utilisation "Passer commandes et demandes"

2.7 Description comportementale des histoires du sprint 4

- **Diagramme de séquence « passer commandes »**

La figure 56 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « passer commandes ».

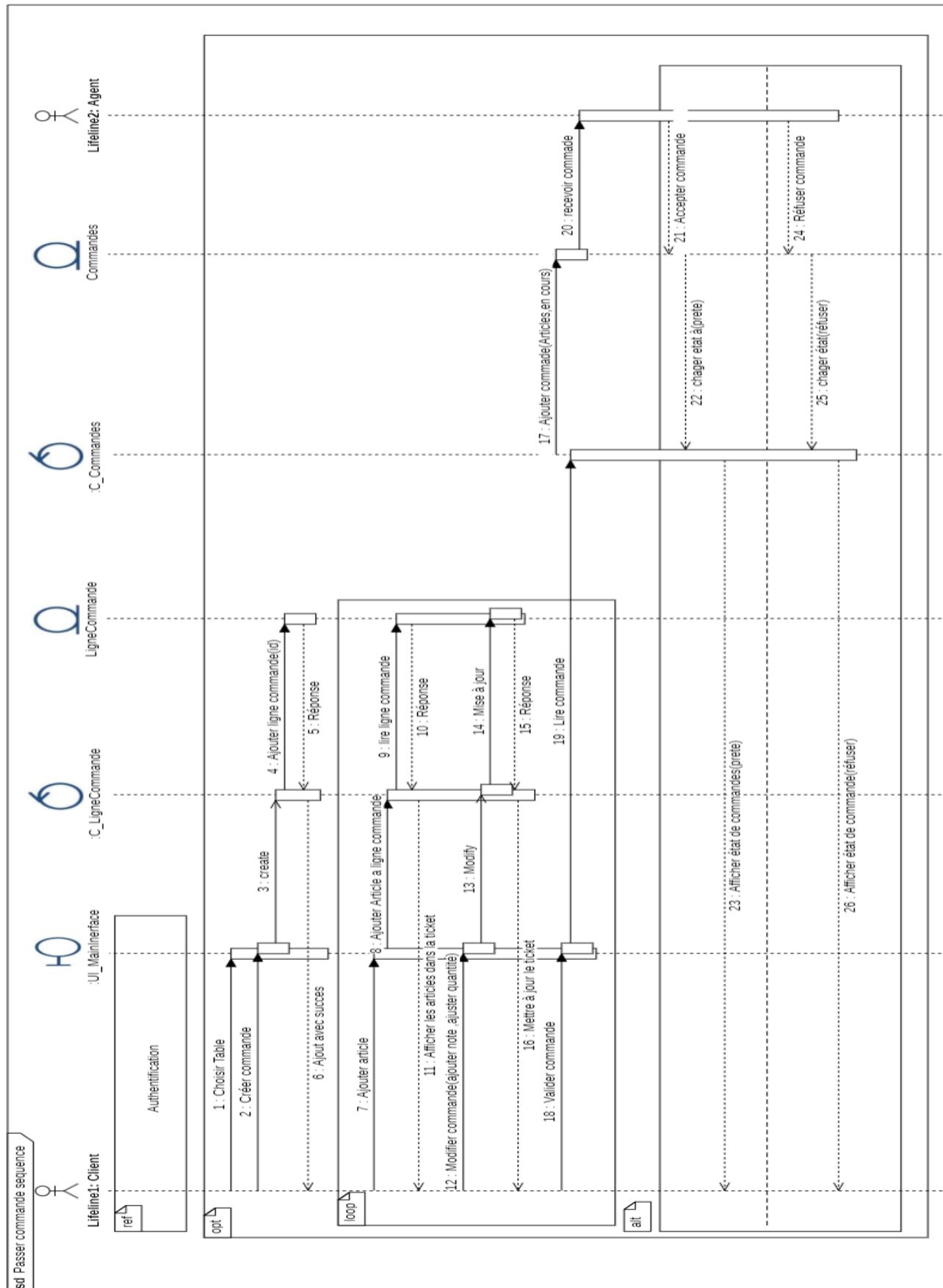


Figure 56: Diagramme de séquence de cas d'utilisation "Passer commande"

- **Diagramme de séquence « passer demande »**

La figure 57 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation « passer demande ».

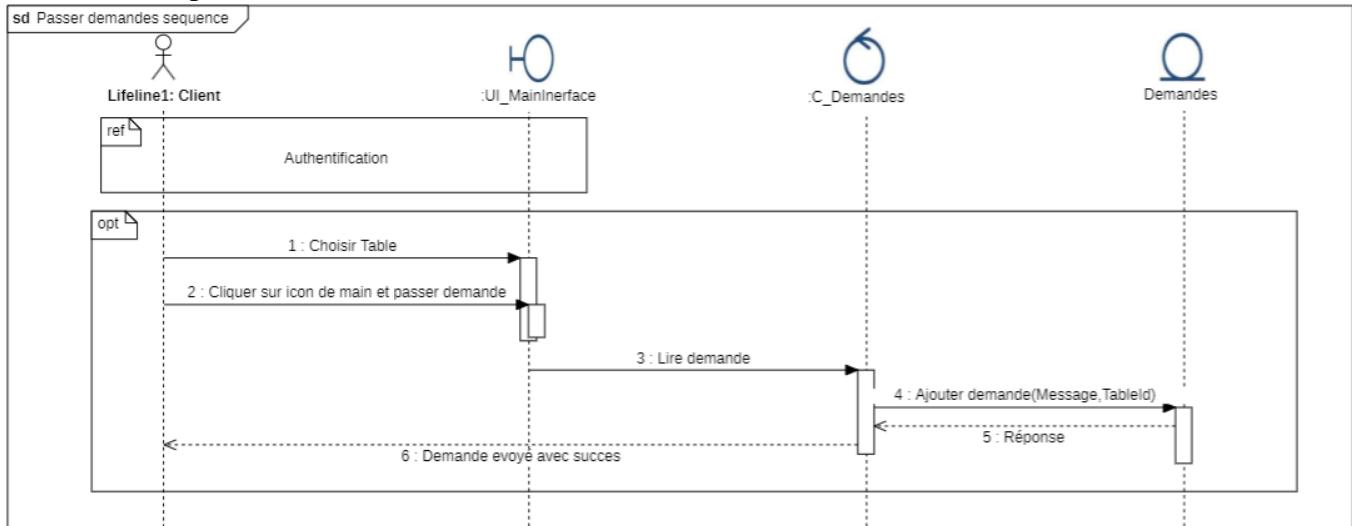


Figure 57:Diagramme de séquence de cas d'utilisation "Passer demande"

2.8 Diagramme de classe de sprint 5

La figure illustre le diagramme de classes pour le troisième sprint de notre projet. Il comporte 8 classes.

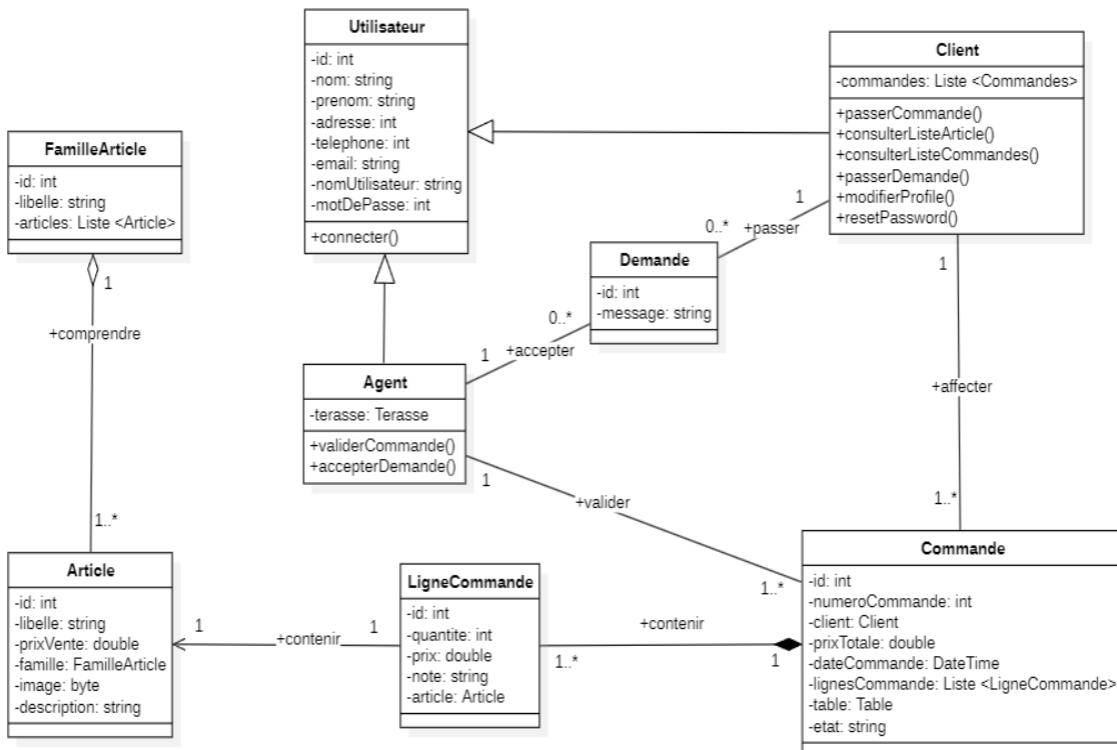


Figure 58: Diagramme de classe sprint 5

3. Test et interfaçage

3.1. Test sprint 5

Cas	Résultat
Gestion des commandes	
Gestion les demandes	Conforme

Tableau 31:Test du sprint 5

3.2. Interfaçage

- Interface de liste des commandes de l'agent

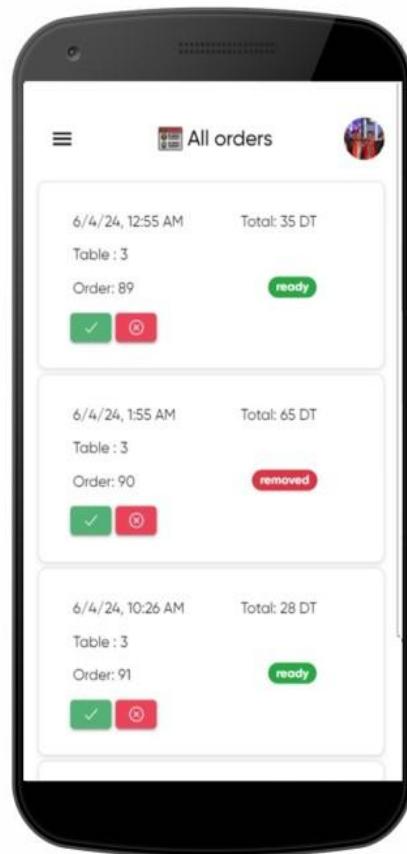


Figure 59: Interface de liste des commandes de l'agent

- **Interface de liste des commandes de client**

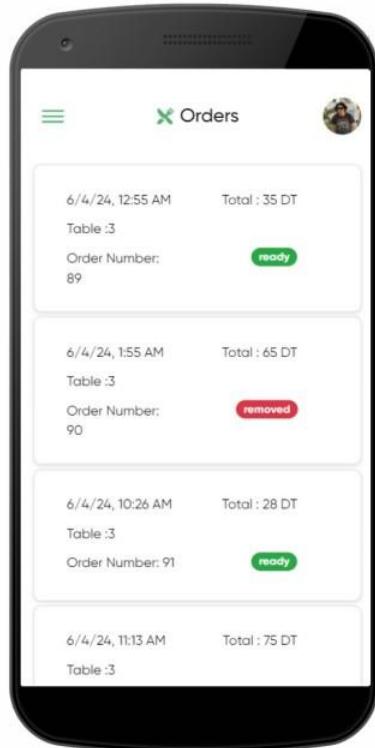


Figure 60: Interface de liste des commandes de client

- **Interface home de client**

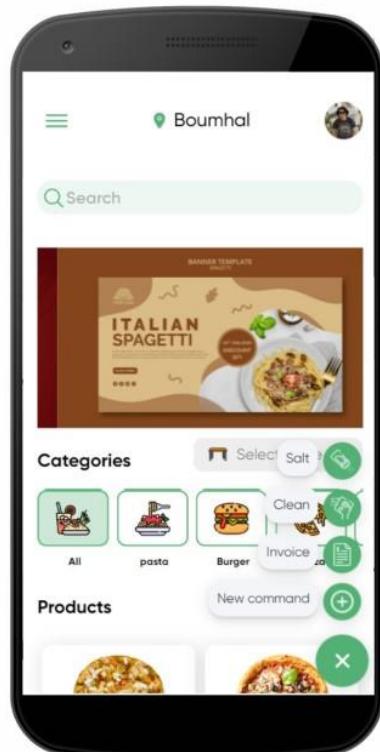


Figure 61: Interface home de client

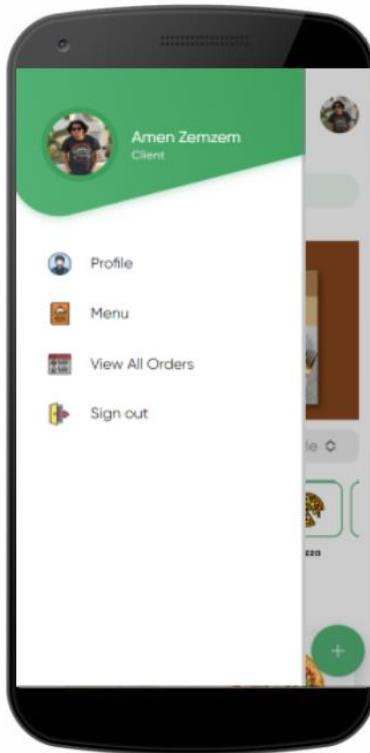


Figure 62 : Interface home de client avec side-menu

- **Interface de liste des demandes de l'agent**

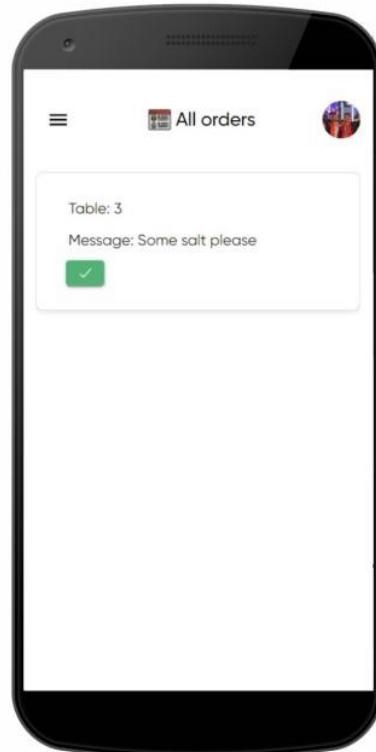


Figure 63: Interface de liste des demandes de l'agent

Conclusion

Le cinquième sprint a été crucial pour renforcer les capacités de notre système grâce à l'introduction de fonctionnalités avancées de gestion des commandes et de traitement des demandes. Les résultats des tests ont validé l'efficacité de ces fonctionnalités, marquant une étape importante vers l'amélioration continue de notre plateforme. Ce sprint a non seulement amélioré notre système, mais a également posé les bases nécessaires pour les développements futurs.

Chapitre 8 : Sprint 6 - Module gestion de tableau de bord

Introduction

Dans ce chapitre, nous abordons le sixième et dernier sprint de notre projet, axé sur l'intégration d'un tableau de bord administratif. Ce sprint est essentiel pour permettre à l'administrateur d'accéder à des visualisations de données et des statistiques en temps réel, optimisant ainsi la prise de décision basée sur les performances quotidiennes du restaurant. Nous décrirons les fonctionnalités mises en place pour consulter et analyser les commandes et les recettes du jour, soulignant les avancées techniques et les adaptations nécessaires pour répondre aux exigences opérationnelles et stratégiques de notre système.

1. Backlog du sprint 6

1.1. Affectation des cas d'utilisations aux acteurs

Le tableau ci-dessous présente l'affectation de chaque cas d'utilisation à son acteur pour notre troisième sprint :

Acteurs	Cas d'utilisation
Administrateur	Consulter des statistiques sur les commandes. Consulter la recette.

Tableau 32:Affectation des cas d'utilisations aux acteurs du sprint 6

1.2. Sprint backlog

Id	User stories	Id-tâches	Tâches
1	En tant qu'administrateur, je peux consulter la liste des commandes du jour.	1.1	-Développer et tester les fonctionnalités de consultation des commandes du jour.
2	En tant qu'administrateur, je peux consulter la recette du jour	2.1	-Développer et tester les fonctionnalités de visualisation des recettes du jour.

Tableau 33: Sprint 6 backlog

2. Spécification fonctionnelle

2.1 Diagramme de cas d'utilisation global de sprint 6

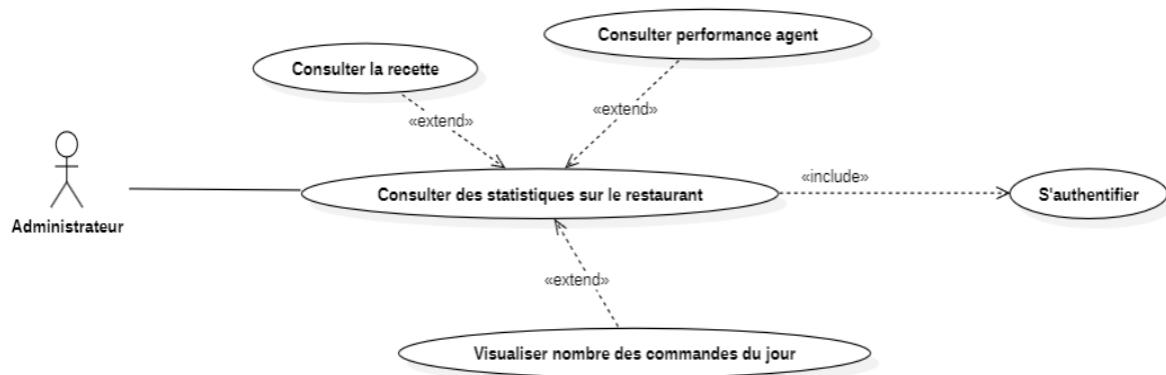


Figure 64 : Cas d'utilisation de sprint 6

2.2 Description textuelle des cas d'utilisation

Cas d'utilisation	Consulter des statistiques sur les commandes
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié.
Post condition	Les statistiques des commandes sont affichées.
Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la section statistique du tableau de bord. 2) Le système récupère les données des commandes du jour. 3) Les statistiques sont affichées, incluant le nombre total de commandes et les revenus générés. 4) L'administrateur peut filtrer ou trier les données pour une analyse plus approfondie.
Scénario alternatif	Si aucune commande n'est disponible, le système affiche un message "Aucune donnée disponible".

Tableau 34 : Consulter des statistiques sur les commandes

Cas d'utilisation	Consulter la recette du jour
Acteurs	Administrateur
Précondition	L'administrateur doit être authentifié.
Postcondition	La recette du jour est affichée.
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à la section des recettes du tableau de bord. 2. Le système affiche les recettes totales du jour. 3. L'administrateur peut visualiser des détails additionnels ou des tendances historiques.
Scénario alternatif	Si aucune recette n'est enregistrée pour la journée, le système affiche "Aucune recette disponible".

Tableau 35: Consulter la recette du jour

4. Test et interfaçage

4.1. Test sprint 5

Cas	Résultat
Consulter statistiques sur les commandes	Conforme
Visualiser recette du jour.	Conforme

Tableau 36:Test du sprint 5

4.2. Interfaçage

- Interface de tableau de bord

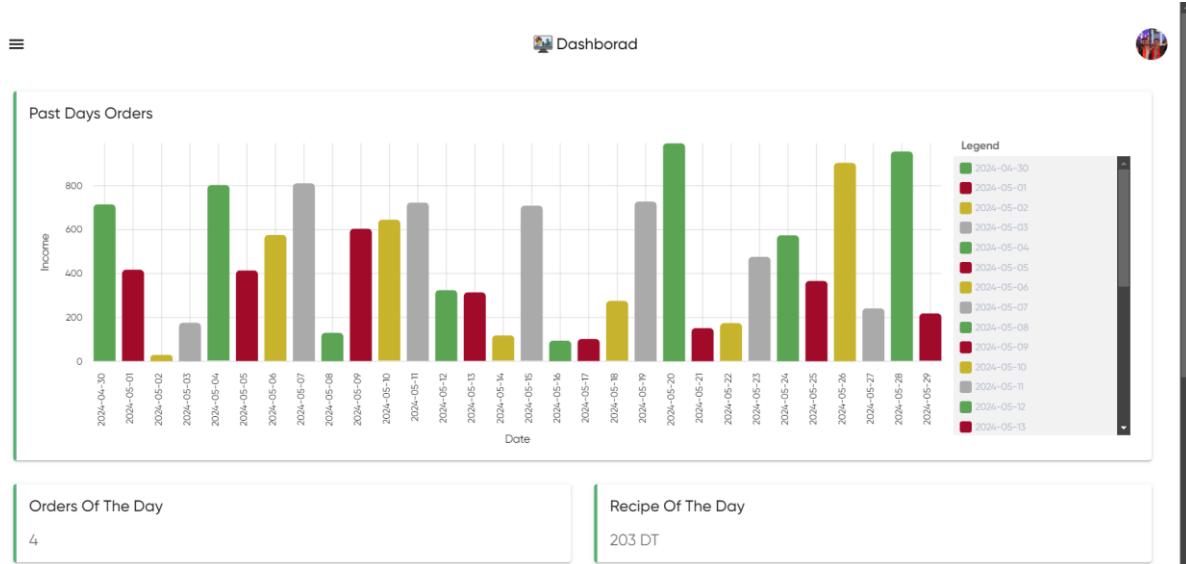


Figure 65 : Dashboard administrateur

Conclusion

Pour clôturer ce chapitre, faites le point sur l'impact des fonctionnalités développées lors de ce sprint sur l'efficacité opérationnelle du restaurant. Mettez en avant comment ces outils améliorent les décisions basées sur les données et augmentent la réactivité de l'administration face aux dynamiques du marché.

Conclusion générale

Ce projet de fin d'études a abouti à la création de notre application, une solution innovante pour la gestion des commandes dans les restaurants. Le projet visait à transformer l'expérience de commande pour les clients tout en optimisant les opérations de service grâce à une interface utilisateur intuitive et une intégration efficace de l'administration en back-office. En adoptant une méthodologie agile, nous avons pu adapter continuellement le système aux besoins changeants des utilisateurs et aux défis opérationnels, assurant ainsi une réponse rapide et appropriée aux demandes de l'environnement de restauration.

Le système développé offre de nombreuses fonctionnalités essentielles, telles que la gestion des commandes, des utilisateurs, des terrasses et des tables, ainsi que la capacité de générer des rapports statistiques détaillés en temps réel. Ces fonctionnalités permettent aux gestionnaires de restaurants de prendre des décisions éclairées basées sur les données collectées, améliorant ainsi l'efficience opérationnelle et la satisfaction des clients.

Les tests continus lors de chaque sprint ont démontré la fiabilité et la robustesse du système, et les retours des utilisateurs finaux ont été extrêmement positifs, soulignant la facilité d'utilisation et les améliorations par rapport aux systèmes de gestion traditionnels. En conclusion, l'application n'est pas seulement une solution technologique; c'est une transformation stratégique dans la gestion des restaurants, rendant les processus plus fluides et moins susceptibles aux erreurs.

Webographie

- [1] Visual Studio Code [En ligne] <https://code.visualstudio.com/docs> [Consulté le 04/02/2024].
- [2] SQL Server [En ligne] <https://docs.microsoft.com/en-us/sql/sql-server/> [Consulté le 12/02/2024].
- [3] Postman [En ligne] <https://www.postman.com/> [Consulté le 21/02/2024].
- [4] Visual Studio [En ligne] <https://visualstudio.microsoft.com/> [Consulté le 02/03/2024].
- [5] StarUML [En ligne] <http://staruml.io/> [Consulté le 10/03/2024].
- [6] Microsoft IIS [En ligne] <https://www.iis.net/> [Consulté le 18/03/2024].
- [7] Android Studio [En ligne] <https://developer.android.com/studio> [Consulté le 25/03/2024].
- [8] C# [En ligne] <https://docs.microsoft.com/en-us/dotnet/csharp/> [Consulté le 01/04/2024].
- [9] TypeScript [En ligne] <https://www.typescriptlang.org/> [Consulté le 07/04/2024].
- [10] Angular [En ligne] <https://angular.io/> [Consulté le 13/04/2024].
- [11] HTML [En ligne] <https://developer.mozilla.org/en-US/docs/Web/HTML> [Consulté le 20/04/2024].
- [12] CSS [En ligne] <https://developer.mozilla.org/en-US/docs/Web/CSS> [Consulté le 25/04/2024].
- [13] .NET Core [En ligne] <https://dotnet.microsoft.com/en-us/> [Consulté le 01/05/2024].
- [14] Entity Framework [En ligne] <https://docs.microsoft.com/en-us/ef/> [Consulté le 08/05/2024].
- [15] JWT [En ligne] <https://jwt.io/> [Consulté le 15/05/2024].
- [16] Microsoft Word [En ligne] <https://www.microsoft.com/en-us/microsoft-365/word> [Consulté le 21/05/2024].
- [17] SMTP [En ligne] <https://support.google.com/mail/answer/7126229?hl=en> [Consulté le 25/05/2024].
- [18] Capacitor [En ligne] <https://capacitorjs.com> [Consulté le 28/05/2024].
- [19] Architecture 3-tiers [En ligne] https://en.wikipedia.org/wiki/Multitier_architecture [Consulté le 01/06/2024].
- [20] Ionic [En ligne] <https://ionicframework.com> [Consulté le 01/06/2024].